SREYAS INSTITUTE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUH)

2-50/5, SyNo.107, Tattiannaram(V), G.S.I Bandlaguda, Nagole, Hyderabad – 500 068

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMPUTER NETWORKS & WEB TECHNOLOGIES LAB MANUAL

III-B Tech – I Semester [Branch: CSE(R18)]

SREYAS INSTITUTE OF ENGINEERING & TECHNOLOGY

Besides Indu Aranya, Nagole, Hyderabad.

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE, Affiliated to JNTUH)

G.S.I. Bandlaguda, Nagole, Hyderabad - 500068

CERTIFICATE

LAB NAME : COMPUTER NETWORKS & WEB TECHNOLOGIES LAB

BRANCH : CSE

YEAR & SEM : III – I

REGULATION : R18

SREYAS INSTITUTE OF ENGINEERING & TECHNOLOGY
(Approved by AICTE, New Delhi & Affiliated to JNTUH)
Thatti Annaram(v), Bandlaguda, Nagole, Hyderabad - 500068.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION & MISSION OF INSTITUTION

VISION

To be a centre of excellence in technical education to empower the young talent through quality education and innovative engineering for well being of the society

MISSION

1. Provide quality education with innovative methodology and intellectual human capital.
2. Provide conducive environment for research and developmental activities.
3. Inculcate holistic approach towards nature, society and human ethics with lifelong learning attitude.

SREYAS INSTITUTE OF ENGINEERING & TECHNOLOGY
(Approved by AICTE, New Delhi & Affiliated to JNTUH)
Thatti Annaram(v), Bandlaguda, Nagole, Hyderabad - 500068.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VISION & MISSION OF DEPARTMENT

Vision

To excel in computer science engineering education with best learning practices, research and professional ethics.

Mission

1. To offer technical education with innovative teaching, good infrastructure and qualified human resources.
2. Accomplish a process to advance knowledge in the subject and promote academic and research environment.
3. To impart moral and ethical values and interpersonal skills to the students.

Program Educational Objectives

Computer Science & Engineering (CSE) is one of the most prominent technical fields in Engineering. The curriculum offers courses with various areas of emphasis on theory, design and experimental work. Subject matter ranges from basics of Computers & Programming Languages to Compiler Design and Cloud Computing. It maintains strong tie-ups with industry and is dedicated to preparing

students for a career in Web Technologies, Object Oriented Analysis and Design, Networking & Security, Databases, Data Mining & Data Warehousing and Software Testing.

PROGRAM OUTCOMES(POs)

Engineering Graduates will be able to:

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

13. Proficiency on the contemporary skills towards development of innovative apps and firmware products.
14. Capabilities to participate in the construction of software systems of varying complexity.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

III Year B.Tech CSE- II Sem

WEB TECHNOLOGIES LAB INSTRUCTIONS TO THE STUDENTS

Things to Do:

1) Students should come in formal dresses.
2) Students must wear their id cards.
3) They have to be in the lab before 10 minutes.
4) They should come up with the observation and the record.
5) Observation should get corrected with the concerned faculty.
6) The programs corrected by the faculty have to copy to record.
7) They should maintain silence in the lab.

Things not to do:

1) Students should not bring any electronic gadgets into the lab.
2) They should not come late.
3) You should not create any disturbances to others.

HOD Lab Incharge

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

III Year B.Tech. CSE. I – Sem **L T P C**

Course Code: CS605PC **3 0 0 3**

COMPUTER NETWORKS & WEB TECHNOLOGIES LAB

Course Objectives

1. To understand the working principle of various communication protocols.
2. To understand the network simulator environment and visualize a network topology and observe its performance
3. To analyze the traffic flow and the contents of protocol frames

Course Outcomes

1. Implement data link layer farming methods
2. Analyze error detection and error correction codes.
3. Implement and analyze routing and congestion issues in network design.
4. Implement Encoding and Decoding techniques used in presentation layer
5. To be able to work with different network tools

List of Experiments

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

4. Implement Dijsktra's algorithm to compute the shortest path through a network
5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6. Implement distance vector routing algorithm for obtaining routing tables at each node.
7. Implement data encryption and data decryption
8. Write a program for congestion control using Leaky bucket algorithm.
9. Write a program for frame sorting technique used in buffers.
10. **Wireshark**
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
11. How to run Nmap scan
12. Operating System Detection using Nmap
13. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
    v. Simulate to Compare Data Rate& Throughput.
    vi. Simulate to Plot Congestion for Different Source/Destination
    vii. Simulate to Determine the Performance with respect to Transmission of Packets

Web Technologies Experiments

1. Write a PHP script to print prime numbers between 1-50.
2. PHP script to
    a. Find the length of a string.
    b. Count no of words in a string.
    c. Reverse a string.
    d. Search for a specific string.
3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.
4. Write a PHP script that reads data from one file and write into another file.
5. Develop static pages (using Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages.
    a) Home page
    b) Registration and user Login
    c) User Profile Page
    d) Books catalog
    e) Shopping Cart
    f) Payment By credit card
    g) Order Conformation
6. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

7. Create and save an XML document on the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.
8. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
9. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

TEXT BOOKS:

1. WEB TECHNOLOGIES: A Computer Science Perspective, Jeffrey C. Jackson, Pearson Education.

REFERENCES:

1. Deitel H.M. and Deitel P.J., "Internet and World Wide Web How to program", Pearson International, 2012, 4th Edition.
2. J2EE: The complete Reference By James Keogh, McGraw-Hill
3. Bai and Ekedhi, The Web Warrior Guide to Web Programming, Thomson
4. Paul Dietel and Harvey Deitel," Java How to Program", Prentice Hall of India, 8th Edition
5. Web technologies, Black Book, Dreamtech press.
6. Gopalan N.P. and Akilandeswari J., "Web Technology", Prentice Hall of India

INDEX

	iii. Viewing Captured Traffic iv. Analysis and Statistics & Filters.	
11	How to run Nmap scan	52-57
12	Operating System Detection using Nmap	58-60
13	1. Do the following using NS2 Simulator i. NS2 Simulator-Introduction ii. Simulate to Find the Number of Packets Dropped iii. Simulate to Find the Number of Packets Dropped by TCP/UDP. iv. Simulate to Find the Number of Packets Dropped due to Congestion v. Simulate to Compare Data Rate& Throughput. vi. Simulate to Plot Congestion for Different Source/Destination vii. Simulate to Determine the Performance with respect to Transmission of Packets.	61-86
	Web Technologies Experiments	
1	Write a PHP script to print prime numbers between 1-50.	87-88
2	PHP script to a. Find the length of a string. b. Count no of words in a string. c. Reverse a string. d. Search for a specific string.	89-93
3	Write a PHP script to merge two arrays and sort them as numbers, in descending order.	94
4	Write a PHP script that reads data from one file and write into another file.	95-96
5	Develop static pages (using Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages. a) Home page b) Registration and user Login c) User Profile Page d) Books catalog e) Shopping Cart f) Payment By credit card	97-106

1. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.

SOURCE CODE:

NAME OF THE EXPERIMENT: Bit Stuffing.

AIM: Implement the data link layer framing methods such as and bit stuffing.

HARDWARE REQUIREMENTS: Intel based Desktop PC:- RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

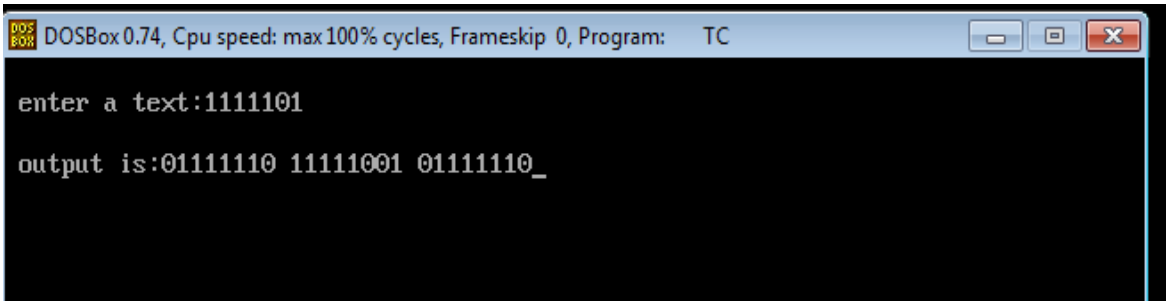
The new technique allows data frames to contain an arbitrary number if bits and allows character codes with an arbitrary no of bits per character. Each frame begins and ends with special bit pattern, 01111110, called a flag byte. When ever the senders data link layer encounters five consecutive one's in the data, it automatically stuffs a 0 bit into the out going bit stream.

SOURCE CODE:

```
// BIT Stuffing program
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
 char ip[50],op[100];
 int i,ln,j;
 clrscr();
 printf("\n enter a text:");
 gets(ip);
 //ln=strlen(ip);

 for(i=0,j=0;ip[i]!='\0';i++,j++)
 {
 op[j]=ip[i];
  if(strncmp(ip+i,"11111",5)==0)
  {
    op[j]='\0';
    strcat(op,"111110");
    i=i+4;
    j=j+5;
  }
  }
  op[j]='\0';
   printf("\n output is:01111110 %s 01111110",op);
```

```
getch();
}
```

OUTPUT:



VIVA QUESTIONS:

1. What is bit stuffing?
2. What is the use of bit stuffing?
3. with bit stuffing the boundary b/w 2 frames can be unambiguously recognized by -------
4. Is analogous to character stuffing
5. Each frame begins and ends with a special bit pattern 01111110 called ---
6. The senders data link layer encounters ----------no of 1's consecutively

NAME OF THE EXPERIMENT: Character Stuffing.

AIM: Implement the data link layer framing methods such as character, character stuffing.

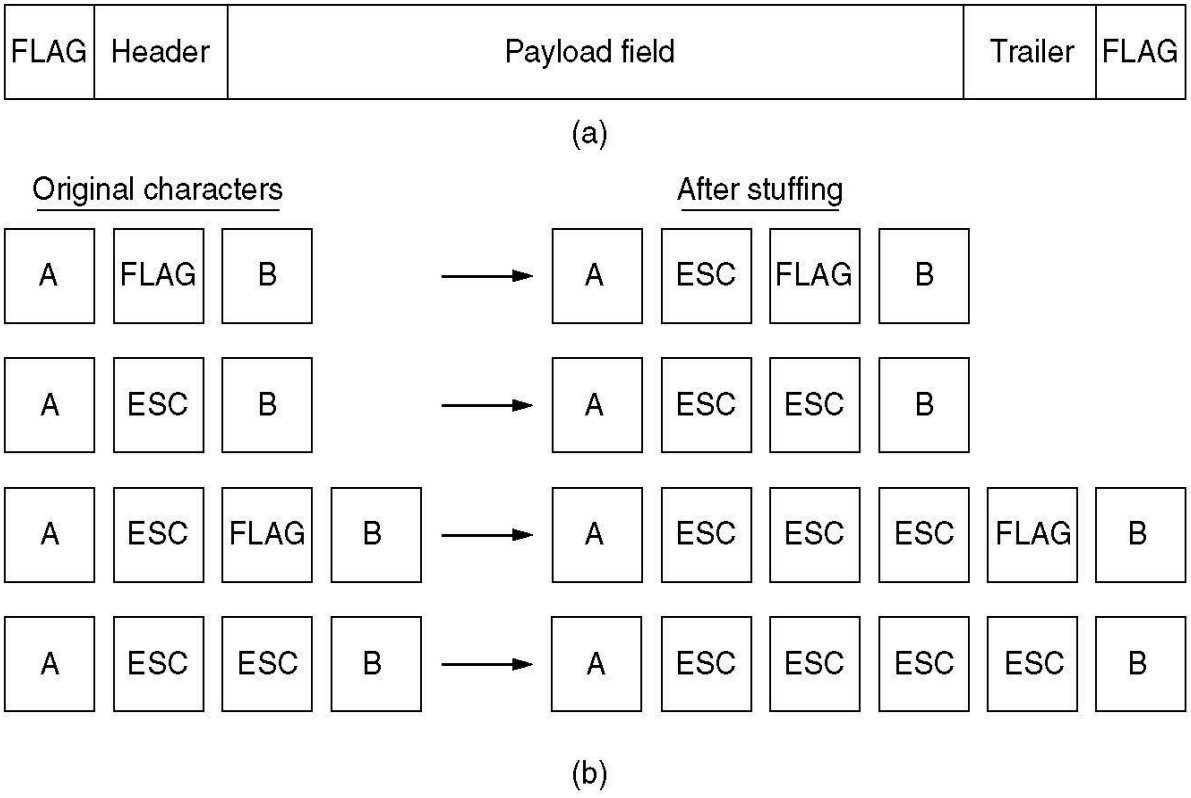
HARDWARE REQUIREMENTS: Intel based Desktop PC**:-**

RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

- Breaking the bit stream up into frames is more difficult than it at first appears.
- One way to achieve this framing is insert time gaps.
- One of four other methods is :
- Flag byte with byte stuffing or character stuffing.



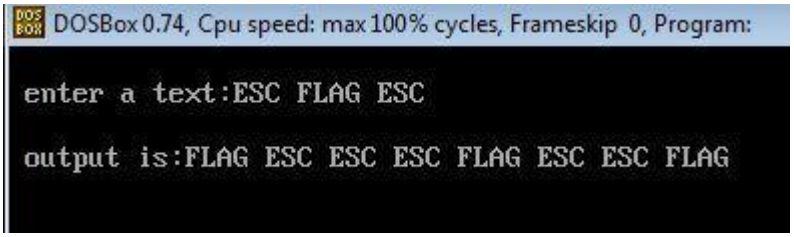
SOURCE CODE:

//PROGRAM FOR CHARACTER STUFFING

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
  char ip[100],op[100];
  int i,ln,j;
  clrscr();
  printf("\n enter a text:");
  gets(ip);
  //ln=strlen(ip);

  for(i=0,j=0;ip[i]!='\0';i++,j++)
  {
    op[j]=ip[i];
    if( strncmp(ip+i,"FLAG",4)==0 )
    {
      op[j]='\0';
```

```
      strcat(op,"ESC FLAG");
      i=i+3;
      j=j+7;
    }
     if( strncmp(ip+i,"ESC",3)==0 )
    {
      op[j]='\0';
      strcat(op,"ESC ESC");
      i=i+2;
      j=j+6;
    }
  }
  op[j]='\0';

  printf("\n output is:FLAG %s FLAG",op);
  getch();
  }
```

OUTPUT:



VIVA QUESTIONS:

1. What is character stuffing?
2. What is the use of character stuffing?
3. _____ is analogous to bit stuffing.
4. _____ are the delimiters for character stuffing
5. Expand DLE STX_____
6. Expand DLE ETX_____

2. Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC CCIP.

SOURCE CODE:

NAME OF THE EXPERIMENT: Cyclic Redundancy Check.

AIM: Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.

HARDWARE REQUIREMENTS: Intel based Desktop PC**:-** RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

CRC (Cyclic Redundancy Check) : Polynomial Codes are based upon treating bit strings as representations of polynomials with coefficients of 0 and 1 only.A k-bit frame is represented as the coefficient list for a polynomial with k-terms ranging from x^{k-1} to x^0 .Such a polynomial is said to be of degree k-1.

The high order (left most) bit is the coefficient of x^{k-1}.The next bit is the coefficient of x^{k-2}, and so on.. For ex: the polynomial $x^5 + x^4 + 1$, results to the polynomial $x^5 + x^4 + x^0$

The algorithm for computing the checksum is as follows:

1. Let r be the degree of G(x). Append r zero bits to the low-order end of the frame so that it contains m+r bits and corresponds to the polynomial x^r .M(x)
2. divide the bit string corresponding to G(x) into the bit string corresponding to x^r .M(x) , using modulo 2 division.
3. Append the remainder (which is always r or fewer bits) from the bit string corresponding to x^r .M(x) using modulo 2 subtraction. The result is the checksummed frame to be transmitted. Call its polynomial T(x).

SOURCE CODE:

```
//PROGRAM FOR CYCLIC REDUNDENCY CHECK
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char m[30],g[20],tm[20];
  int ml,gl,i,j,f;
  clrscr();
  printf("\n enter message");
  gets(m);
  ml=strlen(m);
```

```
printf("\n msg length:%d",ml);
printf("\n enter generator or divisor");
 gets(g);
gl=strlen(g);
printf("\n generator msg length:%d",gl);
for(i=0;i<gl-1;i++)
m[ml+i]='0';
m[ml+i]='\0';
printf("\n after padding,msg is:%s",m);
for(i=0;i<gl;i++)
tm[i]=m[i];
 for(i=0;i<ml;i++)
{
            if(tm[0]=='1')
            {
             for(j=0;j<gl;j++)
             {
                if(tm[j]==g[j])
                tm[j]='0';
                else
                tm[j]='1';
              }
             }
               for(j=0;j<gl-1;j++)
               tm[j]=tm[j+1];
               tm[gl-1]=m[i+gl];
 }
tm[gl-1]='\0';
printf("\n remainder is:%s",tm);
for(i=0;i<gl-1;i++)
   m[ml+i]=tm[i];
            m[ml+i]='\0';
printf("\n after adding remainder,msg is:%s",m);
//*************************************************
 printf("\n enter the received message");
 gets(m);
for(i=0;i<gl;i++)
tm[i]=m[i];
  for(i=0;i<ml;i++)
{
            if(tm[0]=='1')
            {
```

```
                for(j=0;j<gl;j++)
                {
                   if(tm[j]==g[j])
                   tm[j]='0';
                   else
                   tm[j]='1';
                 }
                }
                  for(j=0;j<gl-1;j++)
                  tm[j]=tm[j+1];
                  tm[gl-1]=m[i+gl];
    }
   tm[gl-1]='\0';
   printf("\n remainder is:%s",tm);
   for(i=0,f=0;i<gl-1;i++)
   {
               if(tm[i]!='0')
               {f=1;break;}
   }
   if(f==0)
   printf("\n no error in received message");
   else
   printf("\n error in received message");
   getch();
}
```

OUTPUT:

```
enter message1011

msg length:4
enter generator or divisor101

generator msg length:3
after padding,msg is:101100
remainder is:01
after adding remainder,msg is:101101
enter the received message101101

remainder is:00
no error
```

```
enter message1011

msg length:4
enter generator or divisor101

generator msg length:3
after padding,msg is:101100
remainder is:01
after adding remainder,msg is:101101
enter the received message110001

remainder is:10
error
```

VIVA QUESTIONS:

1. What is CRC?
2. What is the use of CRC?
3. Name the CRC standards
4. Define checksum?
5. Define generator polynomial?
6. Polynomial arithmetic is done by_____

3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

NAME OF THE EXPERIMENT: Go Back N Sliding Window Protocol.

AIM: Develop a simple data link layer that performs the flow control using the sliding window protocol,and loss recovery using the Go-Back-N mechanism.

HARDWARE REQUIREMENTS: Intel based Desktop PC**:-** RAM of 512 MB

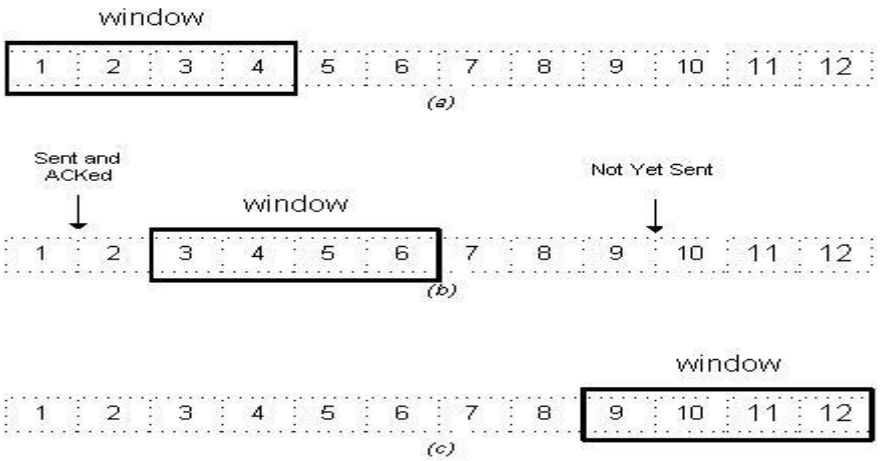
SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

In computer networks sliding window protocol is a method to transmit data on a network. Sliding window protocol is applied on the Data Link Layer of OSI model. At data link layer data is in the form of frames. In Networking, Window simply means a buffer which has data frames that needs to be transmitted.

Both sender and receiver agrees on some window size. If window size=w then after sending w frames sender waits for the acknowledgement (ack) of the first frame.

As soon as sender receives the acknowledgement of a frame it is replaced by the next frames to be transmitted by the sender. If receiver sends a collective or cumulative acknowledgement to sender then it understands that more than one frames are properly received, for eg:- if ack of frame 3 is received it understands that frame 1 and frame 2 are received properly.



In sliding window protocol the receiver has to have some memory to compensate any loss in transmission or if the frames are received unordered.

Efficiency of Sliding Window Protocol

$\eta = (W*t_x)/(t_x+2t_p)$

W = Window Size

t_x = Transmission time

t_p = Propagation delay

Sliding window works in full duplex mode

It is of two types:-

1. Selective Repeat: Sender transmits only that frame which is erroneous or is lost.

2. Go back n: Sender transmits all frames present in the window that occurs after the error bit including error bit also.

SOURCE CODE:

```
//PROGRAM FOR Go Back N Protocol


#include<stdio.h>
int main()
{
int windowsize,sent=0,ack,i;
printf("enter window size\n");
scanf("%d",&windowsize);
while(1)
{
for( i = 1; i <= windowsize; i++)
{
printf("Frame %d has been transmitted.\n",sent);
sent++;
if(sent == windowsize)
break;
}
printf("\nPlease enter the last Acknowledgement received.\n");
scanf("%d",&ack);
if(ack == windowsize)
break;
else
sent = ack;
}
return 0;
}
```

OUTPUT:

```
Enter window size

8

Frame 0 has been transmitted.

Frame 1 has been transmitted.

Frame 2 has been transmitted.
```

Frame 3 has been transmitted.

Frame 4 has been transmitted.

Frame 5 has been transmitted.

Frame 6 has been transmitted.

Frame 7 has been transmitted.

Please enter the last Acknowledgement received.

2

Frame 2 has been transmitted.

Frame 3 has been transmitted.

Frame 4 has been transmitted.

Frame 5 has been transmitted.

Frame 6 has been transmitted.

Frame 7 has been transmitted.

Please enter the last Acknowledgement received.

8

VIVA QUESTIONS:

1.What is Flow Control?
2. In Go-Back-N ARQ, if frames 4, 5, and 6 are received successfully, the receiver may send an ACK _____ to the sender.
3.What is the Maximum window size for Go Back N protocol?
4.List different types of Sliding Window Protocols?
5.What is the efficiency of Go Back N protocol?

4. Implement Dijsktra's algorithm to compute the shortest path through a network.

NAME OF THE EXPERIMENT: Shortest Path.

AIM: Implement Dijkstra's algorithm to compute the Shortest path through a given graph.

HARDWARE REQUIREMENTS: Intel based Desktop PC**:-** RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

- Each node is labeled (in parentheses) with its distance from the source node along the best known path.
- Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths.
- A label may be either tentative or permanent.
- Initially, all labels are tentative.
- When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

SOURCE CODE:

```
//.PROGRAM FOR FINDING SHORTEST PATH FOR A GIVEN GRAPH
#include<stdio.h>
#include<conio.h>
```

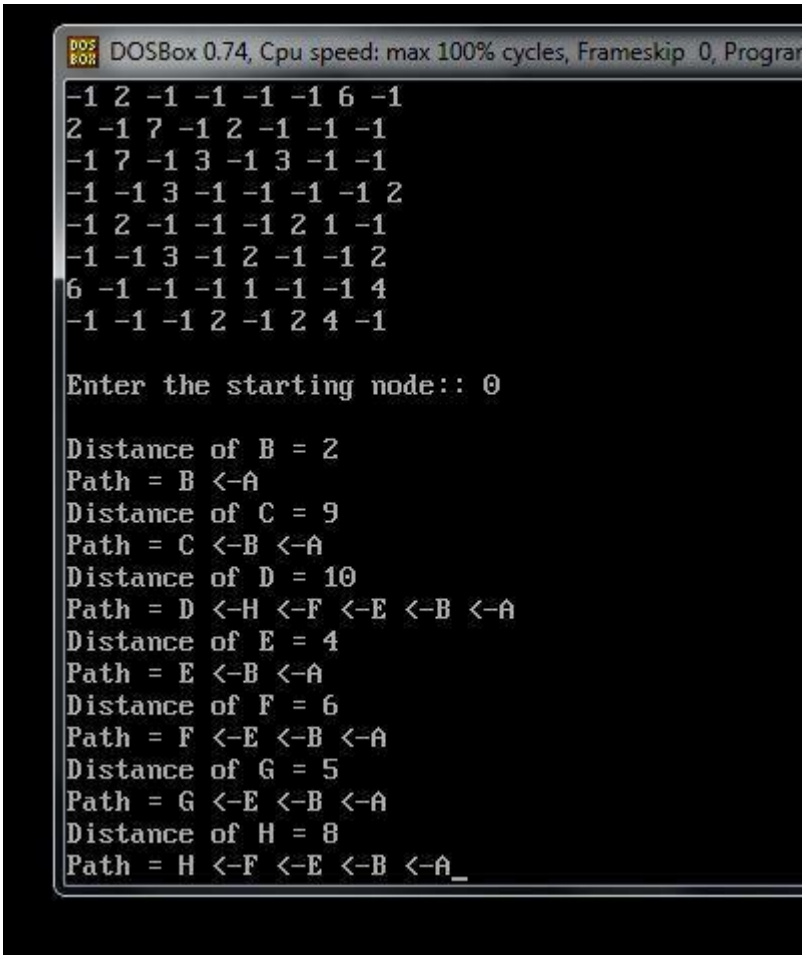
```
#define INFINITY 9999
#define MAX 10
void dijikstra(int G[MAX][MAX], int n, int startnode);
void main(){
        int G[MAX][MAX], i, j, n, s;
        clrscr();
        printf("\nEnter the no. of vertices:: ");
        scanf("%d", &n);
        printf("\nEnter the adjacency matrix,-1 for no edge::\n");
        for(i=0;i < n;i++)
        {
                for(j=0;j < n;j++)
            {
                        scanf("%d", &G[i][j]);
                 if(G[i][j]==-1)
                   G[i][j]=INFINITY;
            }
      }
        printf("\nEnter the starting node:: ");
        scanf("%d", &s);
        dijikstra(G,n,s);
        getch();
}

void dijikstra(int G[MAX][MAX], int n, int startnode)
{
        int distance[MAX], pred[MAX];
        int visited[MAX], count, mindistance, nextnode, i,j;

        for(i=0;i< n;i++)
        {
                distance[i]=G[startnode][i];
                pred[i]=startnode;
                visited[i]=0;//0 means tentative
        }
        distance[startnode]=0;
        visited[startnode]=1;// 1 means permanent
        count=1;
        while(count < n-1)
        {
           // find node with minimum distance  from all the nodes
                mindistance=INFINITY;
                for(i=0;i < n;i++)
                {
                        if(distance[i] < mindistance&&(visited[i]==0))
                        {
                                mindistance=distance[i];
```

```
                        nextnode=i;
                    }
                }
                visited[nextnode]=1;// make that node with minimum distance as permanent
            // find the distance and predecessor node of this node
                for(i=0;i < n;i++)
                {
                    if(visited[i]==0)
                    {
                        if(mindistance+G[nextnode][i] < distance[i])
                        {
                            distance[i]=mindistance+G[nextnode][i];
                            pred[i]=nextnode;
                        }
                }
                }
            count++;
        }

        for(i=0;i < n;i++)
        {
            if(i!=startnode)
            {
                printf("\nDistance of %c = %d", i+65, distance[i]);
                printf("\nPath = %c", i+65);
                j=i;
                do
                {
                    j=pred[j];
                    printf(" <-%c", j+65);
                }
                while(j!=startnode);
            }
        }
}
```

OUTPUT:

Enter the no. of vertices:: 8

Enter the adjacency matrix,-1 for no edge::



5. Take an example subnet of hosts and obtain a broadcast tree for the subnet.

NAME OF THE EXPERIMENT: Broadcast tree

AIM: C program to obtain broadcast tree for a given graph.

HARDWARE REQUIREMENTS: Intel based Desktop PC**:-** RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

- Sending a packet to all destinations simultaneously is called Broadcasting.
- Many methods are there:
- 1. sourse to simply send a packet to each destination.

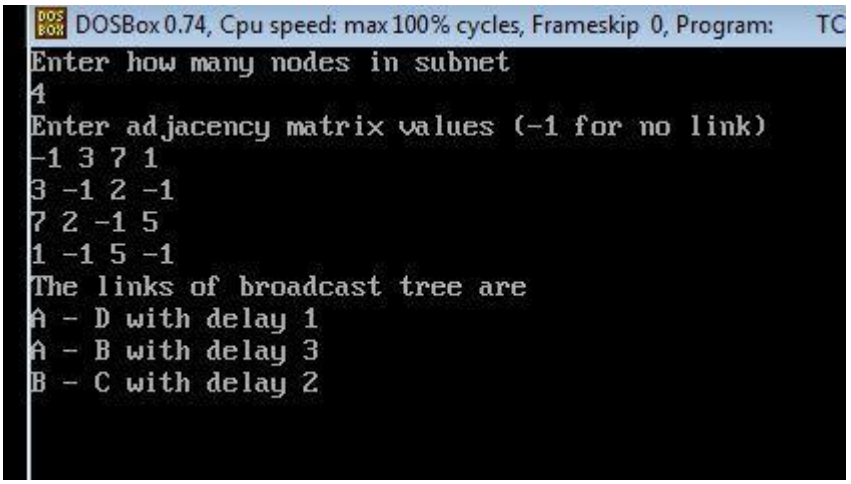
- 2. flooding is another method.
- It generates too many packets and consumes too much space.

SOURCE CODE:

```
// broadcast tree based on minimum spanning tree
#include<stdio.h>

void main()
{
int a[100][100],i,j,n,k,l,v[100],min=32765,c;
clrscr();
printf("Enter how many nodes in subnet\n");
scanf("%d",&n);
printf("Enter adjacency matrix values (-1 for no link)\n");
for(i=0;i<n;i++)
{
v[i]=0;
for(j=0;j<n;j++)
{
scanf("%d",&a[i][j]);
if(a[i][j]!=-1 && min>a[i][j])
{
min=a[i][j];
k=i;
l=j;
}
}
}
printf("The links of broadcast tree are\n");
printf("%c - %c with delay %d\n",k+65,l+65,min);
v[k]=v[l]=1;
c=1;
while(c<=n-2)
{
min=32765;
for(i=0;i<n;i++)
{
                if(v[i]==1)
                {
                  for(j=0;j<n;j++)
                  {
                     if(a[i][j]!=-1 && v[j]==0 && min>a[i][j])
```

```
                    {
                    min=a[i][j];
                    k=i;
                    l=j;
                    }
                  }
                }
  }
v[l]=1;
printf("%c - %c with delay %d\n",k+65,l+65,min);
c++;

}

getch();
}
```

OUTPUT:

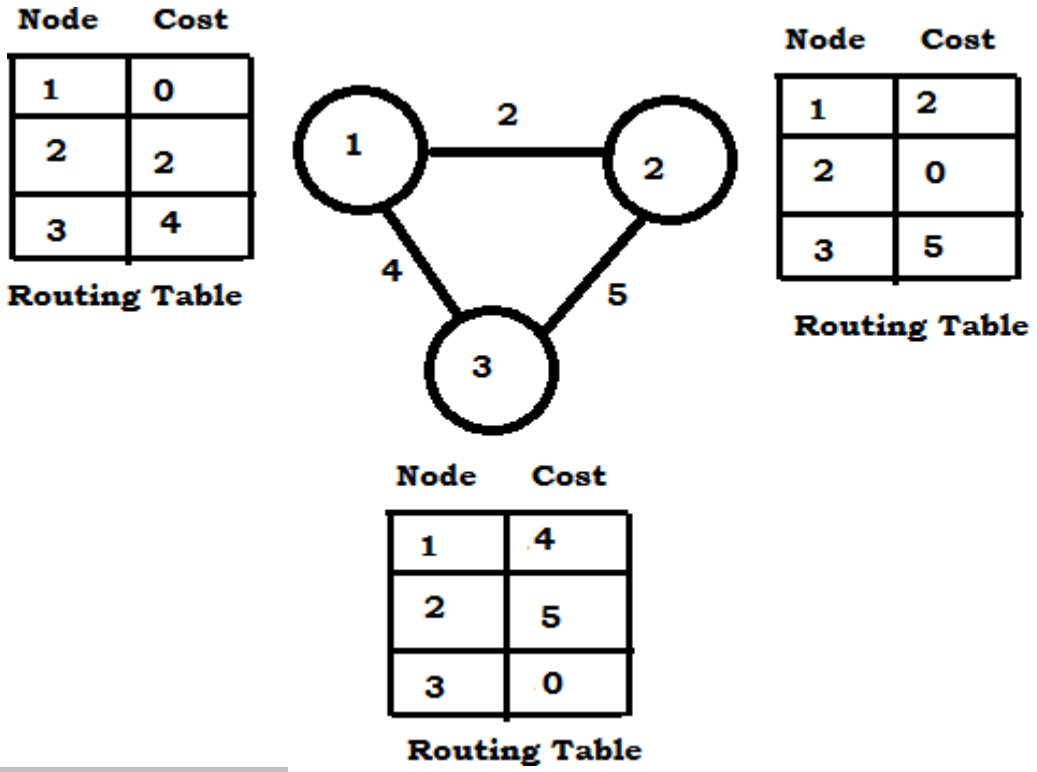


6. Implement distance vector routing algorithm for obtaining routing tables at each node.

SOURCE CODE:

NAME OF THE EXPERIMENT: Distance Vector routing.

AIM: Obtain Routing table at each node using distance vector routing algorithm for obtaining routing tables at each node



HARDWARE REQUIREMENTS: Intel based Desktop PC:- RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

Distance Vector Routing Algorithms calculate a best route to reach a destination based solely on distance. E.g. RIP. RIP calculates the reach ability based on hop count. It's different from link state algorithms which consider some other factors like bandwidth and other metrics to reach a destination. Distance vector routing algorithms are not preferable for complex networks and take longer to converge.

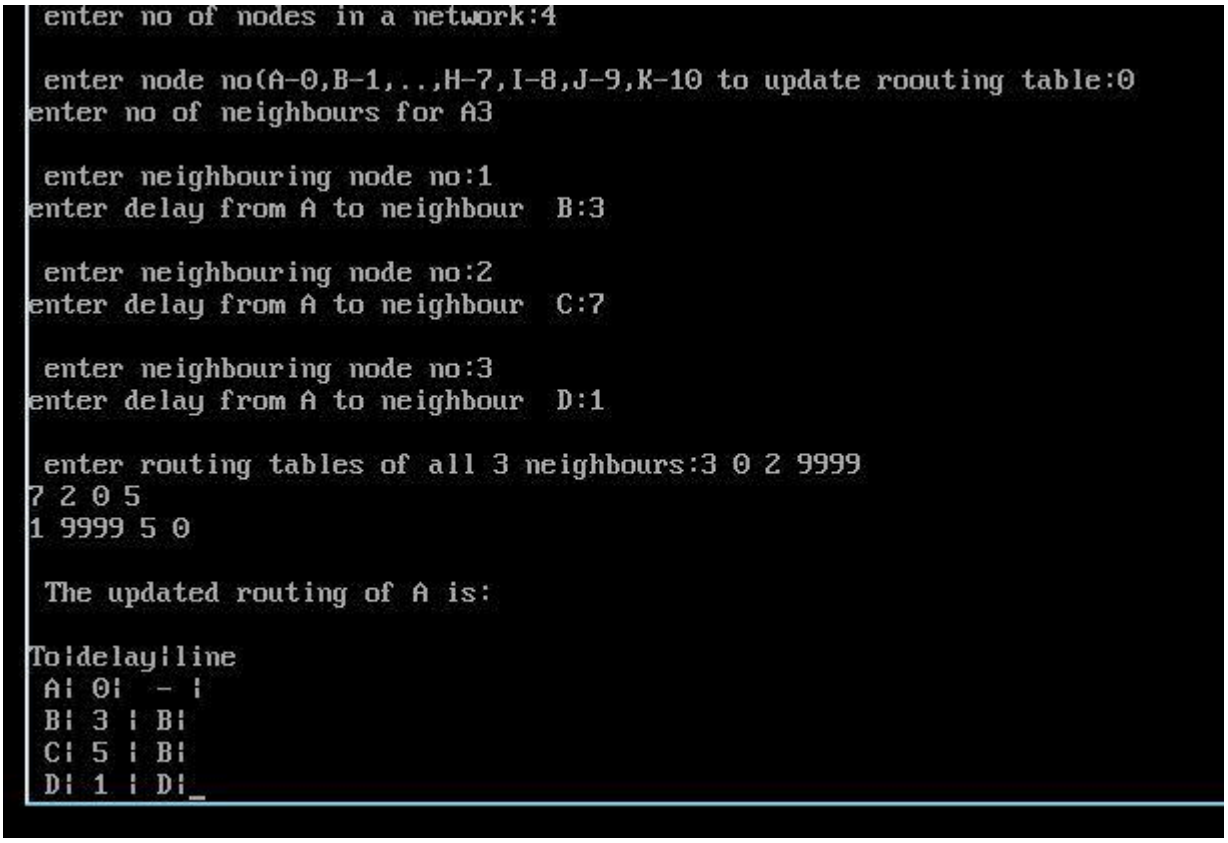
SOURCE CODE:

```
//DISTANCE VECTOR ROUTING ALGORITHM
#include<stdio.h>
void main()
{
  int n,s,nn[10],delay[10],m,rt[20][30],i,j,min,node;
  clrscr();
```

```
  printf("\n enter no of nodes in a network:");
  scanf("%d",&n);
  printf("\n enter node no(A-0,B-1,..,H-7,I-8,J-9,K-10 to update roouting table:");
  scanf("%d",&s);
  printf("\enter no of neighbours for %c",s+65);
  scanf("%d",&m);
  for(i=0;i<m;i++)
  {
    printf("\n enter neighbouring node no:");
    scanf("%d",&nn[i]);
   printf("\enter delay from %c to neighbour  %c:",s+65,nn[i]+65);
  scanf("%d",&delay[i]);
  }
  printf("\n enter routing tables of all %d neighbours:",m);

  for(i=0;i<m;i++)
  {
    for(j=0;j<n;j++)
    {
                scanf("%d",&rt[i][j]);
                rt[i][j]+=delay[i];
    }
  }
  // find neighbour with min delay from all its neighbour to other nodes
  printf("\n The updated routing of %c is:\n",s+65);
printf("\nTo|delay|line");
  for(i=0;i<n;i++)
  {
     min=32767;
     for(j=0;j<m;j++)
     {
                 if(rt[j][i] <min)
                 {
                    min=rt[j][i];
                    node=nn[j];
                 }
     }
     if(s!=i)
     printf("\n %c| %d | %c|",i+65,min,node+65);
               else
                printf("\n %c| 0|  - |",s+65);
  }
```

```
  getch();
}
```

OUTPUT:



VIVA QUESTIONS:

1. What is routing
2. What is best algorithm among all routing algorithms?
3. What is static routing?
4. Difference between static and dynamic
5. How distance vector routing works
6. What is optimality principle?

7. Implement data encryption and data decryption.

NAME OF THE EXPERIMENT: Encryption and Decryption of data

AIM: Implement Data Encryption and Data Decryption.

HARDWARE REQUIREMENTS: Intel based Desktop PC:- RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

Data encryption standard was widely adopted by the industry in security products. Plain text is encrypted in blocks of 64 bits yielding 64 bits of cipher text. The algorithm which is parameterized by a 56 bit key has 19 distinct stages. The first stage is a key independent transposition and the last stage is exactly inverse of the transposition. The remaining stages are functionally identical but are parameterized by different functions of the key. The algorithm has been designed to allow decryption to be done with the same key as encryption

SOURCE CODE:

```
#include<stdio.h>
void main()
{
  char pt[100],ct[100];
  int i,key;
  clrscr();
  printf("\n enter plain text:");
  gets(pt);
  printf("\n enter key");
  scanf("%d",&key);
  printf("\n The plain text is:%s",pt);
  for(i=0;pt[i]!='\0';i++)
  ct[i]=pt[i]+key;
  printf("\n After encryption,The cipher text is:%s",ct);
  for(i=0;pt[i]!='\0';i++)
  pt[i]=ct[i]-key;
  printf("\n After decryption,The plain text is:%s",pt);
```

```
        getch();
}
```

OUTPUT:

enter plain text:hai
enter key1
The plain text is:hai
After encryption,The cipher text is:ibj
After decryption,The plain text is:hai

VIVA QUESTIONS:

1. Expand DES_____
2. What is cipher text?
3. What is plain text?
4. Define public key?
5. Define encryption?
6. Substitutions are performed by_____boxes

8. Write a program for congestion control using Leaky bucket algorithm.

NAME OF THE EXPERIMENT: Leaky bucket Algorithm

AIM: Write a program for congestion control using Leaky bucket algorithm.

HARDWARE REQUIREMENTS: Intel based Desktop PC:- RAM of 512 MB

SOFTWARE REQUIREMENTS: Turbo C / Borland C.

THEORY:

The congesting control algorithms are basically divided into two groups: open loop and closed loop. Open loop solutions attempt to solve the problem by good design, in essence, to make sure it does not occur in the first place. Once the system is up and running, midcourse corrections are not made. Open loop algorithms are further divided into ones that act at source versus ones that act at the destination.

In contrast, closed loop solutions are based on the concept of a feedback loop if there is any congestion. Closed loop algorithms are also divided into two sub categories: explicit feedback and implicit feedback. In explicit feedback algorithms, packets are sent back from the point of congestion to warn the source. In implicit algorithm, the source deduces the existence of congestion by making local observation, such as the time needed for acknowledgment to come back.

The presence of congestion means that the load is (temporarily) greater than the resources (in part of the system) can handle. For subnets that use virtual circuits internally, these methods can be used at the network layer.

Another open loop method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This approach to congestion management is widely used in ATM networks and is called traffic shaping.

The other method is the leaky bucket algorithm. Each host is connected to the network by an interface containing a leaky bucket, that is, a finite internal queue. If a packet arrives at the queue when it is full, the packet is discarded. In other words, if one or more process are already queued, the new packet is unceremoniously discarded. This arrangement can be built into the hardware interface or simulate d by the host operating system. In fact it is nothing other than a single server queuing system with constant service time.

The host is allowed to put one packet per clock tick onto the network. This mechanism turns an uneven flow of packet from the user process inside the host into an even flow of packet onto the network, smoothing out bursts and greatly reducing the chances of congestion.

SOURCE CODE:

```
#include<stdio.h>
void main()
{
   int bucket_size,incoming_rate,outgoing_rate,stored=0;
   clrscr();
   printf("enter bucket size and outgoing rate:\n");
```

```
   scanf("%d%d",&bucket_size,&outgoing_rate);
   do
   {
     printf(" enter incoming rate\n");
     scanf("%d",&incoming_rate);
     if(incoming_rate>bucket_size)
     {
               printf(" %d packets dropped\n",incoming_rate-bucket_size-stored);
               stored=bucket_size;
               printf(" bucket has %d packets out of its capacity %d
packets\n",stored,bucket_size);
               stored=stored-outgoing_rate;
               printf(" after outgoing %d packets,bucket has %d
packets\n",outgoing_rate,stored);
     }
     else
     {
                stored=stored+incoming_rate;
                printf(" bucket has %d packets out of its capacity %d
packets\n",stored,bucket_size);
                stored=stored-outgoing_rate;
                if(stored>0)
       printf(" after outgoing %d packets,bucket has %d packets\n",outgoing_rate,stored);
         else
                 printf("after outgoing, no packets in the bucket\n");
    }
  }while(stored>0);
getch();
}
```

OUTPUT:

```
enter bucket size and outgoing rate:
15
10
 enter incoming rate
30
 15 packets dropped
 bucket has 15 packets out of its capacity 15 packets
 after outgoing 10 packets,bucket has 5 packets
 enter incoming rate
10
 bucket has 15 packets out of its capacity 15 packets
 after outgoing 10 packets,bucket has 5 packets
 enter incoming rate
5
 bucket has 10 packets out of its capacity 15 packets
after outgoing, no packets in the bucket
```

9. Write a program for frame sorting technique used in buffers.

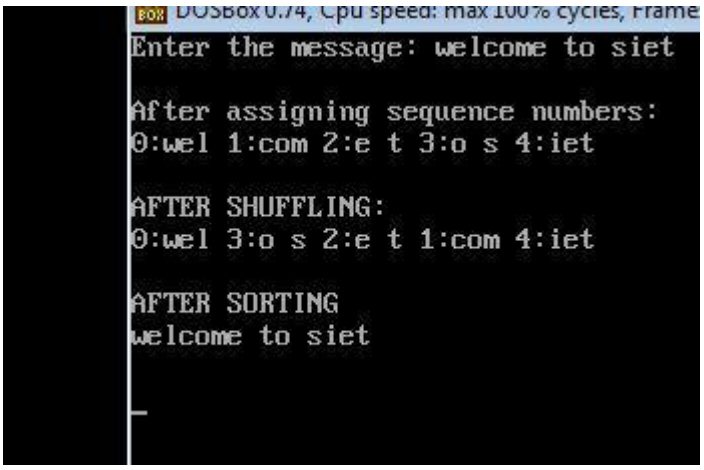
SOURCE CODE:

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define FRAM_TXT_SIZ 4
#define MAX_NOF_FRAM 127
char str[FRAM_TXT_SIZ*MAX_NOF_FRAM];
struct frame // structure maintained to hold frames
{
 int seq_no;
 char text[FRAM_TXT_SIZ];
//int seq_no;
}fr[MAX_NOF_FRAM], shuf_ary[MAX_NOF_FRAM];
int assign_seq_no() //function which splits message
{
int k=0,i,j; //into frames and assigns sequence no
for(i=0; i < strlen(str); k++)
{
fr[k].seq_no = k;
for(j=0; j < FRAM_TXT_SIZ-1 && str[i]!='\0'; j++)
fr[k].text[j] = str[i++];
fr[k].text[j]='\0';
}
printf("\nAfter assigning sequence numbers:\n");
```

```
for(i=0; i < k; i++)
printf("%d:%s ",fr[i].seq_no,fr[i].text);
return k; //k gives no of frames
}
void generate(int *random_ary, const int limit) //generate array of random nos
{ int r, i=0, j;
while(i < limit)
{
r = random(10) % limit;
//printf("\n r=%d,i=%d",r,i);
for(j=0; j < i; j++)
if( random_ary[j] == r )
break;
if( i==j ) random_ary[i++] = r;
} }
void shuffle(const  int no_frames ) // function shuffles the frames
{
int i, random_ary[70];
generate(random_ary, no_frames);
for(i=0; i < no_frames; i++)
shuf_ary[i] = fr[random_ary[i]];
printf("\n\nAFTER SHUFFLING:\n");
for(i=0; i < no_frames; i++)
printf("%d:%s ",shuf_ary[i].seq_no,shuf_ary[i].text);
}
void sort(const int no_frames) // sorts the frames
{
int i,j,flag=1;
struct frame hold;
for(i=0; i < no_frames-1 && flag==1; i++) // search for frames in sequence
{
flag=0;
for(j=0; j < no_frames-1-i; j++) //(based on seq no.) and display
if(shuf_ary[j].seq_no > shuf_ary[j+1].seq_no)
{
hold = shuf_ary[j];
shuf_ary[j] = shuf_ary[j+1];
shuf_ary[j+1] = hold;
flag=1;
}
}
}
```

```
int main()
{
int no_frames,i;
clrscr();
printf("Enter the message: ");
gets(str);
no_frames = assign_seq_no();
shuffle(no_frames);
sort(no_frames);
printf("\n\nAFTER SORTING\n");
for(i=0;i<no_frames;i++)
printf("%s",shuf_ary[i].text);
printf("\n\n");
getch();
return 0;
}
```

OUTPUT:



10.Wireshark

i. Packet Capture Using Wire shark
ii. Starting Wire shark
iii. Viewing Captured Traffic
iv. Analysis and Statistics & Filters.

What Is Wireshark?

Originally known as Ethereal, Wireshark displays data from hundreds of different protocols on all major network types. Data packets can be viewed in real-time or analyzed offline. Wireshark supports dozens of capture/trace file formats, including CAP and ERF. Integrated decryption tools display the encrypted packets for several common protocols, including WEP and WPA/WPA2.

How to Download and Install Wireshark

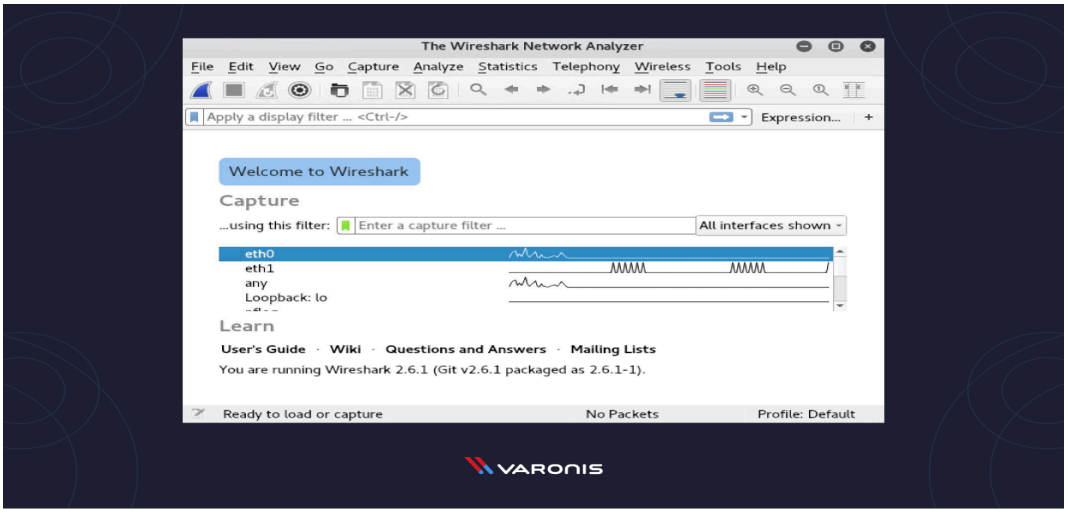
Wireshark can be downloaded at no cost from the Wireshark Foundation website for both macOS and Windows. You'll see the latest stable release and the current developmental release. Unless you're an advanced user, download the stable version.

Data Packets on Wireshark

Now that we have Wireshark installed let's go over how to enable the Wireshark packet sniffer and then analyze the network traffic.

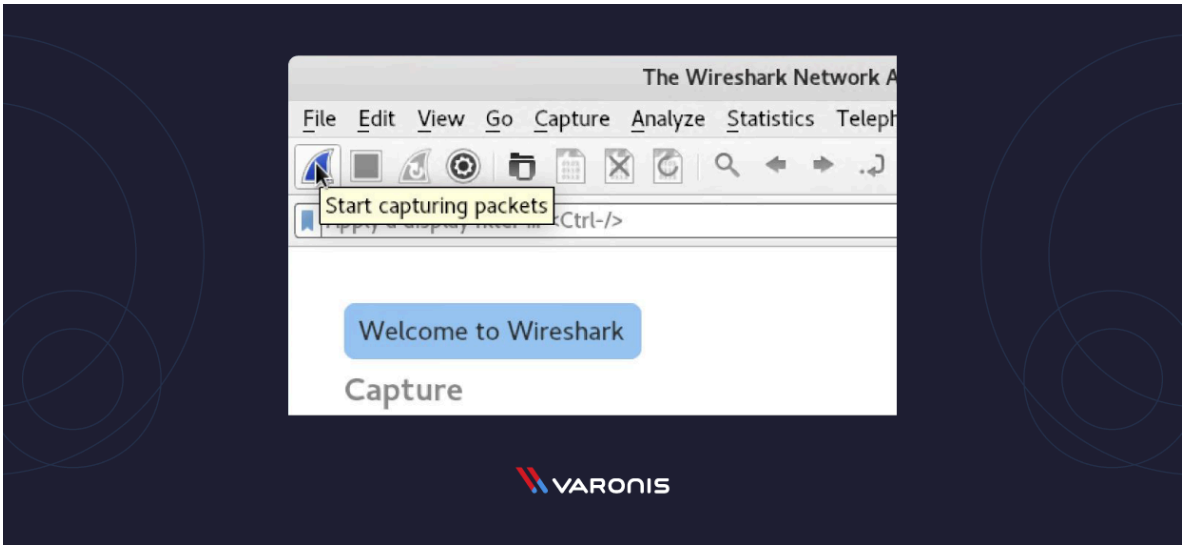
Capturing Data Packets on Wireshark

When you open Wireshark, you see a screen that shows you a list of all of the network connections you can monitor. You also have a capture filter field, so you only capture the network traffic you want to see.

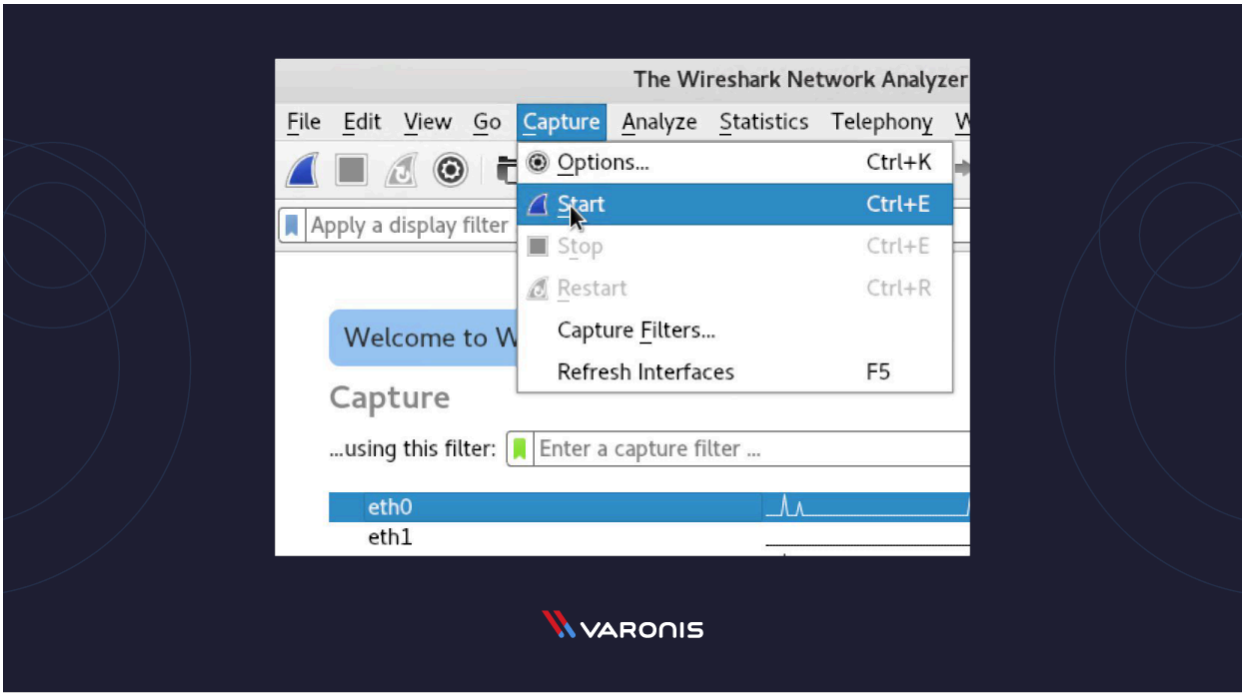


You can select one or more of the network interfaces using "shift left-click." Once you have the network interface selected, you can start the capture, and there are several ways to do that.

Click the first button on the toolbar, titled "Start Capturing Packets."

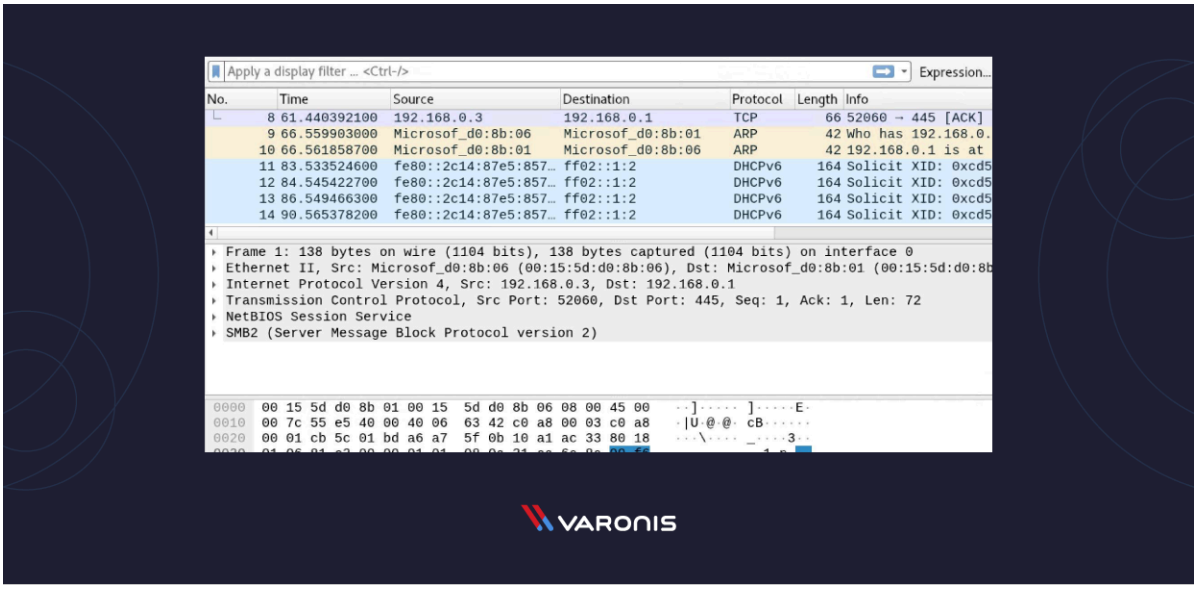


You can select the menu item Capture -> Start.



Or you could use the keystroke Control – E.

During the capture, Wireshark will show you the packets that it captures in real-time.



Once you have captured all the packets you need, you use the same buttons or menu options to stop the capture.

Best practice says that you should stop Wireshark packet capture before you do analysis.

Analyzing Data Packets on Wireshark

Wireshark shows you three different panes for inspecting packet data. The Packet List, the top pane, is a list of all the packets in the capture. When you click on a packet, the other two panes change to show you the details about the selected packet. You can also tell if the packet is part of a conversation. Here are some details about each column in the top pane:

- **No.**: This is the number order of the packet that got captured. The bracket indicates that this packet is part of a conversation.
- **Time**: This column shows you how long after you started the capture that this packet got captured. You can change this value in the Settings menu if you need something different displayed.
- **Source**: This is the address of the system that sent the packet.
- **Destination**: This is the address of the destination of that packet.
- **Protocol**: This is the type of packet, for example, TCP, DNS, DHCPv6, or ARP.
- **Length**: This column shows you the length of the packet in bytes.
- **Info**: This column shows you more information about the packet contents, and will vary depending on what kind of packet it is.

Packet Details, the middle pane, shows you as much readable information about the packet as possible, depending on what kind of packet it is. You can right-click and create filters based on the highlighted text in this field.

The bottom pane, Packet Bytes, displays the packet exactly as it got captured in hexadecimal.

When you are looking at a packet that is part of a conversation, you can right-click the packet and select Follow to see only the packets that are part of that conversation.

Wireshark Filters

One of the best features of Wireshark is the Wireshark Capture Filters and Wireshark Display Filters. Filters allow you to view the capture the way you need to see it so you can troubleshoot the issues at hand. Here are several filters to get you started.

Wireshark Capture Filters

Capture filters limit the captured packets by the filter. Meaning if the packets don't match the filter, Wireshark won't save them. Here are some examples of capture filters:

host IP-*address*: this filter limits the capture to traffic to and from the IP address

net 192.168.0.0/24: this filter captures all traffic on the subnet.

dst host IP-*address*: capture packets sent to the specified host.

port 53: capture traffic on port 53 only.

port not 53 and not arp: capture all traffic except DNS and ARP traffic

Wireshark Display Filters

Wireshark Display Filters change the view of the capture during analysis. After you have stopped the packet capture, you use display filters to narrow down the packets in the Packet List so you can troubleshoot your issue.

The most useful (in my experience) display filter is:

ip.src==*IP-address* and ip.dst==*IP-address*

This filter shows you packets from one computer (ip.src) to another (ip.dst). You can also use ip.addr to show you packets to and from that IP. Here are some others:

tcp.port eq 25: This filter will show you all traffic on port 25, which is usually SMTP traffic.

icmp: This filter will show you only ICMP traffic in the capture, most likely they are pings.

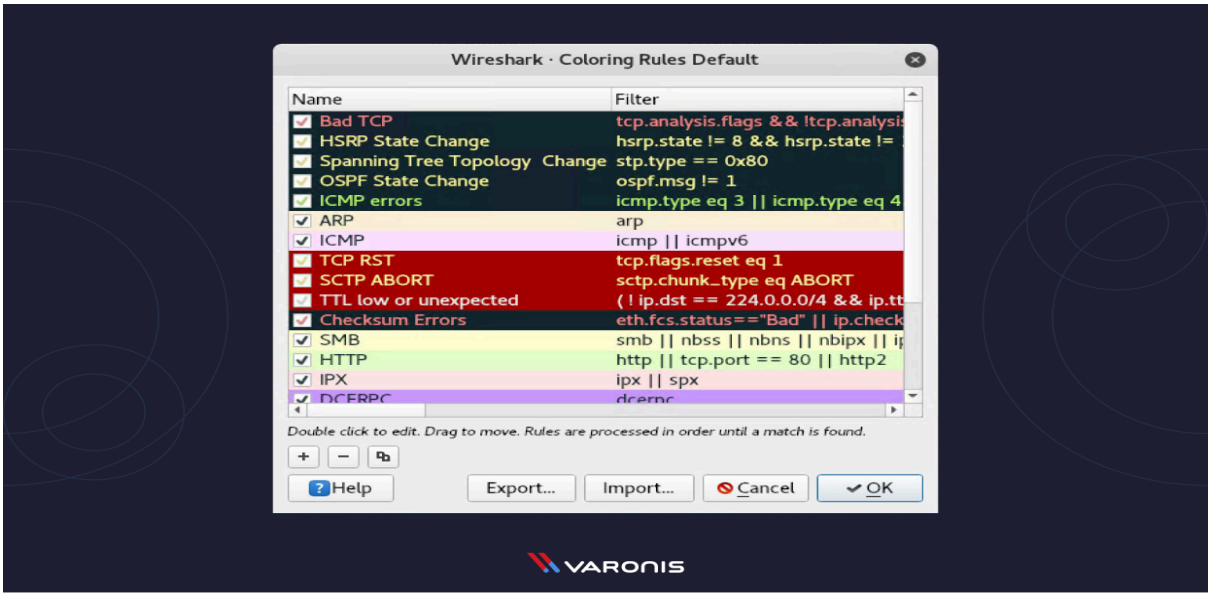
ip.addr != *IP_address*: This filter shows you all traffic except the traffic to or from the specified computer.

Analysts even build filters to detect specific attacks, like this filter to detect the Sasser worm:

ls_ads.opnum==0x09

Wireshark Colorization Options

You can setup Wireshark so it colors your packets in the Packet List according to the display filter, which allows you to emphasize the packets you want to highlight.



Wireshark Promiscuous Mode

By default, Wireshark only captures packets going to and from the computer where it runs. By checking the box to run Wireshark in Promiscuous Mode in the Capture Settings, you can capture most of the traffic on the LAN.

Wireshark Command Line

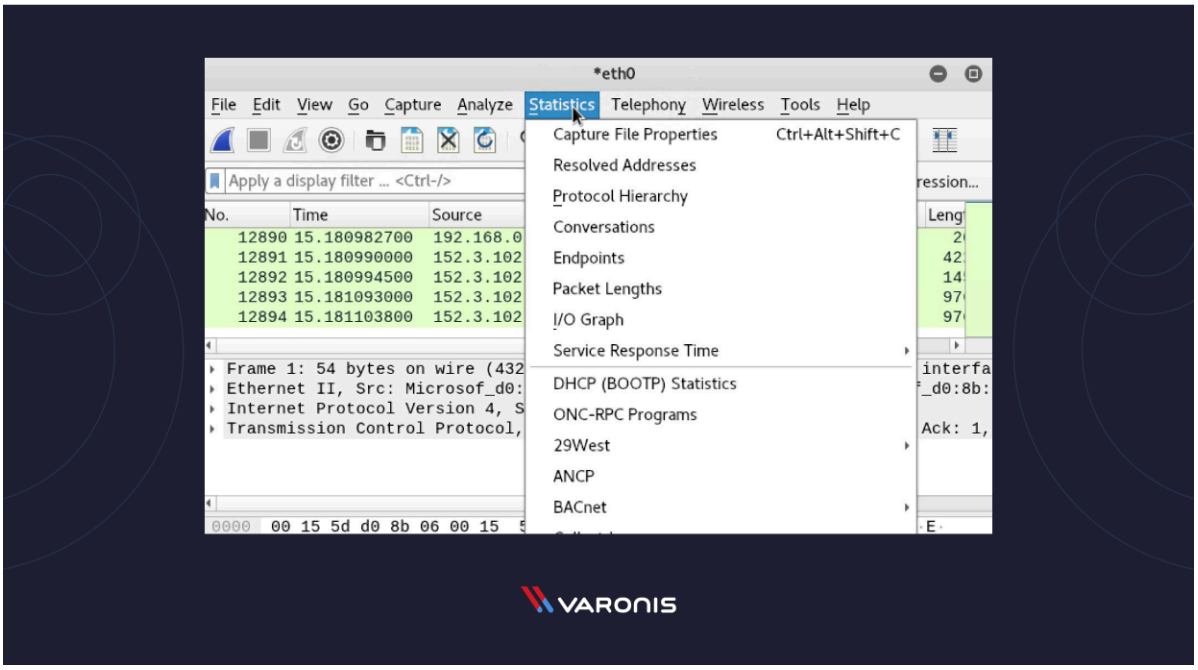
Wireshark does provide a Command Line Interface (CLI) if you operate a system without a GUI. Best practice would be to use the CLI to capture and save a log so you can review the log with the GUI.

Wireshark Commands

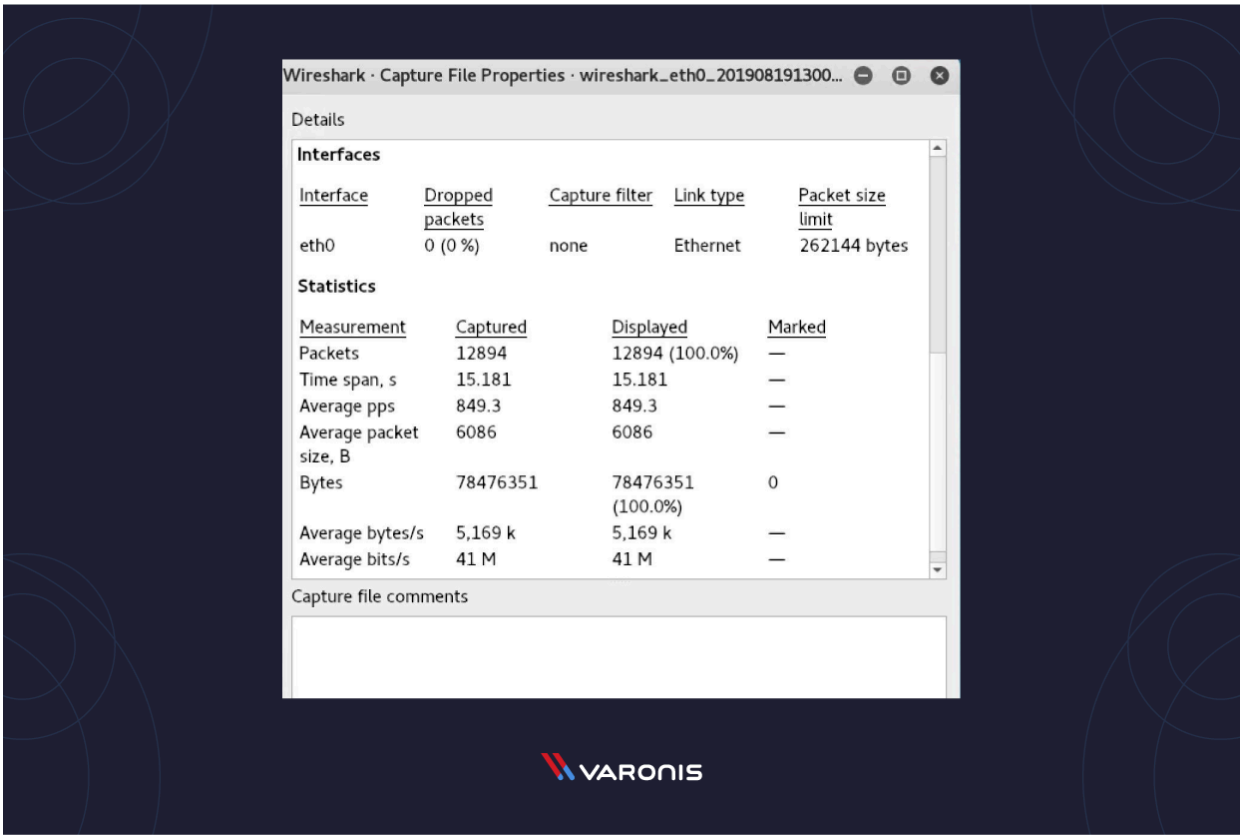
- wireshark : run Wireshark in GUI mode
- wireshark –h : show available command line parameters for Wireshark
- wireshark –a duration:300 –i eth1 –w wireshark. : capture traffic on the Ethernet interface 1 for 5 minutes. –a means automatically stop the capture, -i specifics which interface to capture

Metrics and Statistics

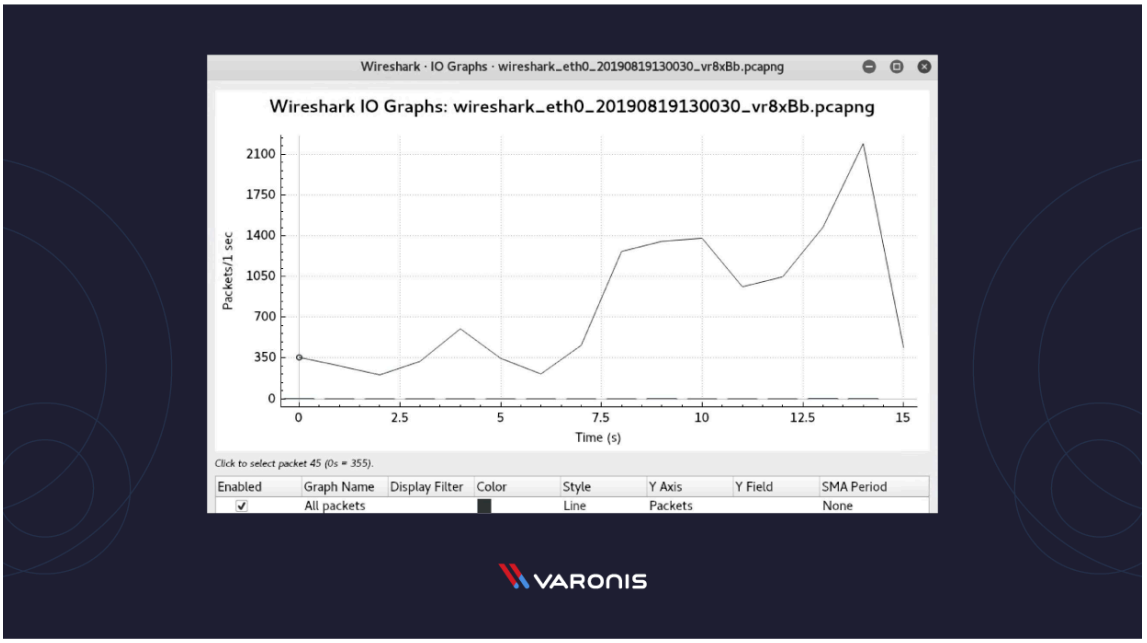
Under the Statistics menu item, you will find a plethora of options to show details about your capture.



Capture File Properties:



Wireshark I/O Graph:


Wireshark · IO Graphs · wireshark_eth0_20190819130030_vr8xBb.pcapng
Wireshark IO Graphs: wireshark_eth0_20190819130030_vr8xBb.pcapng
Packets/1 sec
2100
1750
1400
1050
700
350
0
0
2.5
5
7.5
10
12.5
15
Time (s)
Click to select packet 45 (0s = 355).
Enabled
Graph Name
Display Filter
Color
Style
Y Axis
Y Field
SMA Period
All packets
Line
Packets
None
VARONIS

11. How to run Nmap scan

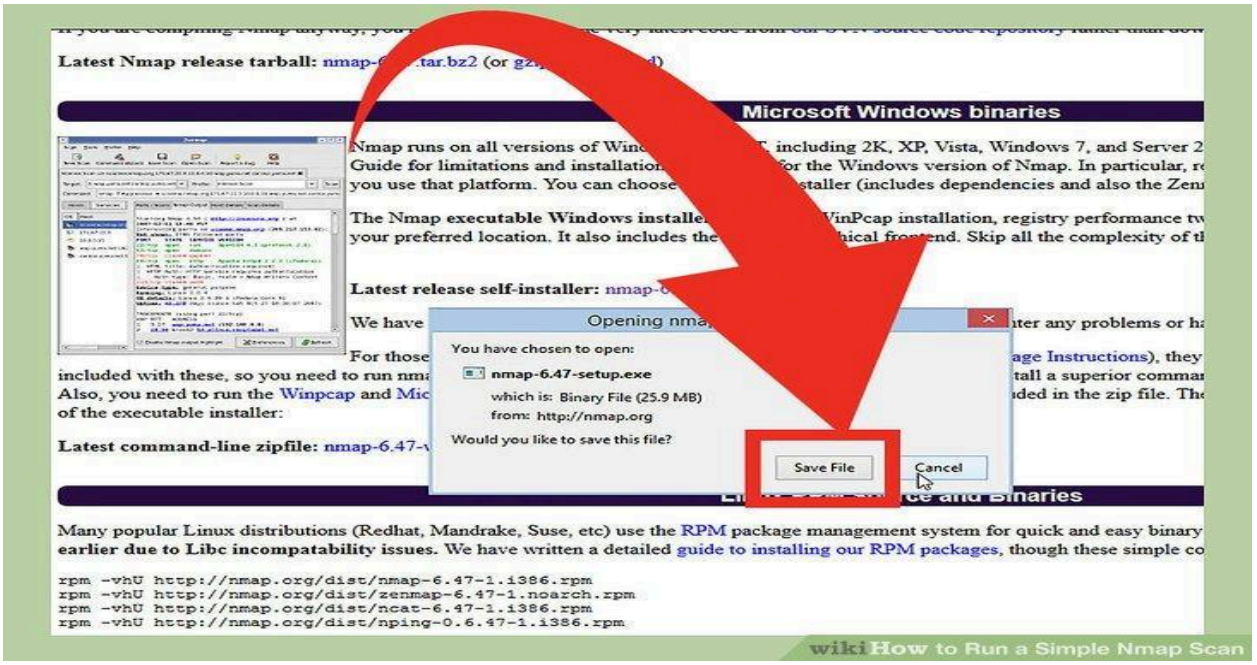
HOW TO RUN A SIMPLE NMAP SCAN

Are you worried about the security of your network or the security of someone else's? Ensuring that your router is protected from unwanted intruders is one of the foundations of a secure network. One of the basic tools for this job is Nmap, or Network Mapper. This program will scan a target and report which ports are open and which are closed, among other things. Security specialists use this program to test the security of a network. To learn how to use it yourself, see Step 1 below.

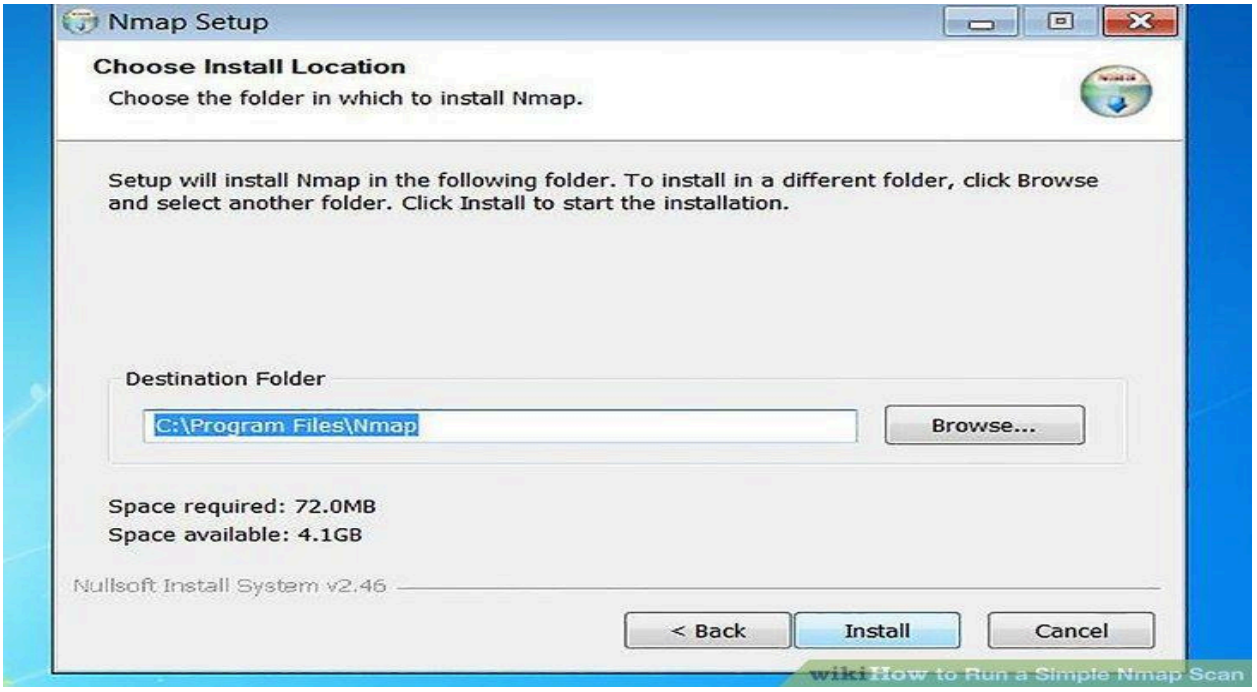
Step 1: Download the Nmap installer. This can be found for free from the developer's website. It is highly recommended that you download directly from the developer to avoid any potential viruses or fake files. Downloading the Nmap installer includes Zenmap, the graphical interface for Nmap which makes it easy for newcomers to perform scans without having to learn command lines.

- The Zenmap program is available for Windows, Linux, and Mac OS X. You can find the installation files for all operating systems on the Nmap website.

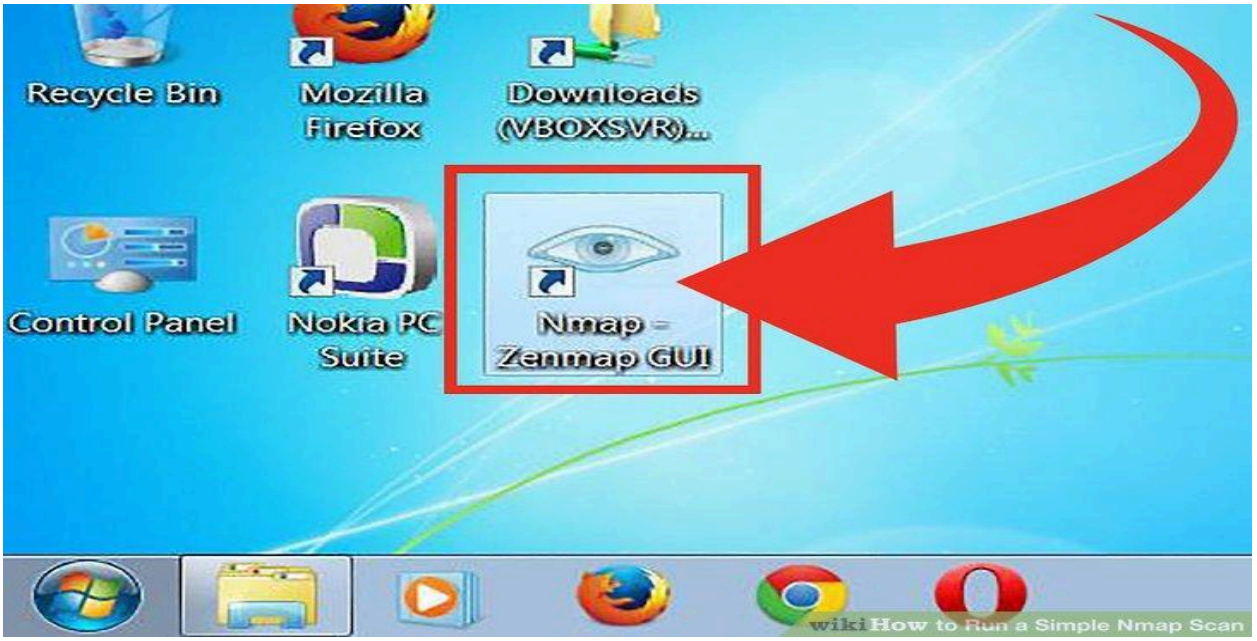
2.



Step 2:Install Nmap. Run the installer once it is finished downloading. You will be asked which components you would like to install. In order to get the full benefit of Nmap, keep all of these checked. Nmap will not install any adware or spyware.

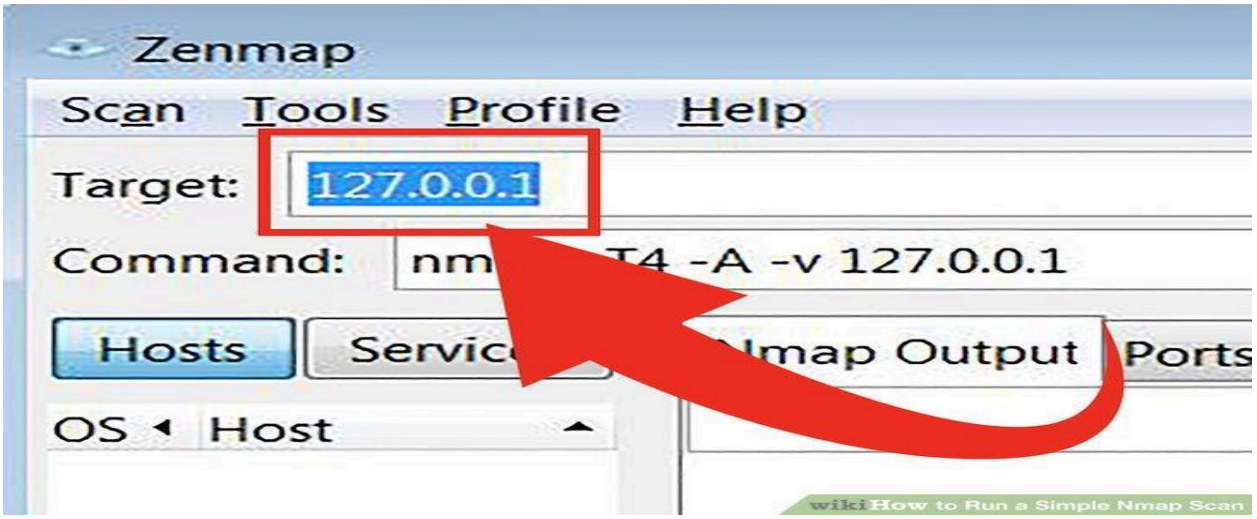


Step 3:Run the "Nmap – Zenmap" GUI program. If you left your settings at default during installation, you should be able to see an icon for it on your desktop. If not, look in your Start menu. Opening Zenmap will start the program.



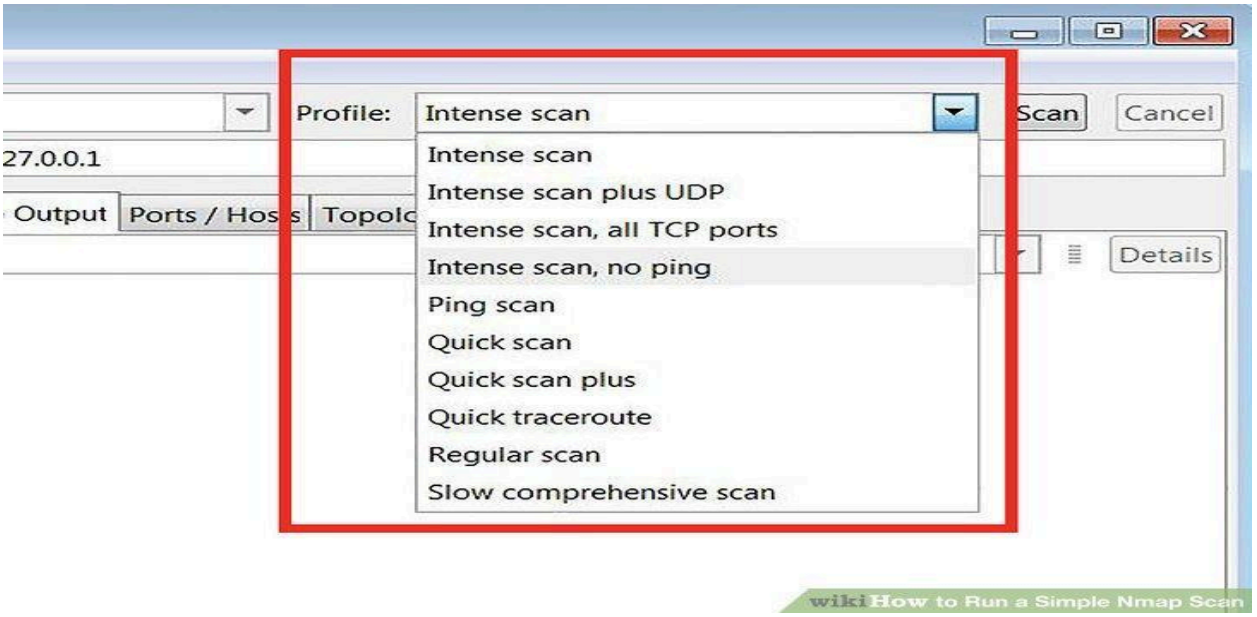
Step 4:Enter in the target for your scan. The Zenmap program makes scanning a fairly simple process. The first step to running a scan is choosing your target. You can enter a domain (example.com), an IP address (127.0.0.1), a network (192.168.1.0/24), or a combination of those.

- Depending on the intensity and target of your scan, running an Nmap scan may be against the terms of your internet service provider, and may land you in hot water. Always check your local laws and your ISP contract before performing Nmap scans on targets other than your own network.

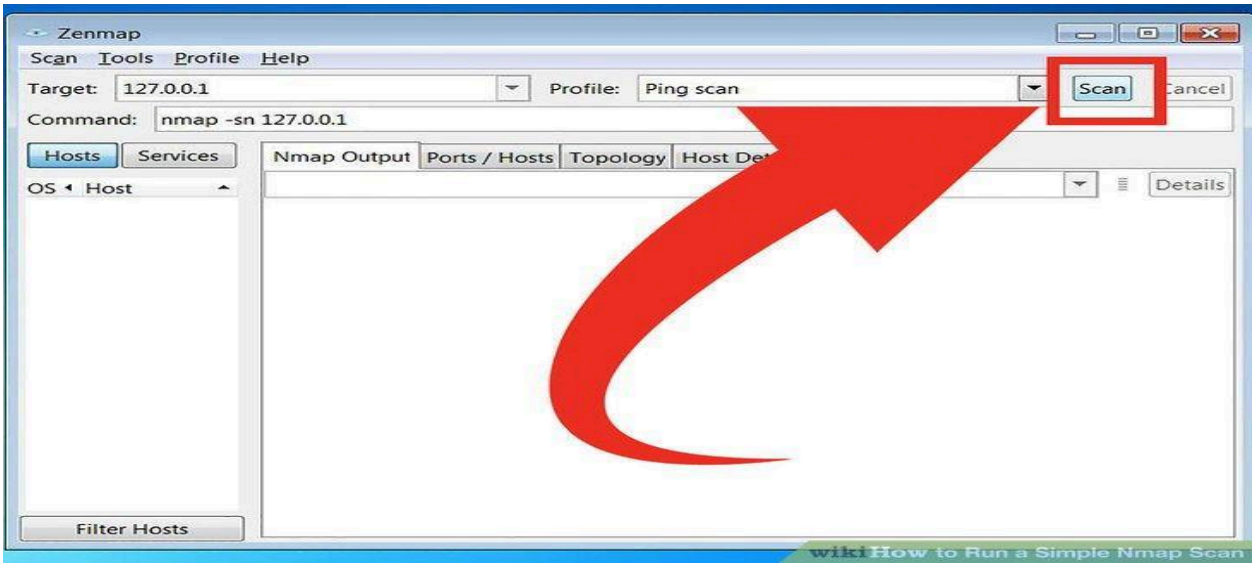


Step 5:Choose your Profile. Profiles are preset groupings of modifiers that change what is scanned. The profiles allow you to quickly select different types of scans without having to type in the modifiers on the command line. Choose the profile that best fits your needs:[1]

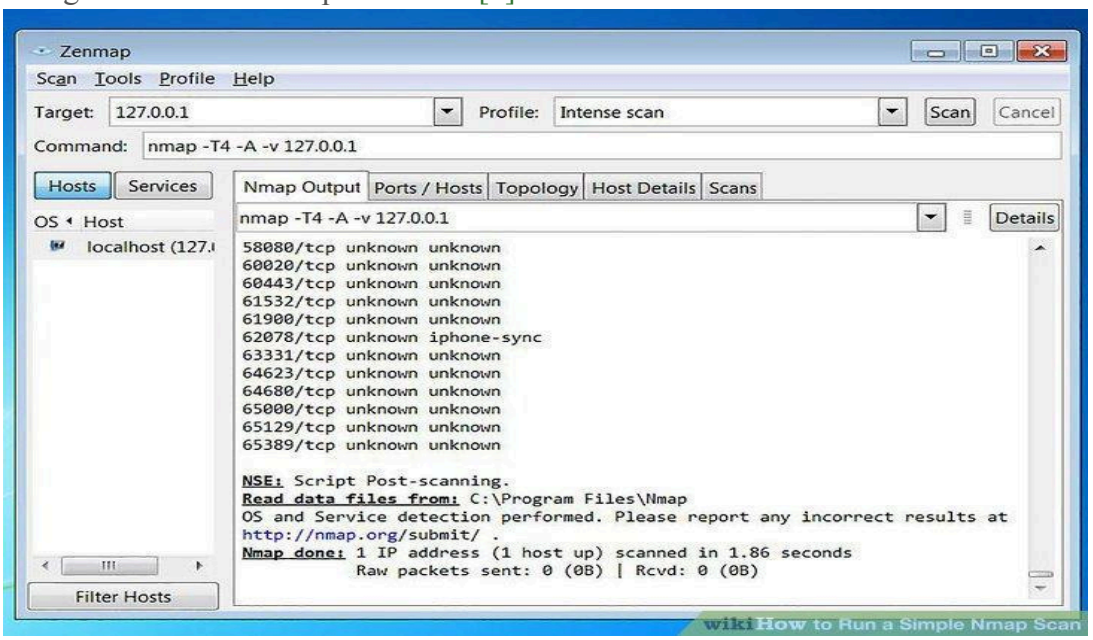
- **Intense scan** - A comprehensive scan. Contains Operating System (OS) detection, version detection, script scanning, traceroute, and has aggressive scan timing. This is considered an intrusive scan.
- **Ping scan** - This scan simply detects if the targets are online, it does not scan any ports.
- **Quick scan** - This is quicker than a regular scan due to aggressive timing and only scanning select ports.
- **Regular scan** - This is the standard Nmap scan without any modifiers. It will return ping and return open ports on the target.



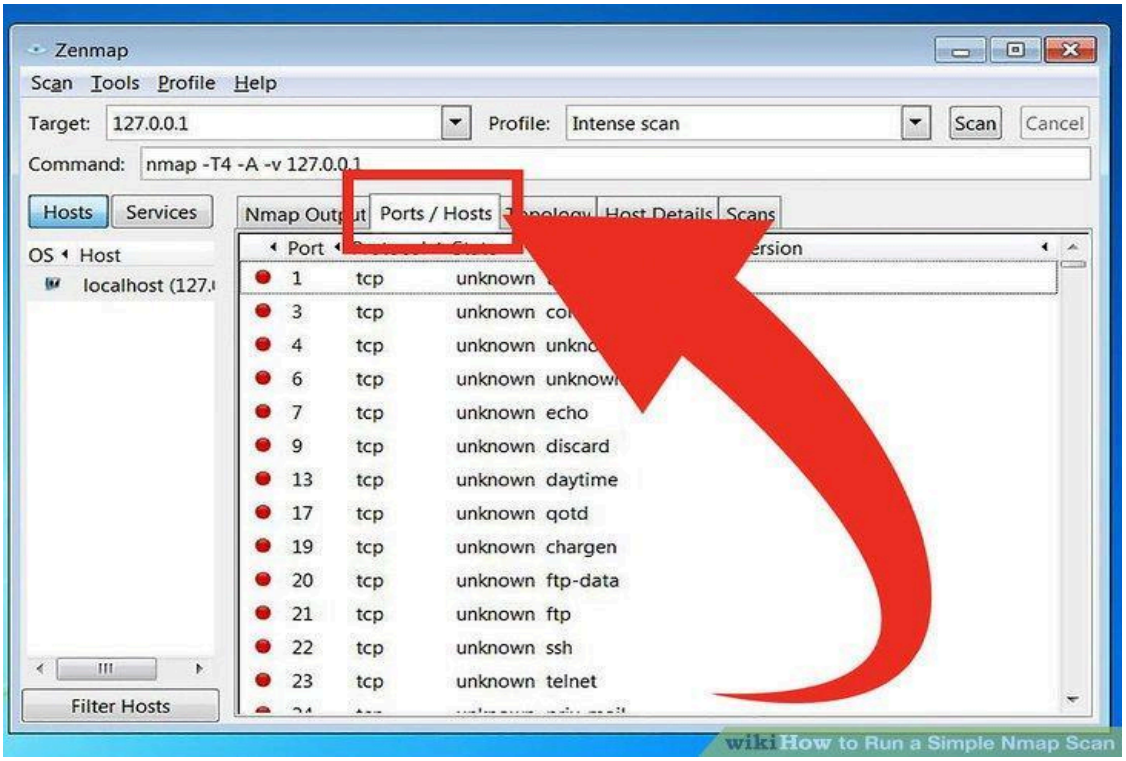
Step 6:Click Scan to start scanning. The active results of the scan will be displayed in the Nmap Output tab. The time the scan takes will depend on the scan profile you chose, the physical distance to the target, and the target's network configuration.



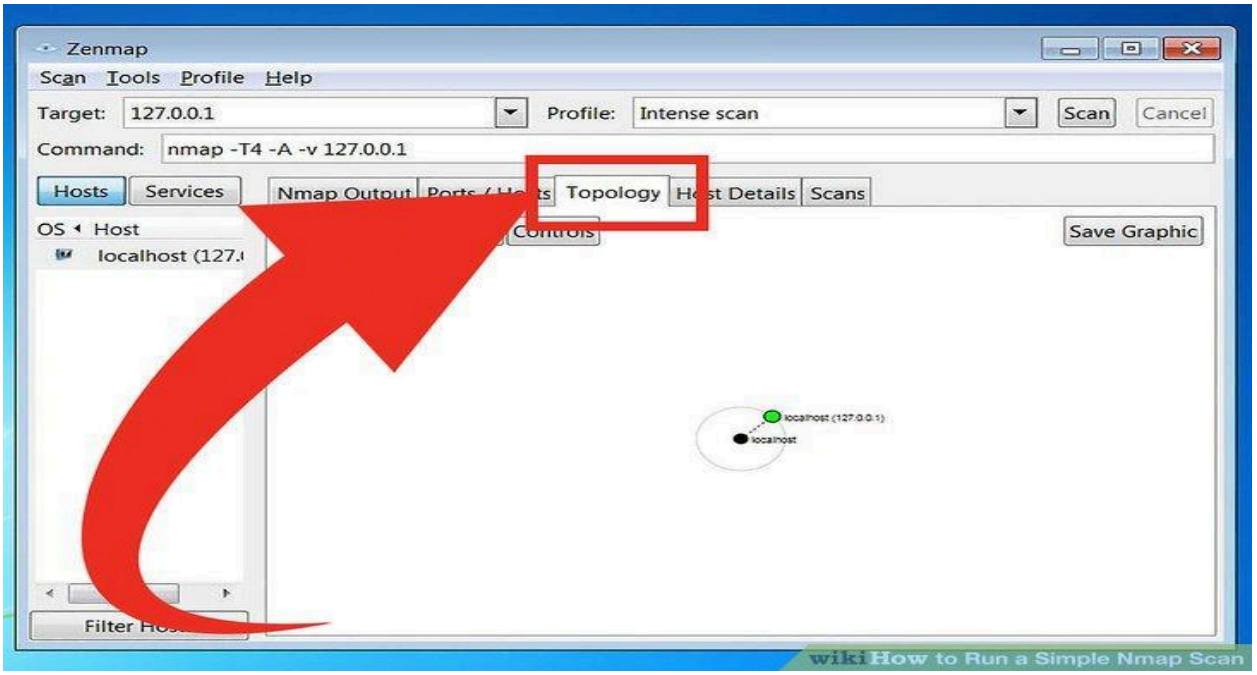
Step 7:Read your results. Once the scan is finished, you'll see the message "Nmap done" at the bottom of the Nmap Output tab. You can now check your results, depending on the type of scan you performed. All of the results will be listed in the main Nmap Output tab, but you can use the other tabs to get a better look at specific data.[2]



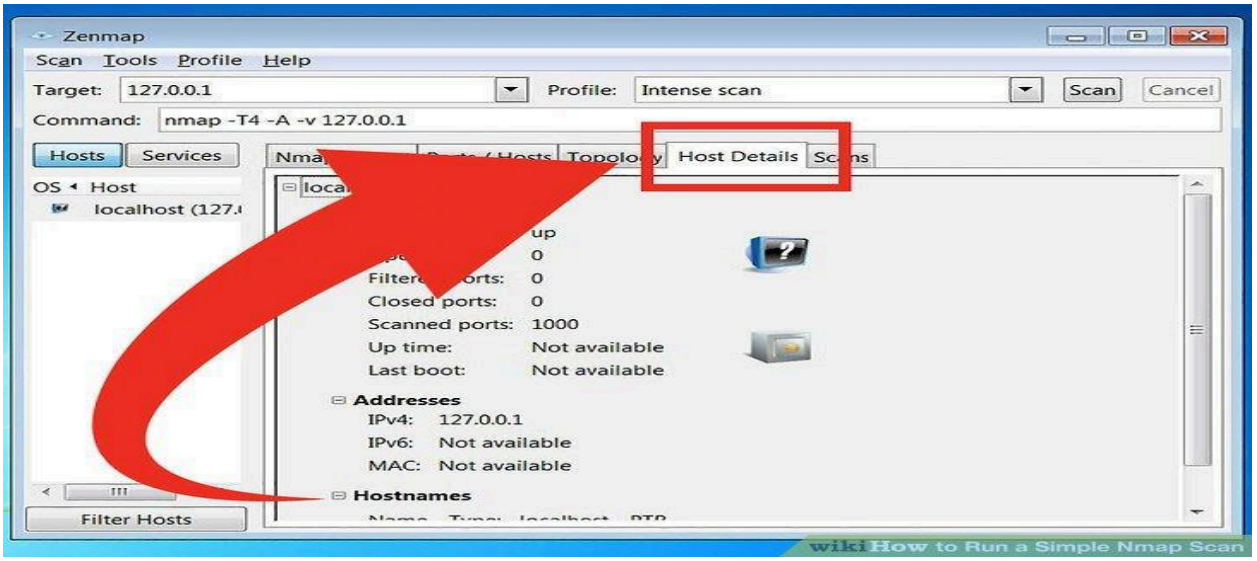
7.1:Ports/Hosts - This tab will show the results of your port scan, including the services for those ports.



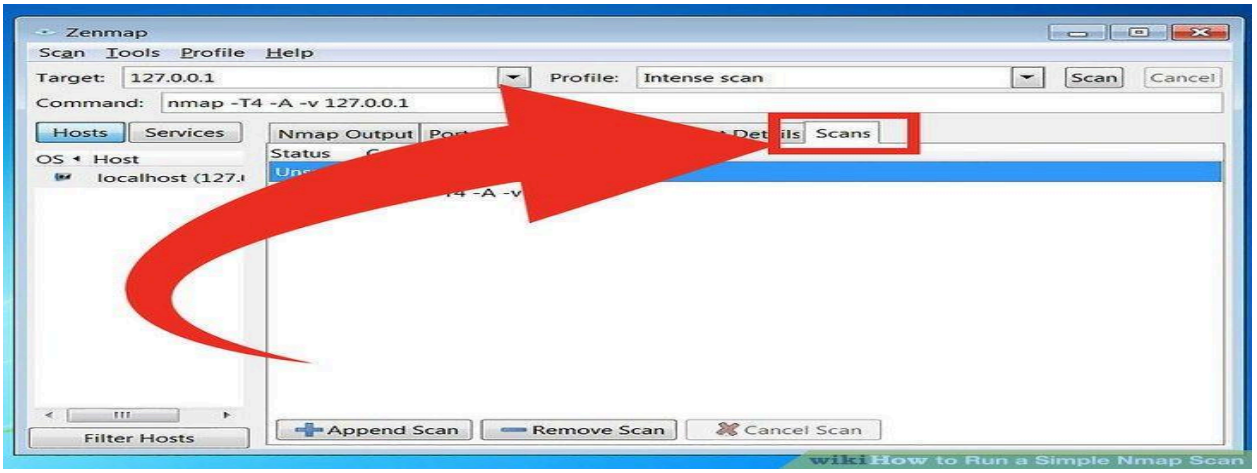
7.2:Topology - This shows the traceroute for the scan you performed. You can see how many hops your data goes through to reach the target.



7.3:Host Details - This shows a summary of your target learned through scans, such as the number of ports, IP addresses, hostnames, operating systems, and more.



7.4:Scans - This tab stores the commands of your previously-run scans. This allows you to quickly re-scan with a specific set of parameters.



12. Operating System Detection using Nmap.

Operating System Detection using Nmap

One of Nmap's best-known features is remote OS detection using TCP/IP stack fingerprinting. Nmap sends a series of TCP and UDP packets to the remote host and examines practically every bit in the responses. After performing dozens of tests such as TCP ISN sampling, TCP options support and ordering, IP ID sampling, and the initial window size check, Nmap compares the results to its nmap-os-db database of more than 2,600 known OS fingerprints and prints out the OS details if there is a match. Each fingerprint includes a freeform textual description of the OS, and a classification which provides the vendor name (e.g. Sun), underlying OS (e.g. Solaris), OS generation (e.g. 10), and device type (general purpose, router, switch, game console, etc).

OS detection is enabled and controlled with the following options:

-O (Enable OS detection)

Enables OS detection, as discussed above. Alternatively, you can use -A to enable OS detection along with other things.

--osscan-limit (Limit OS detection to promising targets)

OS detection is far more effective if at least one open and one closed TCP port are found. Set this option and Nmap will not even try OS detection against hosts that do not meet this criteria. This can save substantial time, particularly on -Pn scans against many hosts. It only matters when OS detection is requested with -O or -A.

--osscan-guess; --fuzzy (Guess OS detection results)

When Nmap is unable to detect a perfect OS match, it sometimes offers up near-matches as possibilities. The match has to be very close for Nmap to do this by default. Either of these (equivalent) options make Nmap guess more aggressively. Nmap will still tell you when an imperfect match is printed and display its confidence level (percentage) for each guess.

--max-os-tries (Set the maximum number of OS detection tries against a target)

When Nmap performs OS detection against a target and fails to find a perfect match, it usually repeats the attempt. By default, Nmap tries five times if conditions are favorable for OS fingerprint submission, and twice when conditions aren't so good. Specifying a lower --max-os-tries value (such as 1) speeds Nmap up, though you miss out on retries which could potentially identify the OS. Alternatively, a high value may be set to allow

even more retries when conditions are favorable. This is rarely done, except to generate better fingerprints for submission and integration into the Nmap OS database.

The inner workings of OS detection are quite complex, but it is one of the easiest features to use. Simply add -O to your scan options. You may want to also increase the verbosity with -v for even more OS-related details. This is shown in Example 8.1.

Example 8.1. OS detection with verbosity (-O -v)

```
# nmap -O -v scanme.nmap.org

Starting Nmap ( http://nmap.org )
Nmap scan report for scanme.nmap.org (74.207.244.221)
Not shown: 994 closed ports
PORT      STATE    SERVICE
22/tcp    open     ssh
80/tcp    open     http
646/tcp   filtered ldp
1720/tcp  filtered H.323/Q.931
9929/tcp  open     nping-echo
31337/tcp open     Elite
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.39
OS details: Linux 2.6.39
Uptime guess: 1.674 days (since Fri Sep  9 12:03:04 2011)
Network Distance: 10 hops
TCP Sequence Prediction: Difficulty=205 (Good luck!)
IP ID Sequence Generation: All zeros

Read data files from: /usr/local/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 5.58 seconds
           Raw packets sent: 1063 (47.432KB) | Rcvd: 1031 (41.664KB)
```

Including the -O -v options caused Nmap to generate the following extra line items:

- Device type
- Running
- OS CPE
- OS details
- Uptime guess
- Network Distance
- TCP Sequence Prediction
- IP ID sequence generation

Example 8.2. Using version scan to detect the OS

```
# nmap -sV -O -v 129.128.X.XX

Starting Nmap ( http://nmap.org )
Nmap scan report for [hostname] (129.128.X.XX)
Not shown: 994 closed ports
PORT      STATE    SERVICE      VERSION
21/tcp    open     ftp          HP-UX 10.x ftpd 4.1
22/tcp    open     ssh          OpenSSH 3.7.1p1 (protocol 1.99)
111/tcp   open     rpc
445/tcp   filtered microsoft-ds
1526/tcp  open     oracle-tns   Oracle TNS Listener
32775/tcp open     rpc
No exact OS matches for host
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: HP-UX
```

In this example, the line "No exact OS matches for host" means that TCP/IP fingerprinting failed to find an exact match. Fortunately, the Service Info field a few lines down discloses that the OS is HP-UX. If several operating systems were detected (which can happen with NAT gateway boxes that redirect ports to several different machines), the field would be OSs and the values would be comma separated. The Service Info line can also contain hostnames and device types found during the version scan.

13. Do the following using NS2 Simulator

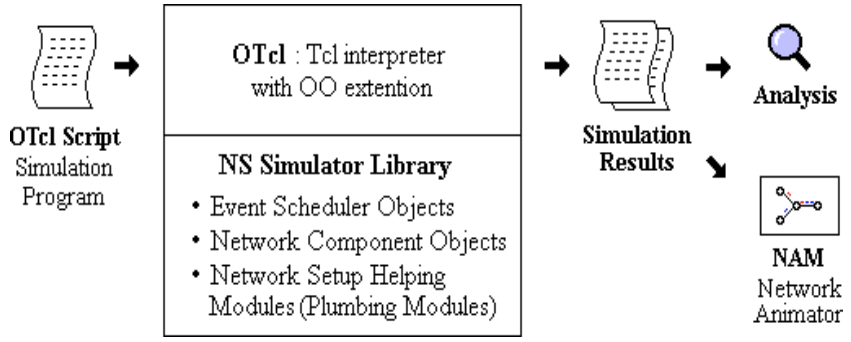
i. **NS2 Simulator-Introduction**
ii. **Simulate to Find the Number of Packets Dropped**
iii. **Simulate to Find the Number of Packets Dropped by TCP/UDP**
iv. **Simulate to Find the Number of Packets Dropped due to Congestion**
v. **Simulate to Compare Data Rate& Throughput.**
vi. **Simulate to Plot Congestion for Different Source/Destination**
vii. **Simulate to Determine the Performance with respect to Transmission of Packets.**

Introduction to NS-2:

NS2 is an open-source simulation tool that runs on Linux. It is a discreet event simulator targeted at networking research and provides substantial support for simulation of routing, multicast protocols and IP protocols, such as UDP, TCP, RTP and SRM over wired and wireless (local and satellite) networks.

- Widely known as NS2, is simply an event driven simulation tool.
- Useful in studying the dynamic nature of communication networks.
- Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2.
- In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviors.

Basic Architecture of NS2



TCL – Tool Command Language

Tcl is a very simple programming language. If you have programmed before, you can learn enough to write interesting Tcl programs within a few hours. This page provides a quick overview of the main features of Tcl. After reading this you'll probably be able to start writing simple Tcl scripts on your own; however, we recommend that you consult one of the many available Tcl books for more complete information.

Basic syntax

Tcl scripts are made up of *commands* separated by newlines or semicolons.

Commands all have the same basic form illustrated by the following example:

```
expr 20 + 10
```

This command computes the sum of 20 and 10 and returns the result, 30. You can try out this example and all the others in this page by typing them to a Tcl application such as tclsh; after a command completes, tclsh prints its result.

Each Tcl command consists of one or more *words* separated by spaces. In this example there are four words: expr, 20, +, and 10. The first word is the name of a command and the other words are *arguments* to that command. All Tcl commands consist of words, but different commands treat their arguments differently. The expr command treats all of its arguments together as an arithmetic expression, computes the result of that expression, and returns the result as a string. In the expr command the division into words isn't significant: you could just as easily have invoked the same command as

```
expr 20+10
```

However, for most commands the word structure is important, with each word used for a distinct purpose.

All Tcl commands return results. If a command has no meaningful result then it returns an empty string as its result.

Variables

Tcl allows you to store values in variables and use the values later in commands. The set command is used to write and read variables. For example, the following command modifies the variable x to hold the value 32:

```
set x 32
```

The command returns the new value of the variable. You can read the value of a variable by invoking set with only a single argument:

```
set x
```

You don't need to declare variables in Tcl: a variable is created automatically the first time it is set. Tcl variables don't have types: any variable can hold any value.
To use the value of a variable in a command, use *variable substitution* as in the following example:

```
expr $x*3
```

When a $ appears in a command, Tcl treats the letters and digits following it as a variable name, and substitutes the value of the variable in place of the name. In this example, the actual argument received by the expr command will be 32*3 (assuming that variable x was set as in the previous example). You can use variable substitution in any word of any command, or even multiple times within a word:

```
set cmd expr set x 11

$cmd $x*$x
```

Command substitution

You can also use the result of one command in an argument to another command.

This is called *command substitution*:

```
set a 44
set b [expr $a*4]
```

When a [appears in a command, Tcl treats everything between it and the matching] as a nested Tcl command. Tcl evaluates the nested command and substitutes its result into the enclosing command in place of the bracketed text. In the example above the second argument of the second set command will be 176.

Quotes and braces:

```
set x 24
set y 18
set z "$x + $y is [expr $x + $y]"
```

After these three commands are evaluated variable z will have the value 24 + 18 is 42. Everything between the quotes is passed to the set command as a single word. Note that (a) command and variable substitutions are performed on the text between the quotes, and (b) the quotes themselves are not passed to the command. If the quotes were not present, the set command would have received 6 arguments, which would have caused an error.

Curly braces provide another way of grouping information into words. They are different from quotes in that no substitutions are performed on the text between the curly braces:

```
set z {$x + $y is [expr $x + $y]}
```

This command sets variable z to the value "$x + $y is [expr $x + $y]".

Control structures:

Tcl provides a complete set of control structures including commands for conditional execution, looping, and procedures. Tcl control structures are just commands that take Tcl scripts as arguments. The example below creates a Tcl procedure called power, which raises a base to an integer power:

```
proc
  power
  {base
  p} {
  set
  result 1
  while {$p > 0} {
    set result [expr
    $result * $base]
    set p [expr $p - 1]
  }
  return $result
}
```

This script consists of a single command, proc. The proc command takes three arguments: the name of a procedure, a list of argument names, and the body of the procedure, which is a Tcl script. Note that everything between the curly brace at the end of the first line and the curly brace on the last line is passed verbatim to proc as a single argument. The proc command creates a new Tcl command named power that takes two arguments. You can then invoke power with commands like the following:

```
power 2 6
power 1.15 5
```

When power is invoked, the procedure body is evaluated. While the body is executing it can access its arguments as variables: base will hold the first argument and p will hold the second.

The body of the power procedure contains three Tcl commands: set, while, and return. The while command does most of the work of the procedure. It takes two arguments, an expression ($p > 0) and a body, which is another Tcl script. The while

Double-quotes allow you to specify words that contain spaces. For example, consider the following script:

command evaluates its expression argument using rules similar to those of the C programming language and if the result is true (nonzero) then it evaluates the body as a Tcl script. It repeats this process over and over until eventually the expression evaluates to false (zero). In this case the body of the while command multiplied the result value by base and then decrements p. When p reaches zero the result contains the desired power of base. The return command causes the procedure to exit with the value of variable result as the procedure's result.

Where do commands come from?

As you have seen, all of the interesting features in Tcl are represented by commands. Statements are commands, expressions are evaluated by executing commands, control structures are commands, and procedures are commands.

Tcl commands are created in three ways. One group of commands is provided by the Tcl interpreter itself. These commands are called *builtin commands*. They include all of the commands you have seen so far and many more (see below). The builtin commands are present in all Tcl applications.

The second group of commands is created using the Tcl extension mechanism. Tcl provides APIs that allow you to create a new command by writing a *command procedure* in C or C++ that implements the command. You then register the command procedure with the Tcl interpreter by telling Tcl the name of the command that the procedure implements. In the future, whenever that particular name is used for a Tcl command, Tcl will call your command procedure to execute the command. The builtin commands are also implemented using this same extension mechanism; their command procedures are simply part of the Tcl library.

When Tcl is used inside an application, the application incorporates its key features into Tcl using the extension mechanism. Thus the set of available Tcl commands varies from application to application. There are also numerous extension packages that can be incorporated into any Tcl application. One of the best known extensions is Tk, which provides powerful facilities for building graphical user interfaces. Other extensions provide object-oriented programming, database access, more graphical capabilities, and a variety of other features. One of Tcl's greatest advantages for building integration applications is the ease with which it can be extended to incorporate new features or communicate with other resources.

The third group of commands consists of procedures created with the proc command, such as the power command created above. Typically, extensions are used for lower-level functions where C programming is convenient, and procedures are used for higher-level functions where it is easier to write in Tcl.

Wired TCL Script Components

Create the event scheduler

Open new files & turn on the tracing

Create the nodes

Setup the links

Configure the traffic type (e.g., TCP, UDP, etc) Set

the time of traffic generation (e.g., CBR, FTP)

Terminate the simulation

NS Simulator Preliminaries.

Initialization and termination aspects of the ns simulator.

Definition of network nodes, links, queues and topology.

Definition of agents and of applications.

The nam visualization tool.

Tracing and random variables.

Features of NS2

NS2 can be employed in most unix systems and windows. Most of the NS2 code is in C++. It uses TCL as its scripting language, Otcl adds object orientation to TCL.NS(version 2) is an object oriented, discrete event driven network simulator that is freely distributed and open source.

- Traffic Models: CBR, VBR, Web etc
- Protocols: TCP, UDP, HTTP, Routing algorithms,MAC etc
- Error Models: Uniform, bursty etc
- Misc: Radio propagation, Mobility models , Energy Models
- Topology Generation tools
- Visualization tools (NAM), Tracing

Structure of NS

- NS is an object oriented discrete event simulator
  – Simulator maintains list of events and executes one event after another
  – Single thread of control: no locking or race conditions
- Back end is C++ event scheduler
  – Protocols mostly
  – Fast to run, more control
- Front end is OTCL

Creating scenarios, extensions to C++ protocols

fast to write and change

Platforms

It can be employed in most unix systems(FreeBSD, Linux, Solaris) and Windows.

Source code

Most of NS2 code is in C++

Scripting language

It uses TCL as its scripting language OTcl adds object orientation to TCL.

Protocols implemented in NS2

Transport layer(Traffic Agent) – TCP, UDP

Network layer(Routing agent)

Interface queue – FIFO queue, Drop Tail queue, Priority queue

Logic link contol layer – IEEE 802.2, AR

How to use NS2

Design Simulation – Determine simulation scenario

Build ns-2 script using tcl.

Run simulation

Simulation with NS2

Define objects of simulation. Connect

the objects to each other

Start the source applications. Packets are then created and are transmitted through network.

Exit the simulator after a certain fixed time.

NS programming Structure

- Create the event scheduler
- Turn on tracing
- Create network topology
- Create transport connections
- Generate traffic
- Insert errors

Sample Wired Simulation using NS-2

Creating Event Scheduler

- Create event scheduler: set ns [new simulator]
- Schedule an event: $ns at <time> <event>

– event is any legitimate ns/tcl function

```
$ns at 5.0 “finish”

proc finish {} {

global ns nf

close $nf

exec nam out.nam &

exit 0

}
```

- Start Scheduler

```
$ns run
```

Tracing

- All packet trace

```
$ns traceall[open out.tr w]

<event> <time> <from> <to> <pkt> <size>

…

<flowid> <src> <dst> <seqno> <aseqno>

+ 0.51 0 1 cbr 500 —— 0 0.0 1.0 0 2

_ 0.51 0 1 cbr 500 —— 0       0.0  1.0  0   2

R 0.514 0 1 cbr 500 ——0      0.0   1.0  0   0
```

- Variable trace

```
set par [open output/param.tr w]

$tcp attach $par
```

$tcp trace cwnd_

$tcp trace maxseq_

$tcp trace rtt_

Tracing and Animation

- Netwo

rk Animator

set nf [open

out.nam w]

$ns namtraceall

$nf

proc finish {} {

global ns nf close

$nf

exec nam out.nam &

exit 0

}

Creating topology

- Two nodes connected by a link
- Creating nodes
  set n0 [$ns node]
  set n1 [$ns node]
- Creating link between nodes

$ns <link_type> $n0 $n1 <bandwidth> <delay><queue-type>

$ns duplex-link$n0 $n1 1Mb 10ms DropTail

Data Sending

- Create UDP agent

set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

- Create CBR traffic source for feeding into UDP agent set cbr0 [new Application/Traffic/CBR]

```
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent$udp0
```

- Create traffic sink

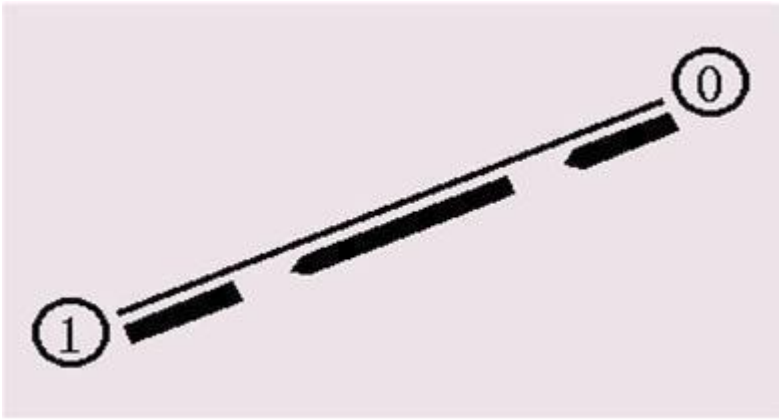
```
set null0 [new Agent/Null]
$ns attach-agent$n1 $null0
```

- Connect two agents

```
$ns connect $udp0 $null0
```

- Start and stop of data

```
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```



Traffic on top of TCP

- FTP

```
set ftp [new Application/FTP]
$ftp attach-agent$tcp0
```

- Telnet

```
set telnet [new Application/Telnet]
```

$telnet attach-agent$tcp0

PROCEDURE

STEP 1: Start

STEP 2: Create the simulator object ns for designing the given simulation STEP 3: Open the trace file and nam file in the write mode

STEP 4: Create the nodes of the simulation using the 'set' command

STEP 5: Create links to the appropriate nodes using $ns duplex-link command STEP 6: Set the orientation for the nodes in the simulation using 'orient' command STEP 7: Create TCP agent for the nodes and attach these agents to the nodes STEP 8: The traffic generator used is FTP for both node0 and node1

STEP 9: Configure node1 as the sink and attach it

STEP10: Connect node0 and node1 using 'connect' command STEP 11:

Setting color for the nodes

STEP 12: Schedule the events for FTP agent 10 sec STEP 13:

Schedule the simulation for 5 minutes.

Structure of Trace Files

When tracing into an output ASCII file, the trace is organized in 12 fields as follows in fig shown below,

The meaning of the fields are:

Event	Time	From Node	To Node	PKT Type	PKT Size	Flags	Fid	Src Addr	Dest Addr	Seq Num	Pkt id

1. The first field is the event type. It is given by one of four possible symbols r, +, -, d which correspond respectively to receive (at the output of the link), enqueued, dequeued and dropped.
2. The second field gives the time at which the event occurs.
3. Gives the input node of the link at which the event occurs.

4. Gives the output node of the link at which the event occurs.
5. Gives the packet type (eg CBR or TCP)
6. Gives the packet size
7. Some flags
8. This is the flow id (fid) of IPv6 that a user can set for each flow at the input OTcl script one can further use this field for analysis purposes; it is also used when specifying stream color for the NAM display.
9. This is the source address given in the form of ―node.port‖.
10. This is the destination address, given in the same form.
11. This is the network layer protocol's packet sequence number. Even though UDP implementations in a real network do not use sequence number, ns keeps track of UDP packet sequence number for analysis purposes

The last field shows the Unique id of the packet.

XGRAPH

The xgraph program draws a graph on an x-display given data read from either data file or from standard input if no files are specified. It can display upto 64 independent data sets using different colors and line styles for each set. It annotates the graph with a title, axis labels, grid lines or tick marks, grid labels and a legend.

Syntax:

Xgraph [options] file-name

Options are listed here

/-bd <color> (Border)

This specifies the border color of the xgraph window.

/-bg <color> (Background)

This specifies the background color of the xgraph window.

/-fg<color> (Foreground)

This specifies the foreground color of the xgraph window.

/-lf <fontname> (LabelFont)

All axis labels and grid labels are drawn using this font.

/-t<string> (Title Text)

This string is centered at the top of the graph.

/-x <unit name> (XunitText)

This is the unit name for the x-axis. Its default is —X‖.

/-y <unit name> (YunitText)

This is the unit name for the y-axis. Its default is —Y‖.

Transmission of Ping messages: Experiment No. 2

Aim: Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

```
set ns [ new Simulator ] set nf [
open lab4.nam w ]

$ns namtrace-all $nf set tf [
open lab4.tr w ]

$ns trace-all $tf set n0
[$ns node] set n1 [$ns
node] set n2 [$ns node]
set n3 [$ns node] set n4
[$ns node] set n5 [$ns
node]


$ns duplex-link $n0 $n4 1005Mb 1ms DropTail

$ns duplex-link $n1 $n4 50Mb 1ms DropTail

$ns duplex-link $n2 $n4 2000Mb 1ms DropTail

$ns duplex-link $n3 $n4 200Mb 1ms DropTail

$ns duplex-link $n4 $n5 1Mb 1ms DropTail


set p1 [new Agent/Ping] # letters A and P should be capital

$ns attach-agent $n0 $p1

$p1 set packetSize_ 50000

$p1 set interval_ 0.0001


set p2 [new Agent/Ping] # letters A and
P should be capital
```

```
$ns attach-agent $n1 $p2


set p3 [new Agent/Ping] # letters A and P should be capital
$ns attach-agent $n2 $p3
$p3 set packetSize_ 30000
$p3 set interval_ 0.00001
set p4 [new Agent/Ping] # letters A and
P should be capital
$ns attach-agent $n3 $p4


set p5 [new Agent/Ping] # letters A and
P should be capital
$ns attach-agent $n5 $p5


$ns queue-limit $n0 $n4 5
$ns queue-limit $n2 $n4 3
                $ns queue-limit $n4 $n5 2
Agent/Ping instproc recv {from
rtt} {
$self instvar node_
puts "node [$node_ id]received answer from $from with round trip time $rtt msec"
}
#                please provide space between $node_ and id. No space
between $ and from. No space between and $ and rtt */


$ns connect $p1 $p5
$ns connect $p3 $p4


proc finish
{ } { global
ns nf tf
$ns
flush-trac
e close
$nf close
$tf
```

```
exec nam
lab4.nam & exit
0
}
$ns at 0.1 "$p1 send"
$ns at 0.2 "$p1 send"
$ns at 0.3 "$p1 send"
$ns at 0.4 "$p1 send"
$ns at 0.5 "$p1 send"
$ns at 0.6 "$p1 send"
$ns at 0.7 "$p1 send"
$ns at 0.8 "$p1 send"
$ns at 0.9 "$p1 send"
$ns at 1.0 "$p1 send"
$ns at 1.1 "$p1 send"
$ns at 1.2 "$p1 send"
$ns at 1.3 "$p1 send"
$ns at 1.4 "$p1 send"
$ns at 1.5 "$p1 send"
$ns at 1.6 "$p1 send"
$ns at 1.7 "$p1 send"
$ns at 1.8 "$p1 send"
$ns at 1.9 "$p1 send"
$ns at 2.0 "finish"
$ns run
```

AWK file: (Open a new editor using "vi command" and write awk file and save with ".awk" extension)

```
BEGIN{
pingDrop=0;
}
{
 if($1= ="d" )
  {
```

```
    pingDrop++;
    }
}
E
N
D
{
printf("Total number of ping packets dropped due to congestion is
=%d\n",pingDrop);
}
```

Simulation of Ethernet Lan Experiment No. 3

Experiment Specific Instructions

1. To analyze the given problem you have to write a Tcl script and simulate with ns2
2. Begin by specifying the trace files and the nam files to be created
3. Define a finish procedure
4. Determine and create the nodes that will be used to create the topology. Here in our experiment we are selecting 6 nodes namely 0, 1, 2, 3, 4, 5
5. Create the links to connect the nodes
6. Set up the LAN by specifying the nodes, and assign values for bandwidth, delay, queue type and channel to it
7. Optionally you can position and orient the nodes and links to view a nice video output with Nam
8. Set up the TCP and/or UDP connection(s) and the FTP/CBR (or any other application) that will run over it
9. Schedule the different events like simulation start and stop, data transmission start and stop
10. Call the finish procedure and mention the time at what time your simulation will end
11. Execute the script with ns

Simulation Script:

```
#Lan simulation

set ns [new Simulator] #define
color for data flows

$ns color 1 Blue

        $ns color 2
         Red #open
         tracefiles

set tracefile1 [open out.tr w] set
winfile [open winfile w]
```

```
$ns trace-all $tracefile1 #open
nam file

set namfile [open out.nam w]

$ns namtrace-all $namfile #define
the finish procedure proc finish {} {

global ns tracefile1 namfile

$ns flush-trace close
$tracefile1 close $namfile

exec nam out.nam & exit 0

} #create six nodes set n0
[$ns node] set n1 [$ns
node] set n2 [$ns node] set
n3 [$ns node] set n4 [$ns
node] set n5 [$ns node]

$n1 color Red

$n1 shape box

#create links between the nodes

$ns duplex-link $n0 $n2 2Mb 10ms DropTail

$ns duplex-link $n1 $n2 2Mb 10ms DropTail

$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail

$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail

set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd
Channel]

#Give node position

$ns duplex-link-op $n0 $n2 orient right-down

$ns duplex-link-op $n1 $n2 orient right-up

$ns simplex-link-op $n2 $n3 orient right

$ns simplex-link-op $n3 $n2 orient left #set
queue size of link(n2-n3) to 20

                $ns queue-limit $n2
                 $n3 20 #setup TCP
                 connection

set tcp [new Agent/TCP/Newreno]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink/DelAck]

$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink

$tcp set fid_ 1

$tcp set packet_size_ 552 #set ftp
over tcp connection set ftp [new
Application/FTP]

$ftp attach-agent $tcp #setup a
UDP connection set udp [new
Agent/UDP]

$ns attach-agent $n1 $udp set null
[new Agent/Null]

$ns attach-agent $n5 $null

$ns connect $udp $null

$udp set fid_ 2

#setup a CBR over UDP connection set cbr
[new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1000

$cbr set rate_ 0.01Mb

$cbr set random_ false
#scheduling the events

$ns at 0.1 "$cbr start"

$ns at 1.0 "$ftp start"

$ns at 124.0 "$ftp stop"

$ns at 125.5 "$cbr stop"

proc plotWindow {tcpSource file} { global
ns

set time 0.1

set now [$ns now]

set cwnd [$tcpSource set cwnd_] puts
$file "$now $cwnd"

                $ns at [expr $now+$time] "plotWindow $tcpSource $file"

}

$ns at 0.1 "plotWindow $tcp $winfile"

                $ns at 125.0 "finish"

                $ns run
```

Wireless Simulation using NS-2

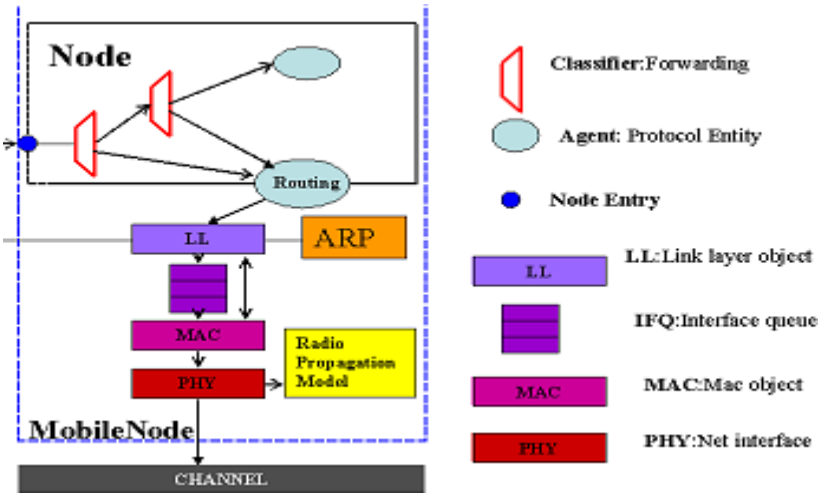
Simple Wireless Program in NS2 is the best way to learn about how to code in NS2.

NS2 is one of the best simulation tool used by majority of scholars today due to its highlighted features like support for OOPs concept, C++ programming fundamentals, real time emulation support etc. NS2 is used to simulate both wired and wireless networks; here we have focused on wireless network simulation in NS2 due to its wide applicability.

Regarding wired simulation in NS2, refer our other articles available in this site. Here, we have taken a simple wireless program in NS2 to explain the students about how to work with wireless networks in NS2.

Simulating a Wireless Network in NS2

- **Wireless Nodes**
  - A mobile node consists of network components:
    - Link Layer (LL)
    - Interface Queue (IfQ)
    - the MAC layer
    - the PHY layer: the wireless channel that the node transmit and receive signals from



  - At the beginning of a wireless simulation, we need to define the **type** for each of these network components.
  - Additionally, we need to define other parameters like:
    - the type of antenna
    - the radio-propagation model

the type of ad-hoc routing protocol used by mobilenodes etc.

- **Configuring a Wireless Node**
  - Creating **wireless nodes** is also achieved using the **ns_ node** command:

```
set ns_ [new Simulator]          ;# Create a NS
  simulator object set n1  [ns_ node]  ;# Create a
  WIRELESS node !!!
```

  - **However:**

    BEFORE creating a **wireless node** you **MUST** first **select (configure)**
    the node configuration parameters to "become" a **wireless node**.

  - The **NS2** command to **select (configure)** node configuration parameters is **node-config** and it is used as follows:

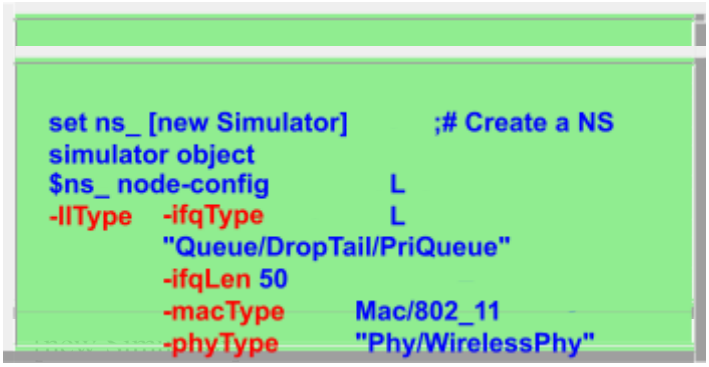
```
set ns_ [new Simulator]          ;# Create a NS simulator object

$ns_ node-config \
-llType      LL
-ifqType     "Queue/DropTail/PriQueue"
-ifqLen      50
-macType Mac/802_11
-phyType  "Phy/WirelessPhy"

-addressingType flat or hierarchical or expanded
-adhocRouting DSDV or DSR or TORA
-propType"Propagation/TwoRayGround"
-antType   "Antenna/OmniAntenna"
-channelType   "Channel/WirelessChannel"
-topoInstance $topo
-energyModel "EnergyModel"
-initialEnergy (in Joules)
-rxPower  (in W)
-txPower  (in W)

-agentTrace      ON or OFF
-routerTrace     ON or OFF
-macTrace        ON or OFF
-movementTrace ON or OFF
```

  - The value of most of the parameters are **simple values**
  - **Except** for the value of the **-topoInstance** parameters.
  - The **topology** is a **Topography** object that you must create.

Example:

```
set topo [new Topography]      ;# Create a Topography
   object
$topo load_flatgrid 500 500 ;# Make a 500x500 grid
   topology
```

A **commonly used** wireless node configuration is:



#Fixing the co-ordinate of simutaion area set
val(x) 500

set val(y) 500

Define options

set val(chan) Channel/WirelessChannel ;# channel type

set val(prop) Propagation/TwoRayGround ;# radio-propagation model

set val(netif) Phy/WirelessPhy ;# network interface type set
val(mac) Mac/802_11 ;# MAC type

set val(ifq) Queue/DropTail/PriQueue ;# interface queue type set val(ll)
LL ;# link layer type

set val(ant) Antenna/OmniAntenna ;# antenna model set
val(ifqlen) 50 ;# max packet in ifq

set val(nn) 2 ;# number of mobilenodes set
val(rp) AODV ;# routing protocol

set val(x) 500 ;# X dimension of topography set
val(y) 400 ;# Y dimension of topography set
val(stop) 10.0 ;# time of simulation end

set up topography object set topo
[new Topography]

$topo load_flatgrid $val(x) $val(y)

#Nam File Creation nam – network animator set
namfile [open sample1.nam w]

#Tracing all the events and cofiguration

$ns namtrace-all-wireless $namfile $val(x) $val(y)

#Trace File creation

set tracefile [open sample1.tr w]

```
#Tracing all the events and cofiguration
$ns trace-all $tracefile

# general operational descriptor- storing the hop details in the network
create-god $val(nn)

# configure the nodes
$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-movementTrace ON

# Node Creation
set node1 [$ns node]
# Initial color of the node
$node1 color black

#Location fixing for a single node
$node1 set X_ 200
$node1 set Y_ 100
$node1 set Z_ 0

set node2 [$ns node]
$node2 color black

$node2 set X_ 200
$node2 set Y_ 300
$node2 set Z_ 0
```

```
# Label and coloring
$ns at 0.1 "$node1 color blue"
$ns at 0.1 "$node1 label Node1"
$ns at 0.1 "$node2 label Node2" #Size
of the node
$ns initial_node_pos $node1 30
$ns initial_node_pos $node2 30 # ending
nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
#Stopping the scheduler
$ns at 10.01 "puts \"end simulation\" ; $ns halt" #$ns at
10.01 "$ns halt"
proc stop {} {
global namfile tracefile ns
$ns   flush-trace   close
$namfile            close
$tracefile
#executing nam file
exec nam sample1.nam &
}
#Starting scheduler
$ns run
```

Experiment 5 &6:

Implement and study the performance of GSM or CDMA on NS2/NS3 (Using MAC layer) or equivalent Environment.

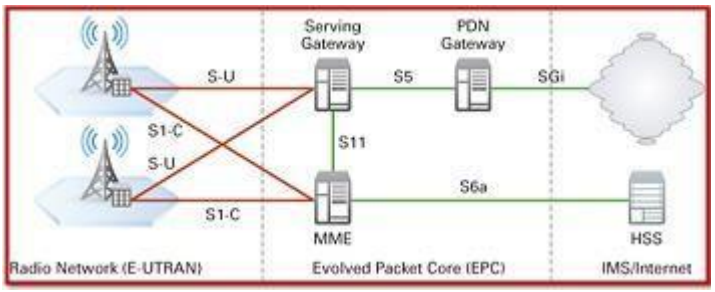
NS3 LTE Simulation

LTE is the latest high-speed cellular transmission network.LTE is a 4G technology with download speeds that run the gamut from 3 to 28 Mbps worldwide.4G LTE is one of several competing 4G standards along with Ultra Mobile Broadband (UMB) andWiMax (IEEE 802.16).Ns3 is the best choice among network simulator for simulating LTE framework.We provide ***customized NS3 LTE Simulation Projects*** based on customer Requirements.

Advantages of LTE:

- LTE will supports seamless connection to existing networks like GSM, CDMA and WCDMA.
- It has simple architecture because of low operating expenditure
- Time required for connecting network and is in range of a few hundred ms and power savings states can now be entered and exited very quickly
- High data rates can be achieved in both downlink as well as uplink.
- Both FDD and TDD can be used on same platform.
- Optimized signaling for connection establishment and other air interface and mobility management procedures have further improved the user experience.

Architecture of LTE:



LTE parameters:

- Transmission bandwidth.
- Mobility.
- Frequency range.
- Duplexing.
- Channel bandwidth.
- Channel coding.
- MIMO.
- Multi-antenna technology.

Code:

```
# General Parameters
    set opt(ecn) 0

    ; set opt(window) 30 ;
# Topology
    set opt(type) gsm ; #type of link: #
AQM parameters
set opt(minth) 5 ; set
opt(maxth) 10 ;
set opt(adaptive) 1 ; # 1 for Adaptive RED, 0 for plain RED #default
downlink bandwidth in bps
set bwDL(gsm) 9600
```

```
#default uplink bandwidth in bps set
bwUL(gsm) 9600
#default downlink propagation delay in seconds set
propDL(gsm) .500
#default uplink propagation delay in seconds set
propUL(gsm) .500
#default buffer size in packets set
buf(gsm) 10


set ns [new Simulator] set
tf [open out.tr w] set nf
[open out1.nam w]
$ns trace-all $tf
$ns namtrace-all $nf set
nodes(s) [$ns node]
set nodes(bs1) [$ns node]
set nodes(ms) [$ns node] set
nodes(bs2) [$ns node] set
nodes(d) [$ns node] proc
cell_topo {}

{
global ns nodes

  $ns duplex-link $nodes(s) $nodes(bs1) 3Mbps 10ms DropTail
  $ns duplex-link $nodes(bs1) $nodes(ms) 1Mbps 1ms RED
  $ns duplex-link $nodes(ms) $nodes(bs2) 1Mbps 1ms RED
  $ns duplex-link $nodes(bs2) $nodes(d) 3Mbps 50ms DropTail puts "Cell Topology"
}


  proc set_link_params {t} {
  global ns nodes bwUL bwDL propUL propDL buf
  $ns bandwidth $nodes(bs1) $nodes(ms) $bwDL($t) simplex
  $ns bandwidth $nodes(ms) $nodes(bs1) $bwUL($t) simplex
  $ns bandwidth $nodes(bs2) $nodes(ms) $bwDL($t) simplex
  $ns bandwidth $nodes(ms) $nodes(bs2) $bwUL($t) simplex
  $ns delay $nodes(bs1) $nodes(ms) $propDL($t) simplex
  $ns delay $nodes(ms) $nodes(bs1) $propDL($t) simplex
  $ns delay $nodes(bs2) $nodes(ms) $propDL($t) simplex
  $ns delay $nodes(ms) $nodes(bs2) $propDL($t) simplex
  $ns queue-limit $nodes(bs1) $nodes(ms) $buf($t)
  $ns queue-limit $nodes(ms) $nodes(bs1) $buf($t)
  $ns queue-limit $nodes(bs2) $nodes(ms) $buf($t)
  $ns queue-limit $nodes(ms) $nodes(bs2) $buf($t)
}
```

```
# RED and TCP parameters
Queue/RED set summarystats_ true
Queue/DropTail set summarystats_ true
Queue/RED set adaptive_ $opt(adaptive)
Queue/RED set q_weight_ 0.0
Queue/RED set thresh_ $opt(minth)
Queue/RED set maxthresh_ $opt(maxth)
Queue/DropTail set shrink_drops_ true
Agent/TCP set ecn_ $opt(ecn) Agent/TCP
set window_ $opt(window) DelayLink set
avoidReordering_ true #Create topology
switch $opt(type) {
gsm - gprs - umts {cell_topo}
}
  set_link_params $opt(type)
  $ns insert-delayer $nodes(ms) $nodes(bs1) [new Delayer]
  $ns insert-delayer $nodes(bs1) $nodes(ms) [new Delayer]
  $ns insert-delayer $nodes(ms) $nodes(bs2) [new Delayer]
  $ns insert-delayer $nodes(bs2) $nodes(ms) [new Delayer]


# Set up forward TCP connection
set tcp1 [$ns create-connection TCP/Sack1 $nodes(s) TCPSink/Sack1 $nodes(d) 0]
set ftp1 [[set tcp1] attach-app FTP]
  $ns at 0.5 "$ftp1 start"
proc stop {} {
global nodes ns opt nf tf
$ns flush-trace
close  $nf close
$tf
exec nam out1.nam
& exit 0
}
$ns at 100 "stop"
$ns run
```

AWK FILE

```
 BEGI
    N {
PacketRcvd=0;
Throughput=0.0
;
}
{
if(($1=="r") && ($5=="tcp") && ($10=4.0))
{
```

```
PacketRcvd++;
}
}
E
N
D
{
Throughput=((PacketRcvd*1000*8)/(95.0*1000000));
printf("packet received:%f\n", PacketRcvd); printf( "the
throughput is:%f\n",Throughput);
}
```

1. Write a PHP script to print prime numbers between 1-50.

1.1 OBJECTIVE:

Write a PHP script to print prime numbers between 1-50.

1.2 RESOURCES:

NOTEPAD++, 1GB RAM, Hard Disk 80 GB

1.3 PROGRAM LOGIC:

1. Create a PHP file.
2. Read a number in one text field and display that number name in another text field.
3. Include the JavaScript to convert number into words.

1.4 PROCEDURE:

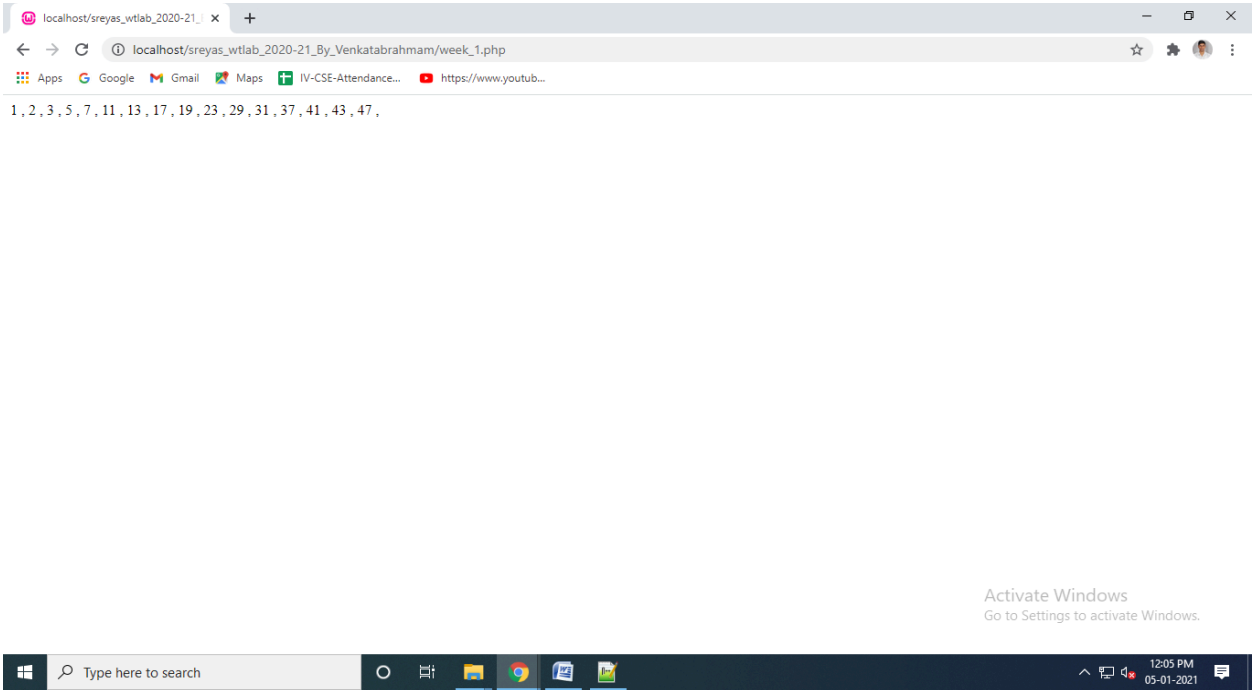
To execute a html program:
1. Open Notepad++ and Save the PHP program in **"SREYAS"** folder in drive.
3. To run the html file open the browser and type the following URL directory name/filename.

1.5 SOURCE CODE:

```
<?php
$num = 1 ;
while ($num < 50 ) {
   $count=0;
   for ( $i=1;$i<=$num;$i++) {
      if (($num%$i)==0) {
         $count++;
      }
   }
   if ($count<3) {
```

```
        echo $num." , ";
        } $num=$num+1;
    }
    ?>
```

OUTPUT:



WEEK-2: PHP script to

a. **Find the length of a string.**
b. **Count no of words in a string.**
c. **Reverse a string.**
d. **Search for a specific string.**

2.1 OBJECTIVE:

PHP script to

a. Find the length of a string.
b. Count no of words in a string.
c. Reverse a string.
d. Search for a specific string.

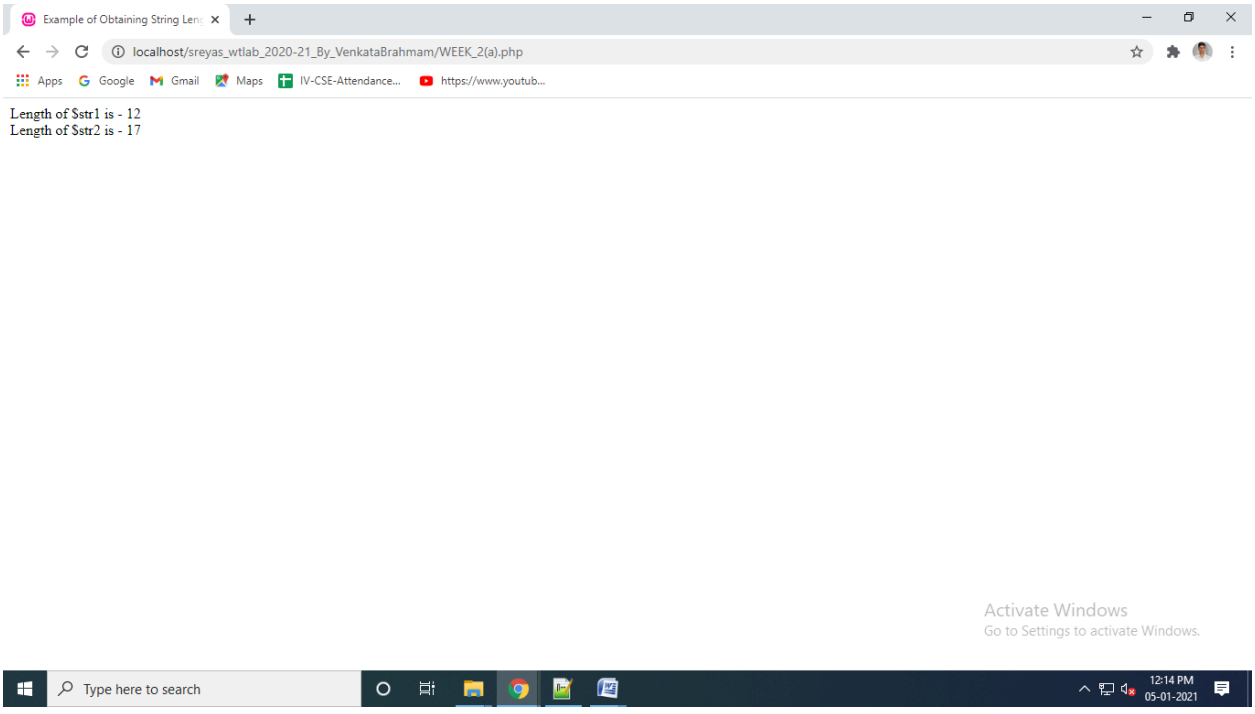
a). Find the length of a string.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Example of Obtaining String Length in PHP</title>
    </head>
<body>
```

```
<?php
$str1 = 'Hello world!';
        echo 'Length of $str1 is - ' . strlen($str1);
        echo "<br>";
$str2 = ' Hello    world! ';
        echo 'Length of $str2 is - ' . strlen($str2);
?>
</body>
</html>
```

OUTPUT:

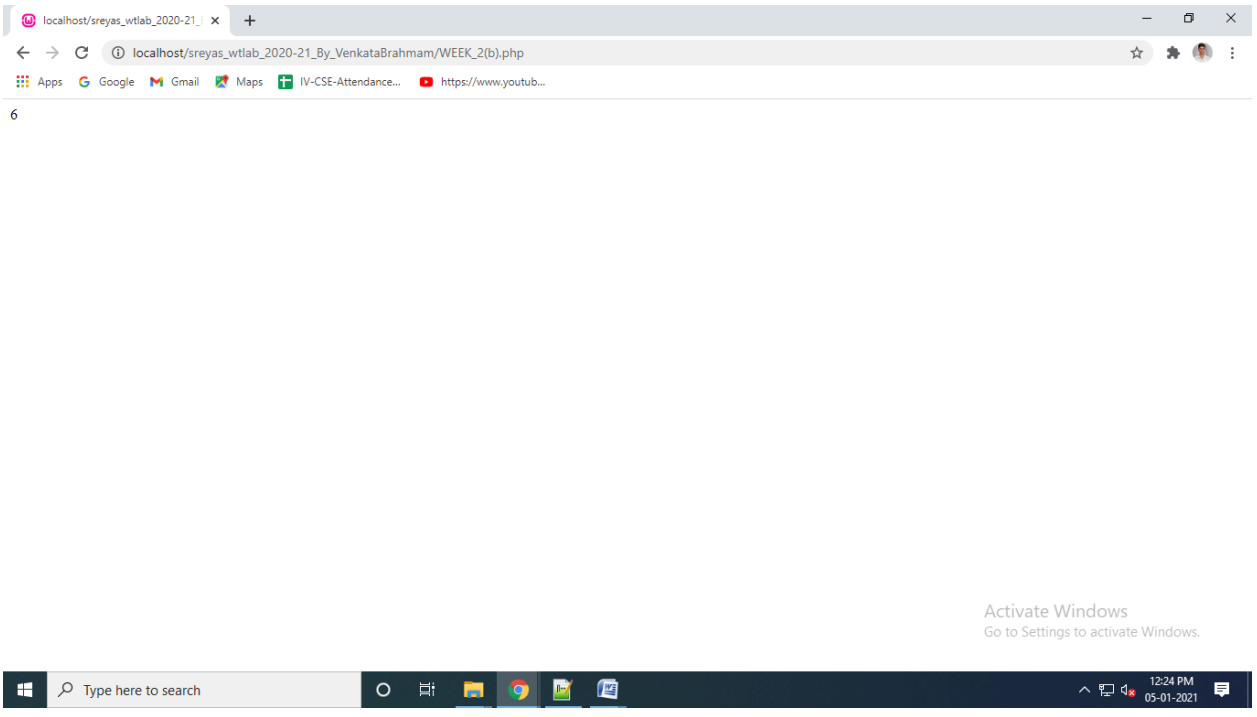


b). Count no of words in a string.

```
<?php
```

```
// PHP program to count number of words in a string

// Function to count the words
function get_num_of_words($string) {
                        $string = preg_replace('/\s+/', ' ', trim($string));
                        $words = explode(" ", $string);
                        return count($words);
}

$str = " SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY";

// Function call
$len = get_num_of_words($str);

// Printing the result
echo $len;
?>
```

OUTPUT:

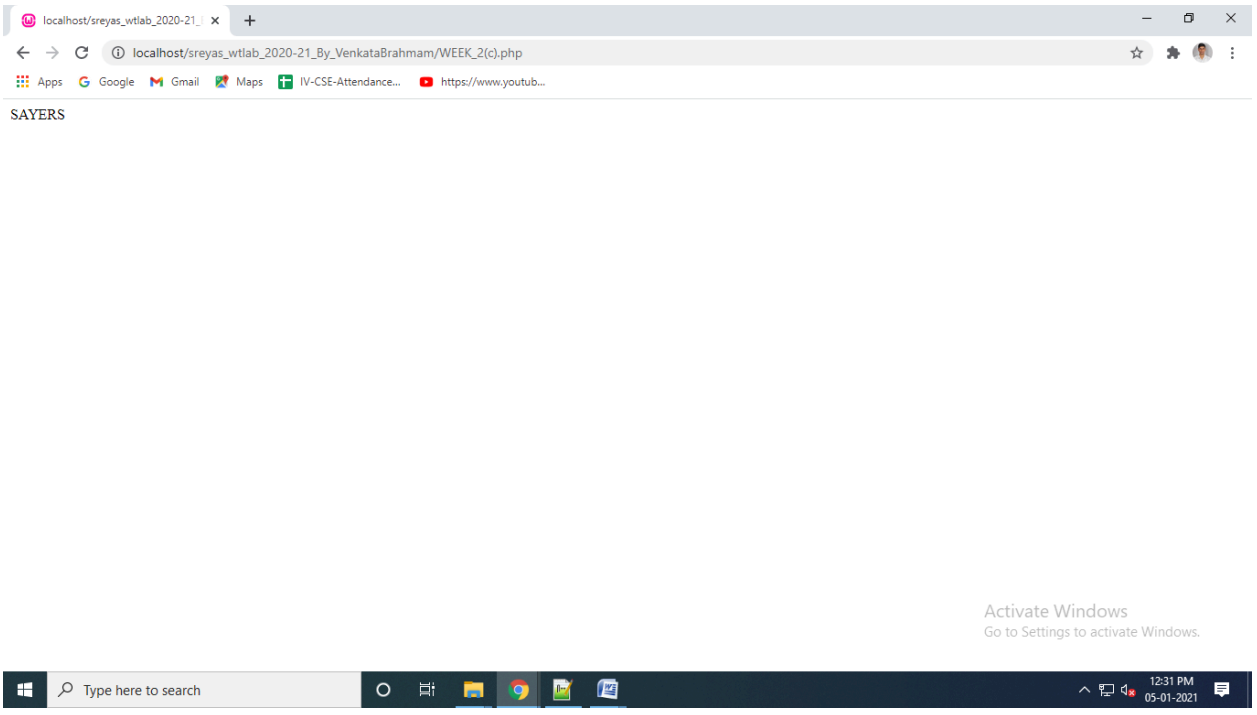


c). Reverse a string.

SOURCE CODE:

```
<?php
// PHP program to reverse a string using strrev()

function Reverse($str){
        return strrev($str);
}

// Driver Code
$str = "SREYAS";
echo Reverse($str)
?>
```

OUTPUT:

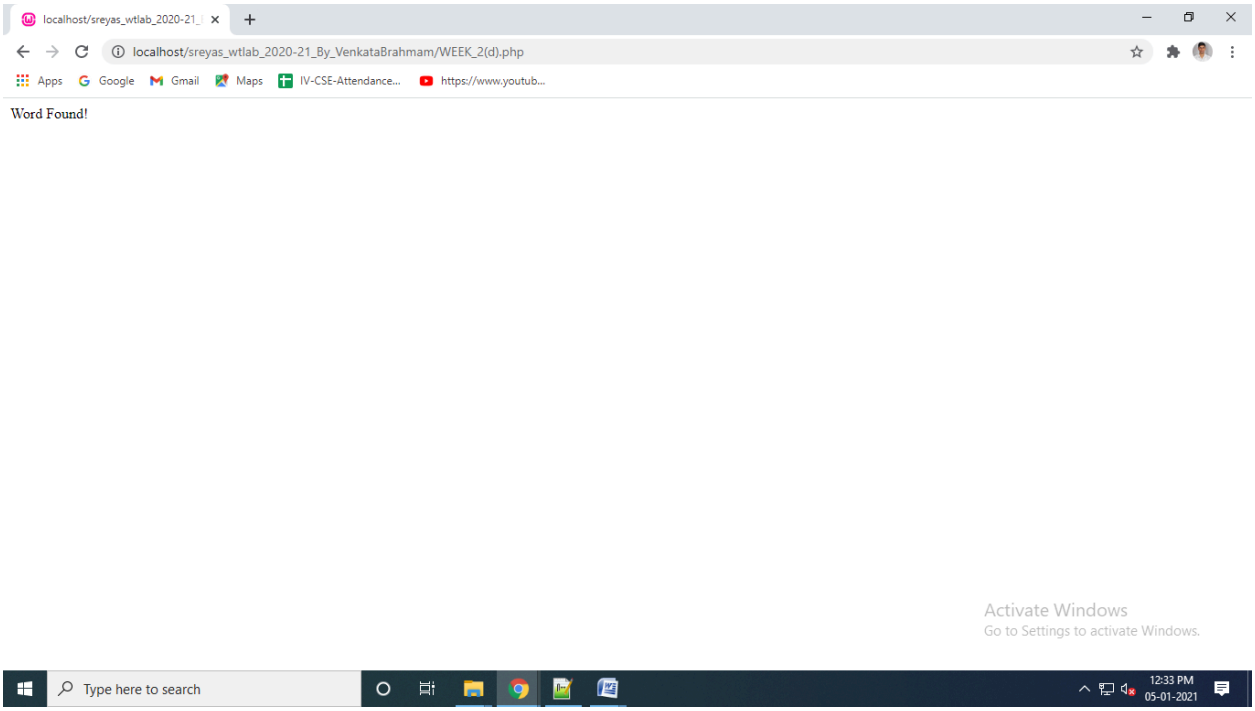


d). Search for a specific string.

SOURCE CODE:

```
<?php
$word = "SREYAS";
$mystring = "SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY";

// Test if string contains the word
if(strpos($mystring, $word) !== false){
   echo "Word Found!";
} else{
   echo "Word Not Found!";
}
?>
```

OUTPUT:

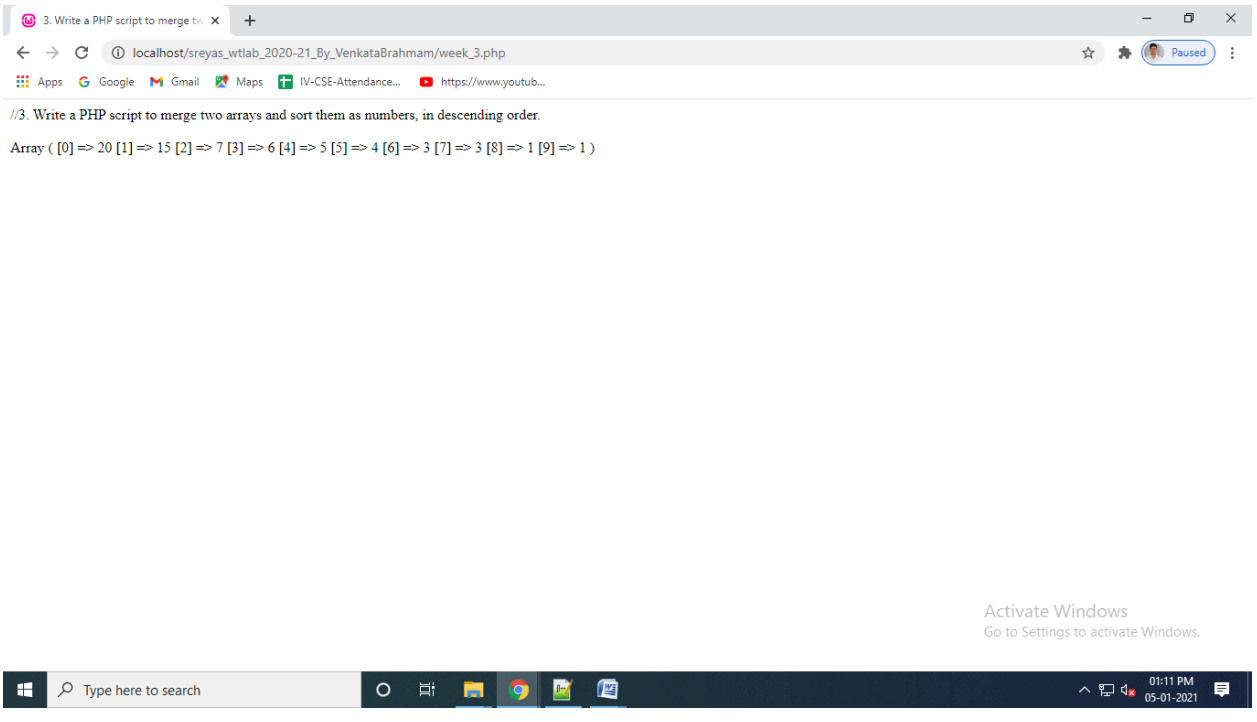

localhost/sreyas_wtlab_2020-21_
localhost/sreyas_wtlab_2020-21_By_VenkataBrahmam/WEEK_2(d).php
Apps
Google
Gmail
Maps
IV-CSE-Attendance...
https://www.youtub...
Word Found!
Activate Windows
Go to Settings to activate Windows.
Type here to search
12:33 PM
05-01-2021

3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.

SOURCE CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.</title>
</head>
<body>
//3. Write a PHP script to merge two arrays and sort them as numbers, in descending order.

<?php
        echo"<br><br>";
   $a1=array(1,3,15,7,5);
   $a2=array(4,3,20,1,6);
   $num=array_merge($a1,$a2);
   array_multisort($num,SORT_DESC,SORT_NUMERIC);
        print_r($num);

?>

</body>
</html>
```

OUTPUT:

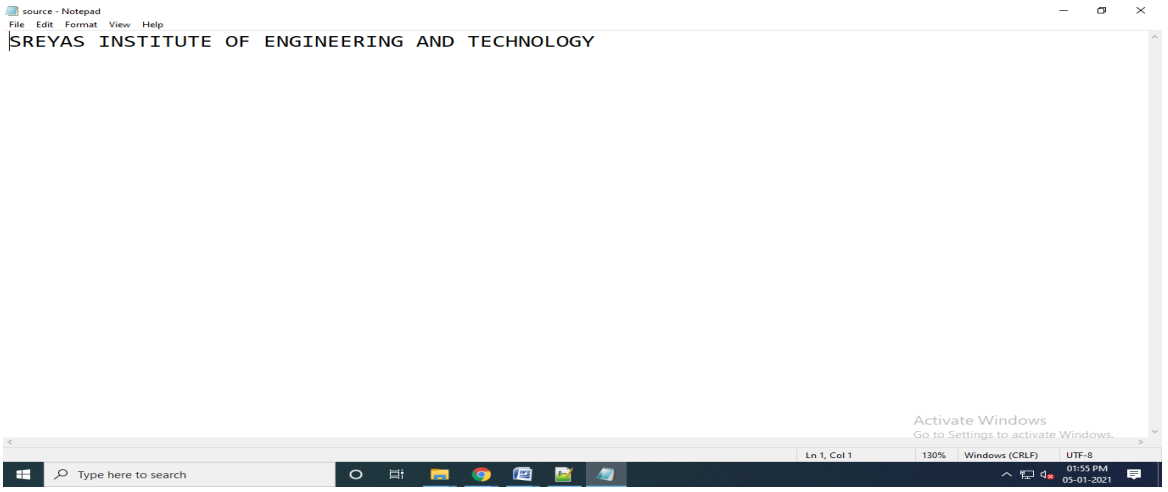


4. Write a PHP script that reads data from one file and write into another file.

SOURCE CODE:

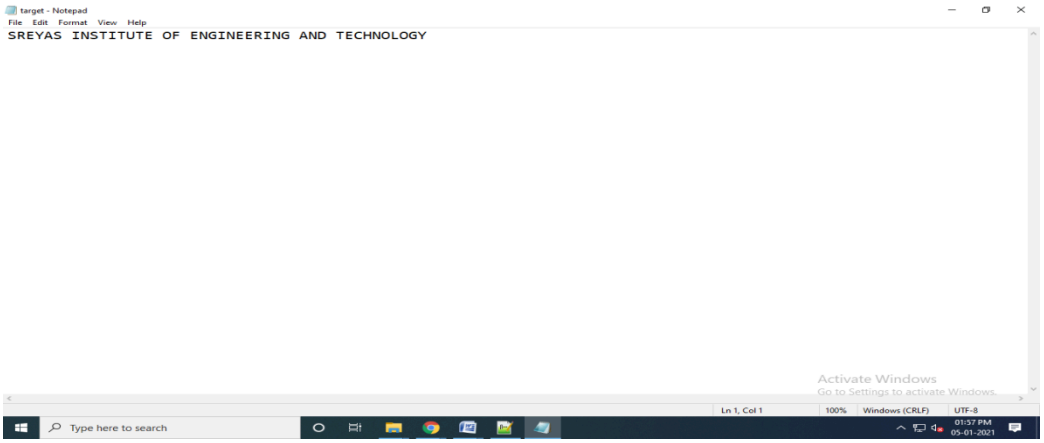
```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Write a PHP script that reads data from one file and write into another file.</title>
</head>
<body>
//4.      Write a PHP script that reads data from one file and write into another file.

<?php
        $source='source.txt';
$target='target.txt';

$lines=file( $source );
$data=array();

foreach( $lines as $line ){
   /* store each line with a leading zero into a temp array */
   $data[]='0' . trim( $line );
}

/* write the content back to another file */
file_put_contents( $target, implode( PHP_EOL, $data ) );

?>

</body>
</html>
```

source.txt:



OUTPUT:

target.txt:



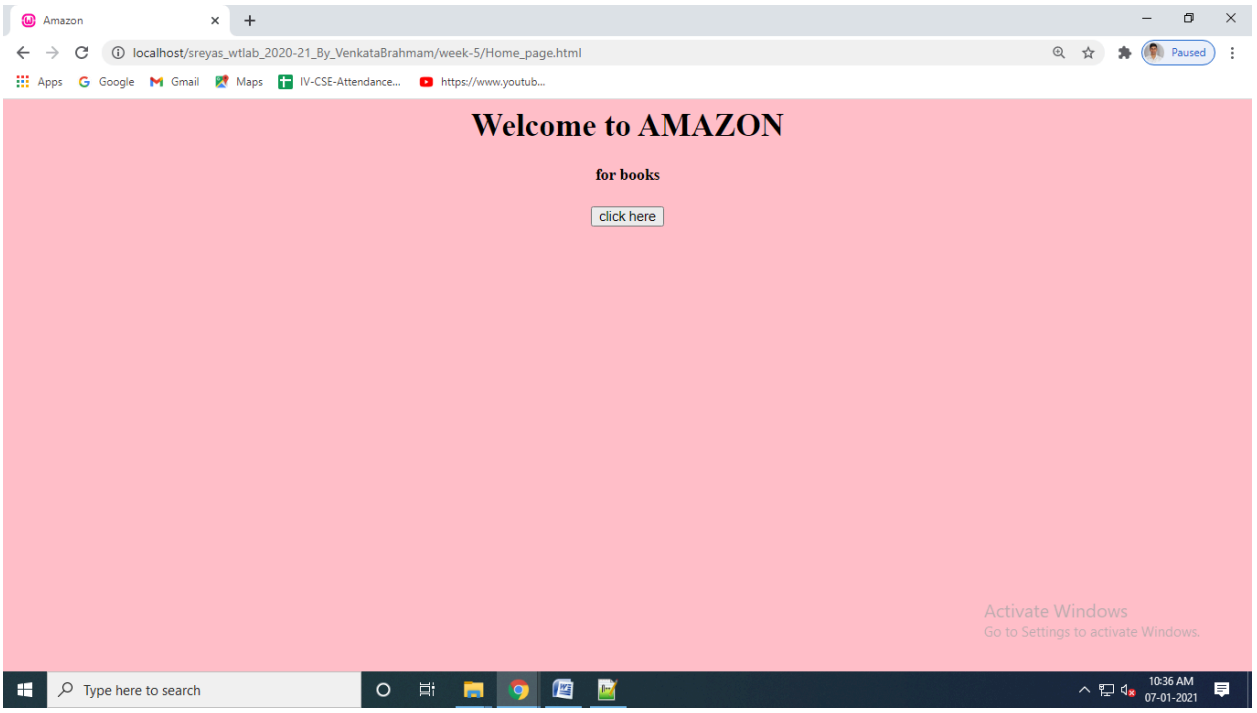
5. Develop static pages (using Only HTML) of an online book store. The pages should resemble: www.amazon.com. The website should consist the following pages.

i. **Home page**
ii. **Registration and user Login**
iii. **User Profile Page**
iv. **Books catalog**
v. **Shopping Cart**
vi. **Payment By credit card**
vii. **Order Conformation.**

SOURCE CODE:
i. Home_Page.html:

```
<html>
        <head>
                <title>
                        Amazon</title>
        </head>
<body bgcolor="pink"> <center>
        <strong><h1>Welcome to AMAZON</h1></strong>
                <form method="post" action="login.html" target=_blank >
        <h4>for books</h4><input type="submit" value="click here">
        </form>
                </center>
        </body>
</html>
```

OUTPUT:

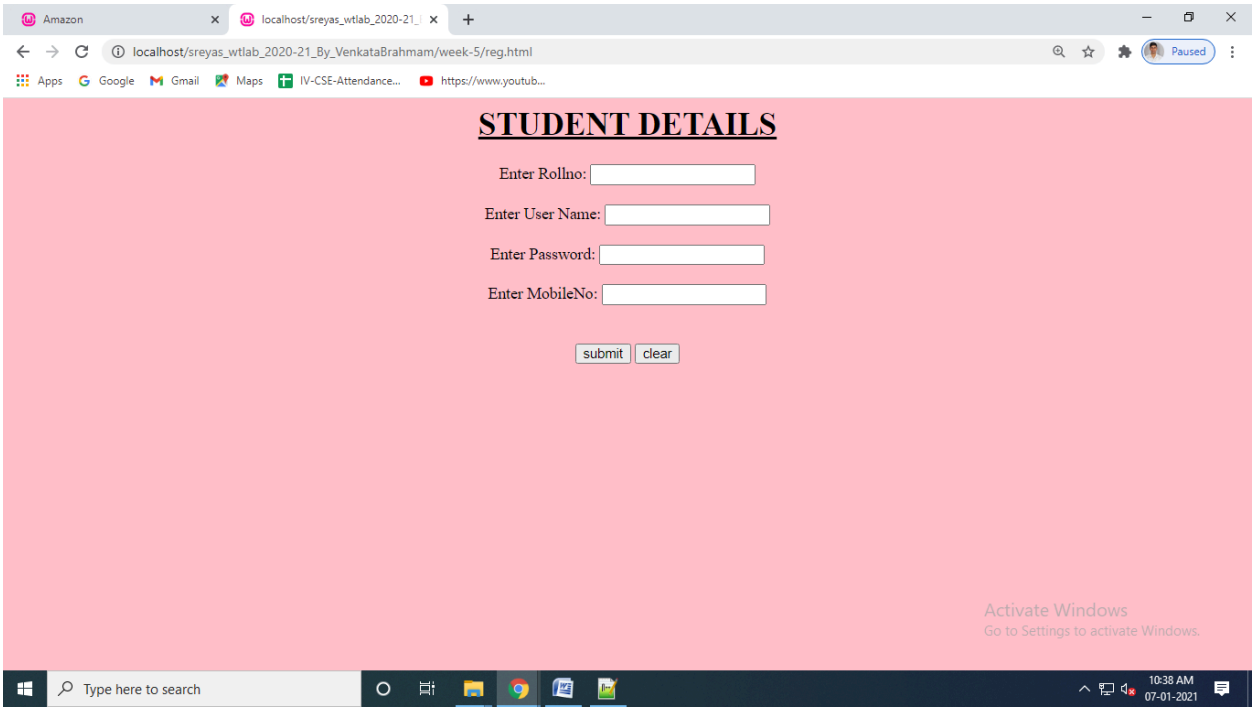


ii. Registration and user Login:

Reg.html:

```
<html>
<body bgcolor="pink">
<center>
<h1> <b> <u>STUDENT DETAILS</u></b> </h1>
<form action="insert_form.php" method="post">

Enter Rollno: <input type="text" name="rollno"><br/><br/>
Enter User Name: <input type="text" name="name"><br/><br/>
Enter Password: <input type="password" name="password"><br/><br/>
Enter MobileNo: <input type="text" name="mobile_no"><br/><br/><br/>

<input type="submit" value="submit">
<input type="reset" value="clear">
</form>
</center>
</body>
</html>
```

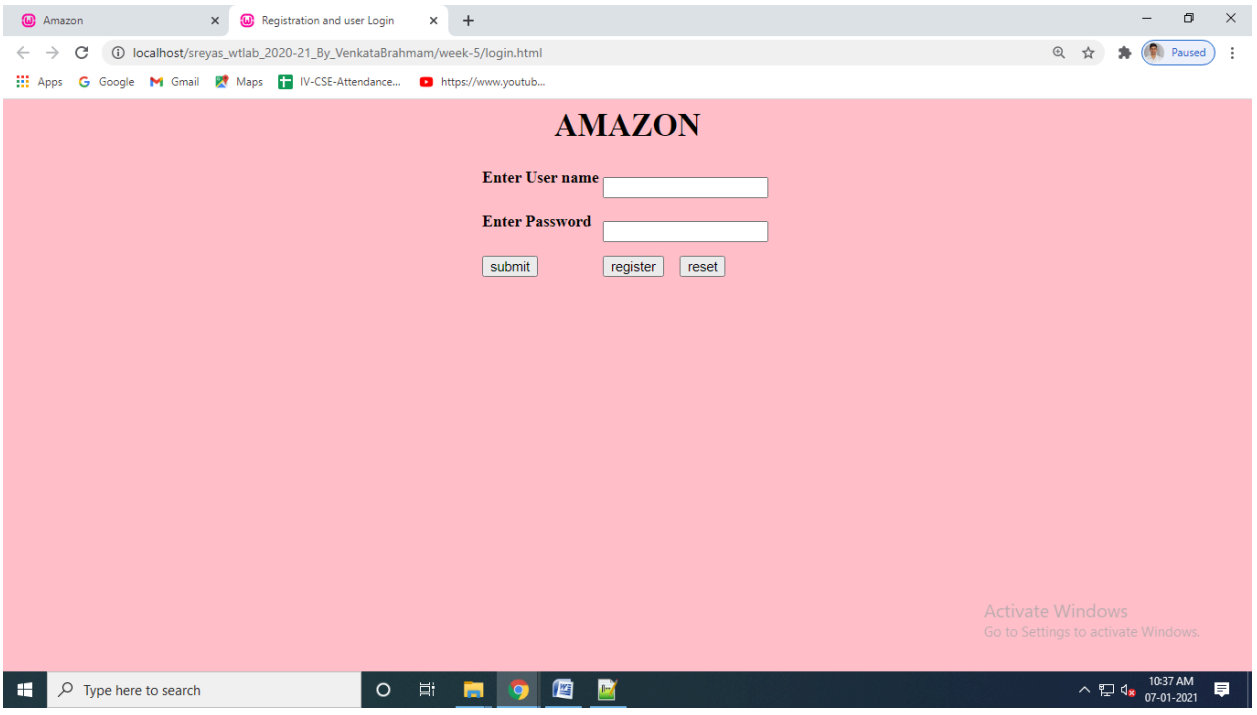
OUTPUT:



Login.html:

```
<html>
<head>
<title>
Registration and user Login</title>
</head>
<body bgcolor="pink"> <center>
<strong><h1> AMAZON </h1></strong></center>
<table align="center">
<tr>
<td><h4>Enter User name</td>
<td><input type="text" ></td>
<td></td>
</tr>
<tr>
<td><h4>Enter Password</td>
<td><input type="password"></td>
<td></td>
```

```
</tr>
<tr>
<td>
<form method="post" action="catalog.html" >
<input type="submit" value="submit" >
</form>
</td>
<td>
<form method="post" action="userprofile.html" >
<input type="submit" value="register" >
  
<input type="reset" value="reset"></form></td>
</tr>
</table>
</body>
</html>
```

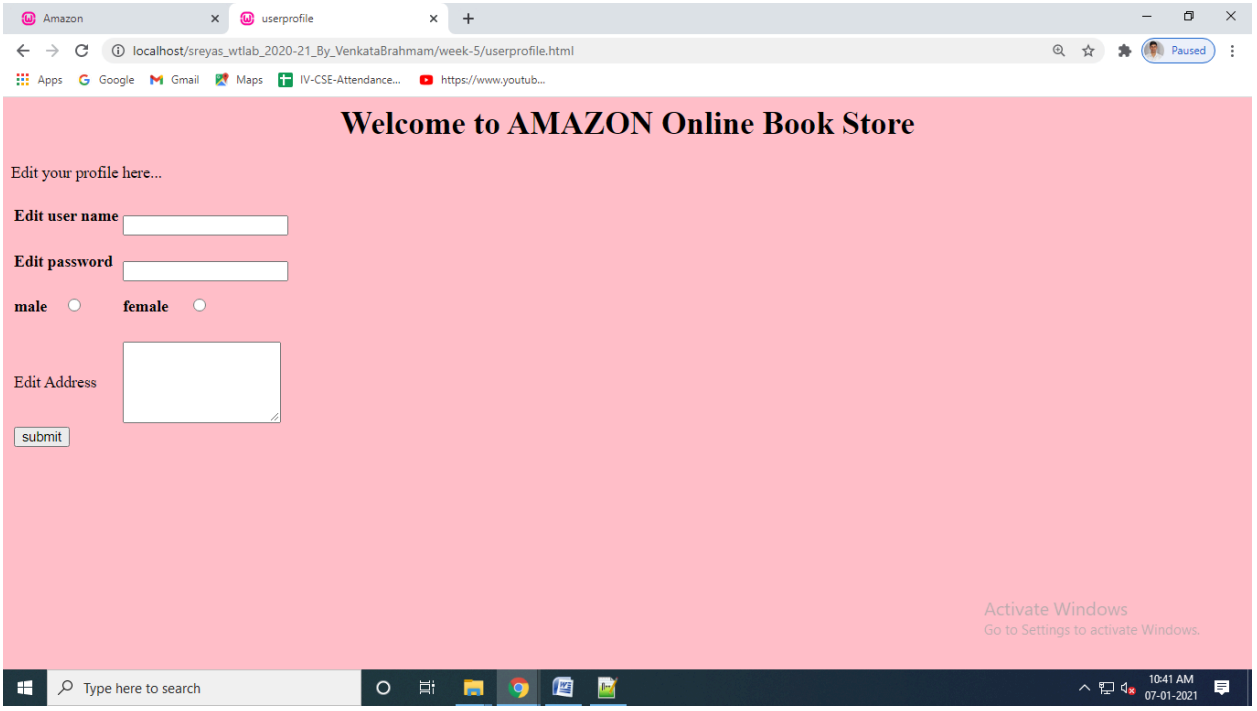


iii. User Profile Page

```
<html>
<head>
<title>
userprofile</title>
</head>
<body bgcolor="pink"> <center>
```

```
<strong><h1>Welcome to AMAZON Online Book Store </h1></strong></center>
Edit your profile here...
<form method="post" action="catalog.html" >
<right>
<table align="left">
<tr>
<td><h4>Edit user name</td>
<td><input type="text" ></td>
<tr>
<tr>
<td><h4>Edit password</td>
<td><input type="password"></td>
</tr>
<tr>
<option >
<td><h4>male   
<input type="radio" name="sex" id="male"></td>
<td><h4>female    
<input type="radio" name="sex" id="female" ></td>
</option>
</tr>
<tr>
<td>Edit Address</td>
<td><textarea name="address" rows=5 cols=19>
</textarea>
</td>
<tr>
<td>
<input type="submit" value="submit" ></td>
</tr>
</form>
</body>
</html>
```

OUTPUT:

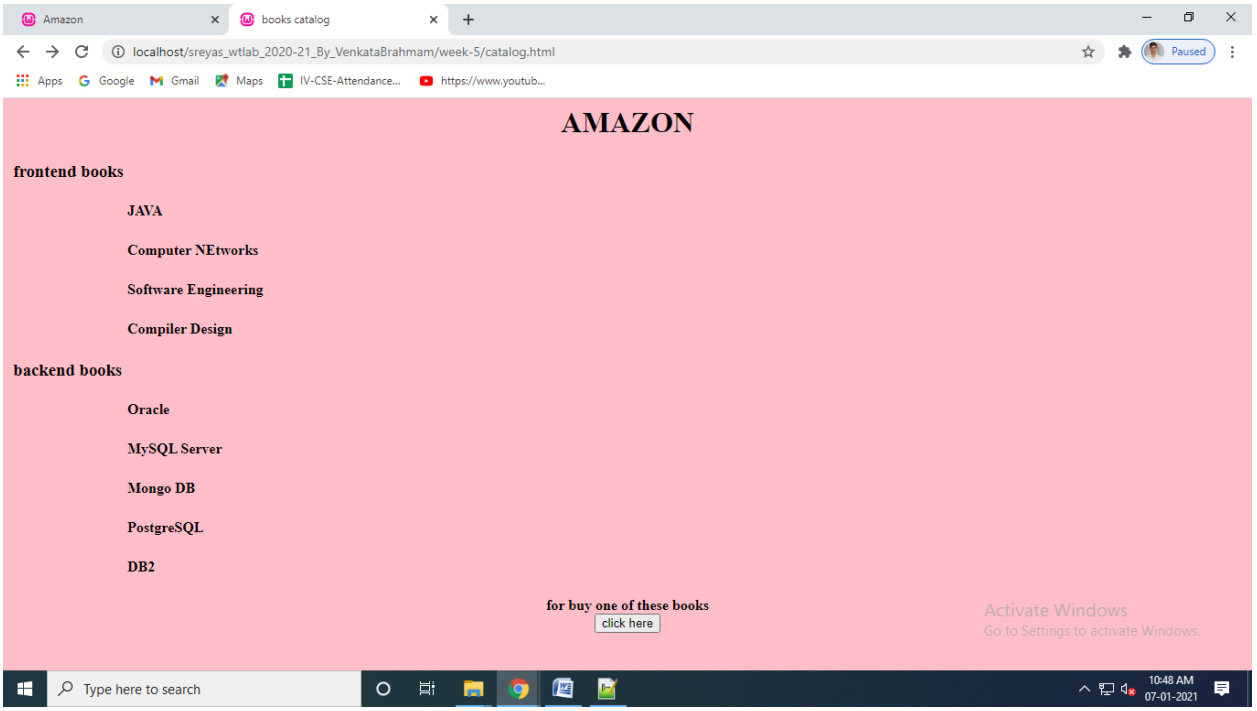


iv. Books catalog:

```
<html>
<head>
<title>
books catalog</title>
</head>
<body bgcolor="pink">
<center><h1>AMAZON</h1></center>
<form method="post" action="shopping.html">
<left>
<table>
<tr>
<td><b><h3>frontend books</td>
<td></td></tr>
<tr>
<td></td>
<td><h4>JAVA</td>
</tr>
<tr>
<td></td>
<td><h4>Computer Networks</td>
</tr>
<tr>
<td></td>
```

```
<td><h4>Software Engineering
</td></tr>
<tr>
<td></td>
<td><h4>Compiler Design
</td></tr>

<tr>
<td><b><h3>backend books</td>
<td></td>
</tr>
<tr>
<td></td>
<td><h4>Oracle</td>
</tr>
<tr>
<td></td>
<td><h4>MySQL Server
</td></tr>
<tr>
<td></td>
<td><h4>Mongo DB </td>
</tr>
<tr>
<td></td>
<td><h4>PostgreSQL</td>
</tr>
<tr>
<td></td>
<td><h4>DB2</td>
</tr>
</table>
</h4>
<center>
<b>for buy one of these books
<br>
</b><input type="submit" value="click here">
</center>
</form>
</body>
</html>
```

OUTPUT:

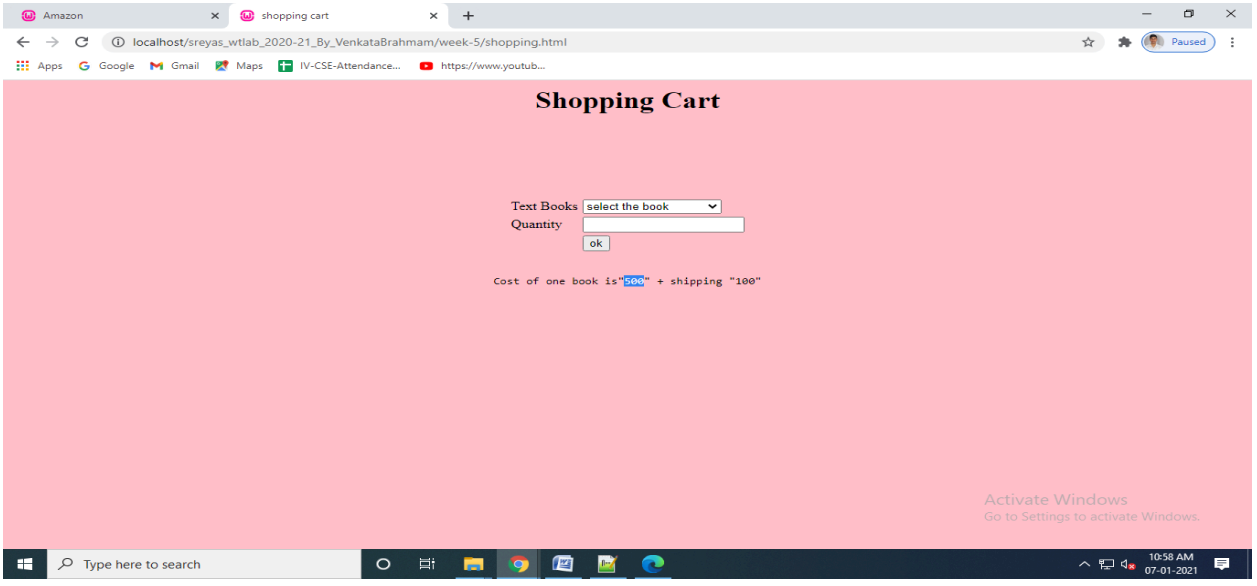


v. Shopping Cart:

```
<html>
<head><title>shopping cart</title>
</head>
<body bgcolor="pink">
<center><h1>
Shopping Cart</h1></center>
<br><br><br><br><br>
<table align="center">
<tr>
<td>Text Books</td>
<td>
<select name="My_menu" value="My_menu">

<option value="select">select the book
<option value="JAVA">JAVA
<option value="Computer Networks">Computer Networks
<option value="Software Engineering">Software Engineering
<option value="Compiler Design">Compiler Design
<option value="Oracle">Oracle
<option value="MySQL Server">MySQL Server
<option value="Mongo DB">Mongo DB
<option value="PostgreSQL">PostgreSQL
<option value="DB2">DB2
```

```
</select>
</td></tr>
<tr>
<td>
Quantity</td>
<td>
<input type="text" id="q">
</td></tr>
<tr>
<td></td>
<td>
<form method=post action="payment.html">
<input type="submit" value=ok />
</form>
</td></tr>
</table>
<center>
<pre>Cost of one book is"500" + shipping "100"</pre>
</center>
<body>
</html>
```

OUTPUT:

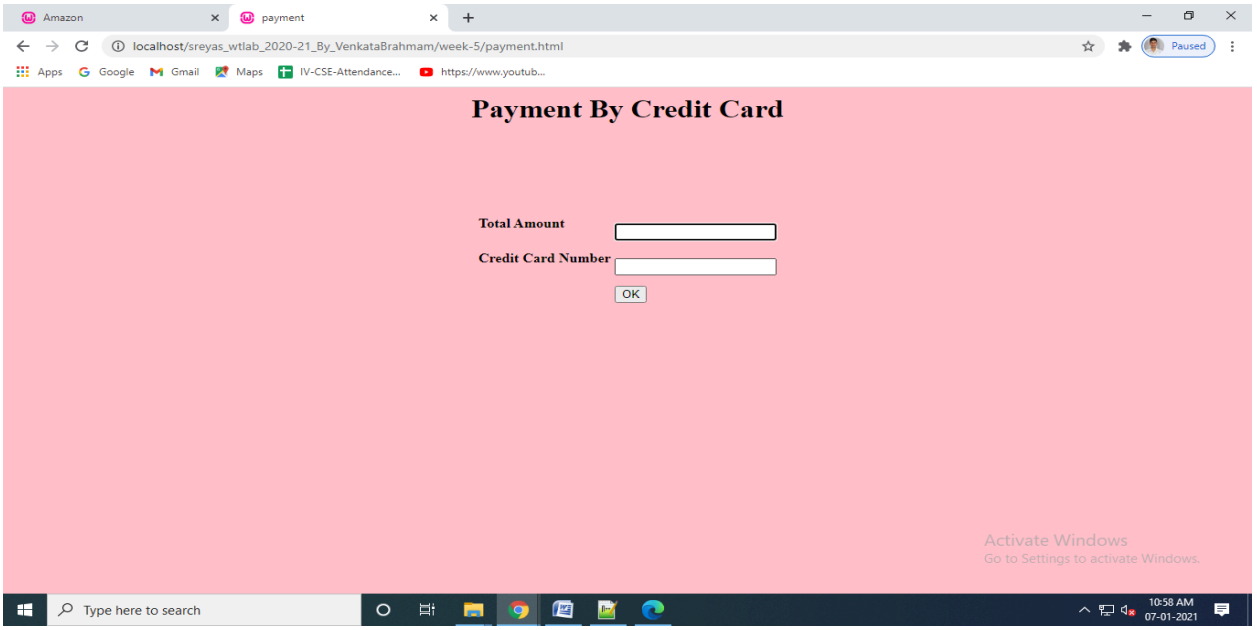


vi. Payment By credit card:

```
<html>
```

```
<head><title>payment</title></head>
<body bgcolor="pink">
<center><h1>Payment By Credit Card</h1></center>
<form method=post action="ordrconform.html">
<br><br><br><br><br>
<table align="center">
<tr>
<td>
<h4>Total Amount</h4></td>
<td><input type="text">
</td>
</tr>
<tr>
<td><h4>Credit Card Number</td>
<td><input type="text"></td>
</tr>
<tr>
<td>
</td>
<td><input type="submit" value=OK>
</td>
</tr>
</table>
</form></body>
</html>
```

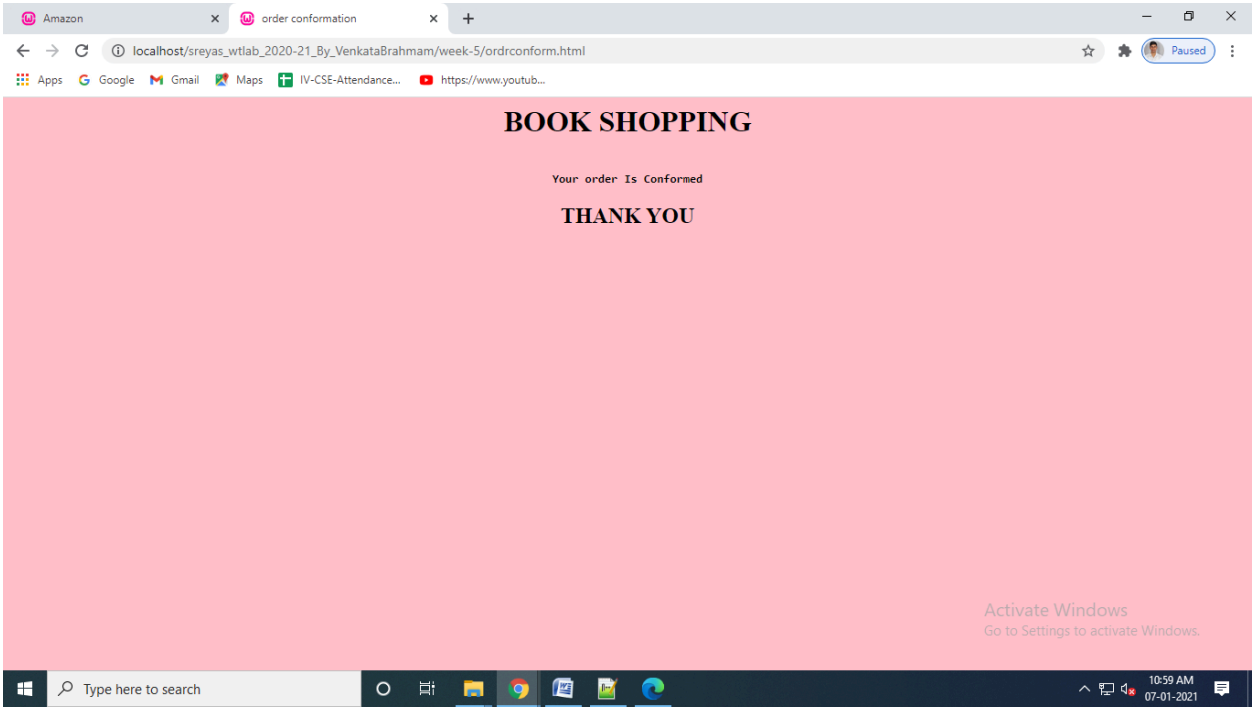
OUTPUT:



vii. Order Conformation:

```
<html>
<head><title>order conformation</title><M/head>
<body bgcolor="pink">
<center>
<h1><b>BOOK SHOPPING</h1>
<pre><strong>
<b>Your order Is Conformed
</strong></pre>
<h2><b>THANK YOU</h2>
</center>
</body></html>
```

OUTPUT:



6. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

SOURCE CODE:

Main.html:

```
<html>
<head>
        <title>
                Home Page
        </title>
</head>
<frameset rows="30%,70%">
        <frame src="top.html" name="top">
<frameset cols="30%,70%">
        <frame src="left.html" name="left">
        <frame src="right.html" name="right">
</frameset>
</frameset>
</html>
```

Top.html:

```
<html>
<head>
<title>
Amazon</title>
</head>
<body bgcolor="pink"> <center>
<strong><h1>Welcome to AMAZON</h1></strong>
<br><br>
<h1 align=”center”><b><u>ONLINE BOOK STORAGE</u></b></h1>
</body>
</html>
```

Right.html:

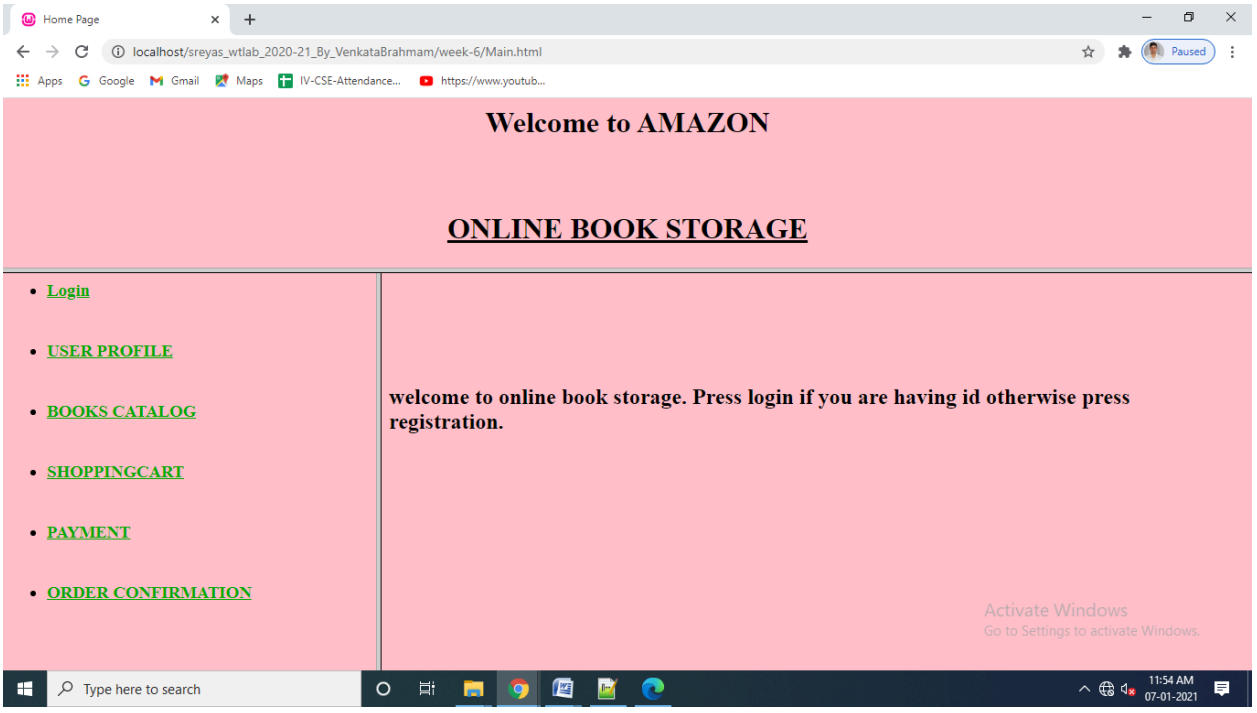
```
<html>
<body bgcolor="pink">
<br><br><br><br><br>
<h2 align=”center”>
<b><p> welcome to online book storage. Press login if you are having id otherwise press registration.
</p></b></h2>
</body></html>
```

Left.html:

```
<html>
<body bgcolor="pink">
<h3>
<ul>
<li><a href="login.html" target="right">
        <font color=”black”>Login </font>
        </a>
</li><br><br>
<li>
        <a href="profile.html" target="right">
          <font color=”black”> USER PROFILE </font>
        </a>
</li><br>
<br>
<li>
        <a href="catalog.html" target="right">
          <font color=”black”> BOOKS CATALOG</font>
          </a>
</li>
<br><br>
<li>
        <a href="scart.html" target="right">
          <font color=”black”> SHOPPINGCART</font>
        </a>
</li><br>
<br>
<li>
        <a href="payment.html" target="right">
          <font color=”black”> PAYMENT</font>
        </a>
</li><br><br>
<li>
        <a href="order.html" target="right">
          <font color=”black”> ORDER CONFIRMATION</font>
        </a>
</li><br><br>
</ul>
</body>
```

```
</html>
```

OUTPUT:



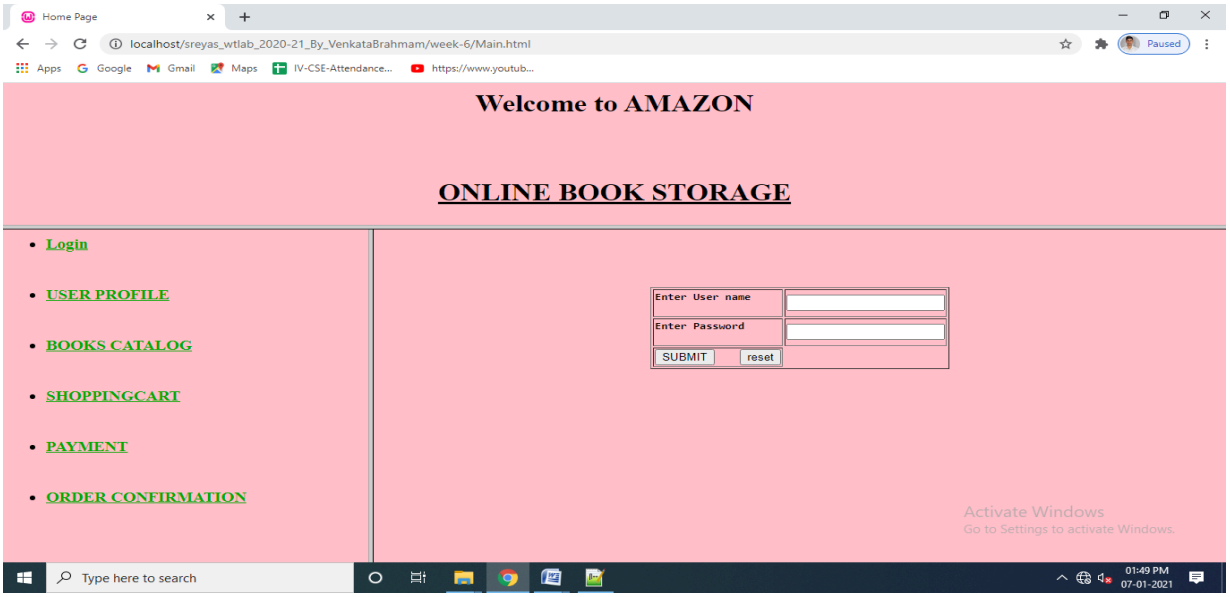
Registration and user Login

Login.html:

```
<html>
<head>
<title>
User Login</title>
</head>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.id.value==""||
document.myform.pwd.value=="")
{
flag=0;
}
if(flag==1)
```

```
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}
}
</script>
<form name="myform">
<div align="center"><pre>
<table align="center" border="1">
<tr>
<td><h4>Enter User name</td>
<td><input type="text" ></td>

</tr>
<tr>
<td><h4>Enter Password</td>
<td><input type="password"></td>
</tr>

<tr colspan="2" align="center">
<td >

<input type="submit" value="SUBMIT" onClick="validate()">
  
<input type="reset" value="reset">
</td>
</tr>
</table>
</form>
</body>
</html>
```

OUTPUT:



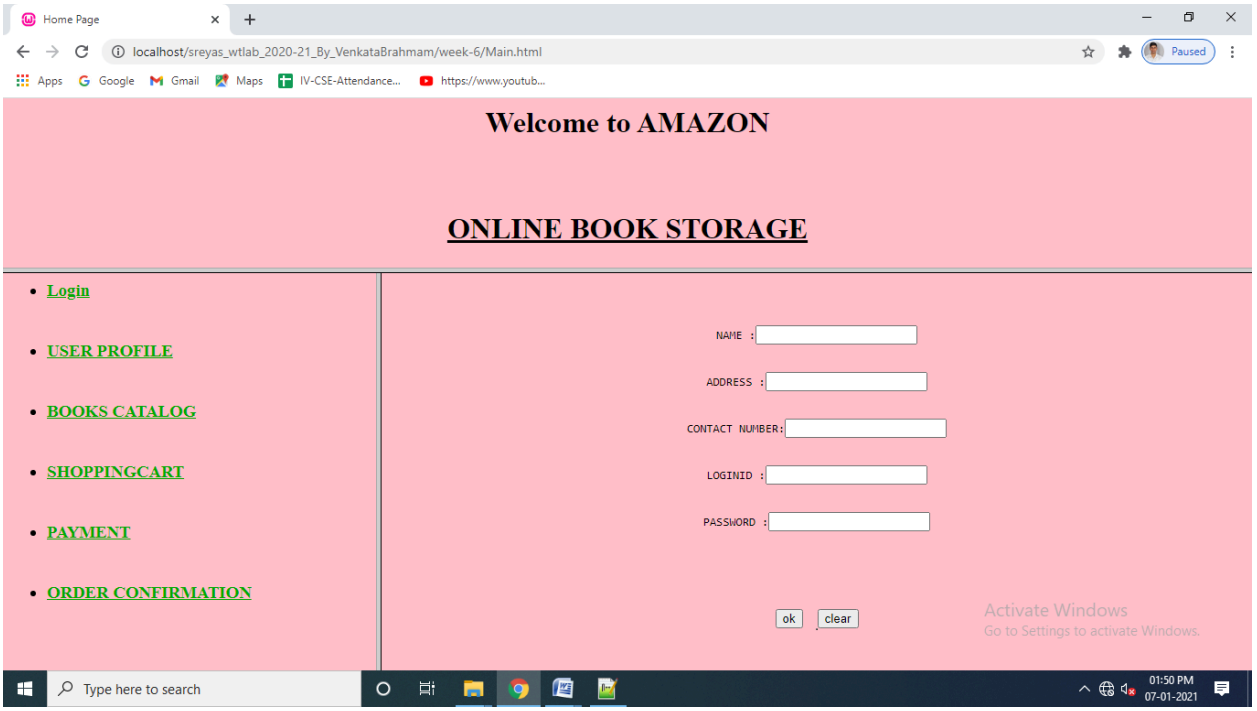
User profile page

Profile.html:

```
<html>
<body bgcolor="pink"><br><br>
<script language=”javascript”>
function validate()
{
var flag=1;
if(document.myform.name.value==””||
document.myform.addr.value==””||
document.myform.phno.value==””||
document.myform.id.value==””||
document.myform.pwd.value==””)
{
flag=0;
}
var str=document.myform.phno.value;
var x;
for(var i=0;i<str.length;i++)
{
x=str.substr(i,1)
if(!(x<=9))
{
flag=0;
break;
}
}
if(flag==1)
```

```
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}}
</script>
<table align="center"border="1">
<form name="myform">
<div align="center"><pre>
NAME :<input type="text" name="name"><br><br>
ADDRESS :<input type="type" name="addr"><br><br>
CONTACT NUMBER:<input type="text" name="phno"><br><br>
LOGINID :<input type="text" name="id">
<br>
PASSWORD :<input type="password" name="pwd"></pre><br><br>
</div>
<br><br>
<div align="center">
<input type="submit" value="ok" onClick="validate()">   
<input type="reset" value="clear">
</form>
</table>
</body></html>
```

OUTPUT:



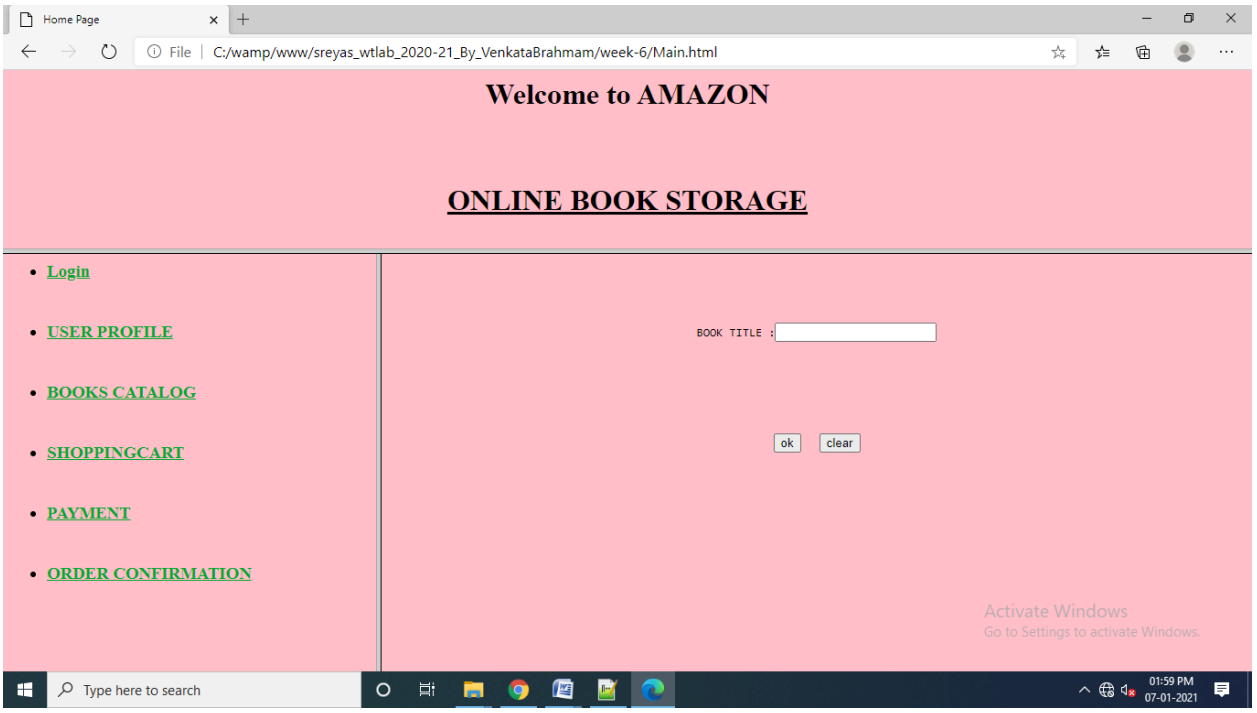
Books catalog :

Scart.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
```

```
if(document.myform.title.value=="")
{
flag=0;
}
str=document.myform.title.value;
if(str=="c")
{
document.writeln("<body bgcolor=pink>");
document.writeln("title-->c"+" cost-->444");
}
else if(str=="jsp")
{
document.writeln("<body bgcolor=pink>");
document.writeln("title-->jsp"+" cost-->555");
}
else
{
flag=0;
}
if(flag==1)
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}
}
</script>
<form name="myform" >
<div align="center"><pre>
BOOK TITLE :<input type="text" name="title"><br>
</pre><br><br>
</div>
<br><br>
<div align="center">
<input type="submit" value="ok" onClick="validate()">    
<input type="reset" value="clear">
</form>
</body>
</html>
```

OUTPUT:



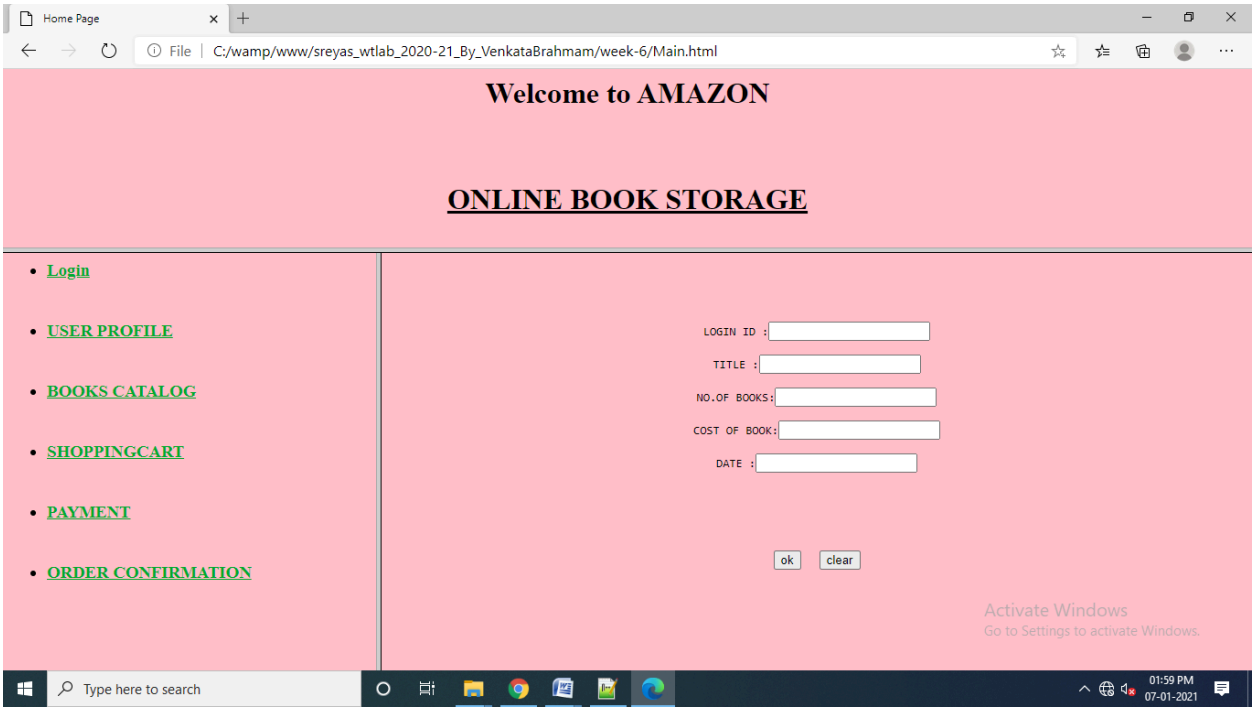
Shopping cart:

Catalog.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.id.value==""||
document.myform.title.value==""||
document.myform.no.value==""||
document.myform.cost.value==""||
document.myform.date.value=="")
{
flag=0;
}
var str=document.myform.no.value;
var x;
for(var i=0;i<str.length;i++)
{
x=str.substr(i,1)
if(!(x<=9))
{
flag=0;
break;
}
}
str=document.myform.title.value;
var str1=document.myform.cost.value;
if(!((str=="c"&& str1==444) || (str=="jsp" && str1==555)))
{
flag=0;
}
if(flag==1)
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}
}
```

```
</script>
<form name="myform" >
<div align="center"><pre>
LOGIN ID :<input type="text" name="id"><br>
TITLE :<input type="text" name="title"><br>
NO.OF BOOKS:<input type="text" name="no"><br>
COST OF BOOK:<input type="text"name="cost"><br>
DATE :<input type="text" name="date"><br></pre><br><br>
</div>
<br><br>
<div align="center">
<input type="submit" value="ok" onClick="validate()">    
<input type="reset" value="clear">
</form>
</body>
</html>
```

OUTPUT:



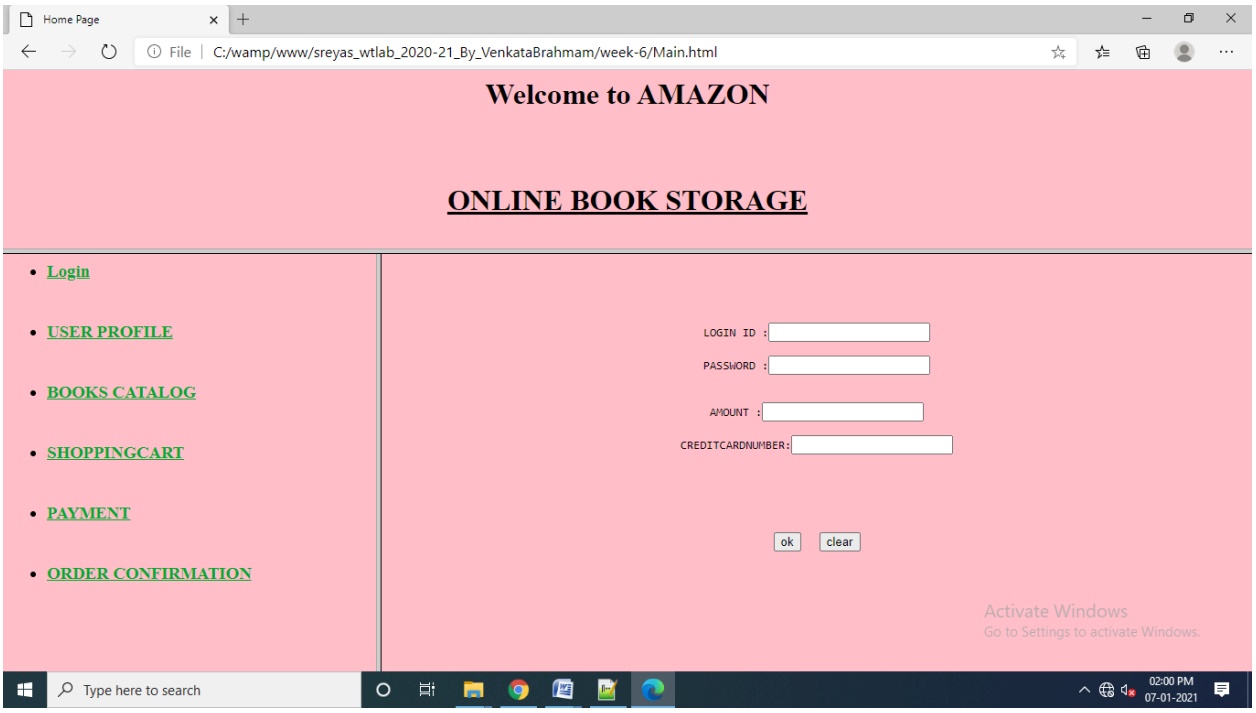
Payment by credit card

Payment.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.id.value==""||
document.myform.pwd.value==""||
document.myform.amount.value==""||
document.myform.num.value=="")
{
flag=0;
}
var str=document.myform.amount.value;
var x;
for(var i=0;i<str.length;i++)
{
x=str.substr(i,1);
if(!(x<=9))
{
flag=0;
break;
}
}
str=document.myform.num.value;
for(var i=0;i<str.lenght;i++)
{
x=str.substr(i,1);
if(!(x<=9))
{
flag=0;
break;
}
}
if(flag==1)
```

```
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}
}
</script>
<form name="myform">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id"><br>
PASSWORD :<input type="password" name="pwd">
<br>
AMOUNT :<input type="text" name="amount"><br>
CREDITCARDNUMBER:<input
type="PASSWORD"
name="num+"><br></pre><br><br>
</div>
<br><br>
<div align="center">
<input type="submit" value="ok" onClick="validate()">    
<input type="reset" value="clear" >
</form>
</body>
</html>
```

OUTPUT:



Order Conformation

Order.html:

```
<html>
<head><title>order conformation</title><M/head>
<body bgcolor="cyan">
<center>
<h1><b>AMAZON</h1>
<pre><strong>
<b>Your order Is Conformed
</strong></pre>
<h2><b>THANK YOU</h2>
</center>
</body>
</html>
```

OUTPUT:

7. Create and save an XML document on the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

SOURCE CODE:

```
import java.io.File;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.util.Scanner;
public class Exp5aDOMParser{
public static void main(String[] args) throws Exception {

DocumentBuilderFactory fac = DocumentBuilderFactory.newInstance();

DocumentBuilder b = fac.newDocumentBuilder();
```

```
Document doc = b.parse(new File("users.xml"));
doc.getDocumentElement().normalize();
Element root = doc.getDocumentElement();
Scanner in = new Scanner(System.in);
System.out.println("Enter User ID:");

int n = in.nextInt();
int flag = 0;
NodeList nl = doc.getElementsByTagName("user");
for(int i = 0; i < nl.getLength(); i++)
{
Node node = nl.item(i);
if(node.getNodeType() == Node.ELEMENT_NODE)
{
Element e = (Element) node;
int x =

Integer.parseInt(e.getElementsByTagName("userid").item(0).getTextcontent());
if(x==n)
{
System.out.println(root.getNodeName());
System.out.println("==========================================");
System.out.println("User id :" +
e.getElementsByTagName("userid").item(0).getTextcontent());

System.out.println("User Name :" +
e.getElementsByTagName("name").item(0).getTextcontent());

System.out.println("Adress :" +
e.getElementsByTagName("address").item(0).getTextcontent());

System.out.println("Gender :" +
e.getElementsByTagName("Gender").item(0).getTextcontent());
flag=1;
break;
}
else
{
flag=0;
}
}
}
if(flag==0)
System.out.println("User Id is not present.Try Again!!!");
}
}
```

users.xml:

```
<users-details>
<user>
<userid>1111</userid>
<name>Sammulal</name>
<address>Hyderabad</address>
<gender>Male</gender>
</user>

<user>
<userid>1112</userid>
<name>Sanjana</name>
<address>Banglore</address>
<gender>Female</gender>
</user>

<user>
<userid>1113</userid>
<name>kishor</name>
<address>karimnagar</address>
<gender>Male</gender>
</user>

<user>
<userid>1114</userid>
<name>Srindhar</name>
<address>USA</address>
<gender>Male</gender>
</user>

<user>
<userid>1115</userid>
<name>Rajitha</name>
<address>Hyderbad</address>
<gender>Female</gender>
</user>

<user>
<userid>1116</userid>
<name>Aradhya</name>
<address>Hyderabad</address>
<gender>Female</gender>
</user>

<user>
<userid>1117</userid>
```

```
<name>Vikram</name>
<address>Ludan</address>
<gender>Male</gender>
</user>

<user>
<userid>1118</userid>
<name>SaiKrishna</name>
<address>Pune</address>
<gender>Male</gender>
</user>

<user>
<userid>1119</userid>
<name>Deeksha</name>
<address>NewYork</address>
<gender>Female</gender>
</user>

<user>
<userid>1120</userid>
<name>Charan</name>
<address>Chennai</address>
<gender>Male</gender>
</user>
</users-details>
```

OUTPUT:

```
D:\madhucse\labprograms>javac Exp5aDOMParser.java

D:\madhucse\labprograms>java Exp5aDOMParser
Enter User ID:
1111
users-details
================================================
User id :1111
User Name :Sammulal
Adress :Hyderabad
Gender :Male
```

8. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

PROCEDURE:

1. First install the tomcat into the system.
2. Then make a subdirectly(eg., tr) in the \tomcat\webapps.
3. Under tr create WEB-INF directory and also place the html files in this tr directory only.
4. Next under WEB-INF create two subclasses lib,classes and web.xml
5. Next place all the class files under the classes and jar files(servlet-api.jar,classes12.jar etc…) under lib
6. subdirectories.
7. After this start tomcat by giving the following command at the instll_dir>tomcat>bin Catalina.bat run
8. At the I.E(web browser) give the url as http;//localhost:8080//tr/htmlfile or servlet url pattern
9. Portno 8080 is assigned for the tomcat.

APACHE TOMCAT INSTALLATION

Step 1:

Installation of JDK:

Before beginning the process of installing Tomcat on your system, ensure first the availability of JDK on your system program directory. Install it on your system if not already installed (because any version of tomcat requires the Java 1.6 or higher versions) and then set the class path (environment variable) of JDK. To set the **JAVA_HOME Variable:** you need to specify the location of the java run time environment to support the Tomcat else Tomcat server can not run.

This variable contains the path of JDK installation directory.

```
set JAVA_HOME=C:\Program Files\Java\jdk1.6
```

Note: it should not contain the path up to bin folder. Here, we have taken the URL path according to our installation convention.

For Windows OS, go through the following steps:

Start menu->Control Panel->System->Advanced tab->Environment Variables->New->set the Variable name = JAVA_HOME and variable value = C:\Program Files\Java\jdk1.6

Now click on all the subsequent ok buttons one by one. It will set the JDK path.

Step 2:

For setting the class path variable for JDK, do like this:

Start menu->Control Panel->System->Advanced tab->Environment Variables->New->
Set PATH="C:\Program Files\Java\jdk1.6\bin"; %PATH%
OR

First, right click on the

My Computer->properties->advance->Environment Variables->path.

Now, set bin directory path of JDK in the path variable

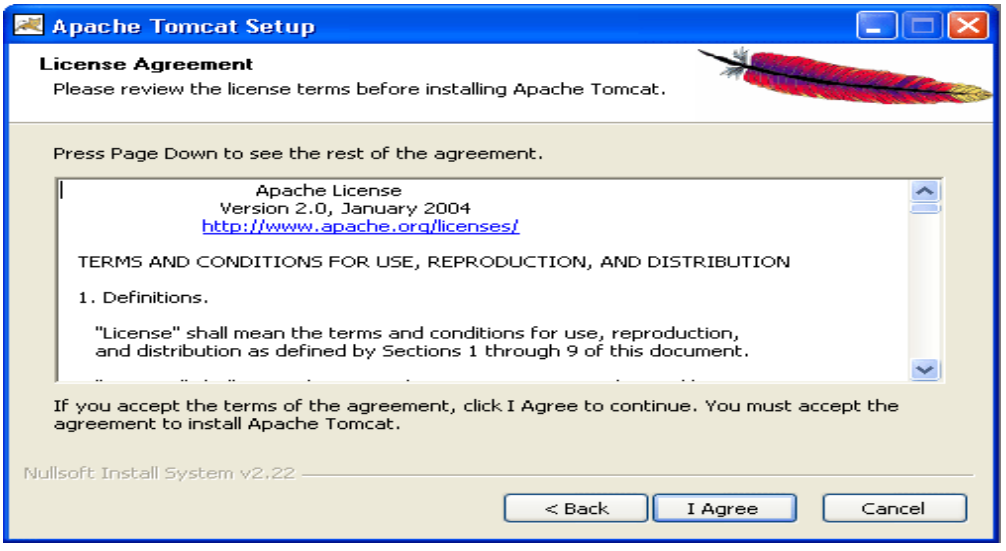
Step 3:

The process of installing Tomcat 6.0 begins here from now. It takes various steps for installing and configuring the Tomcat 6.0.

For Windows OS, Tomcat comes in two forms: .zip file and .exe file (the Windows installer file). Here we are exploring the installation process by using the .exe file. First unpack the zipped file and simply execute the '.exe' file.

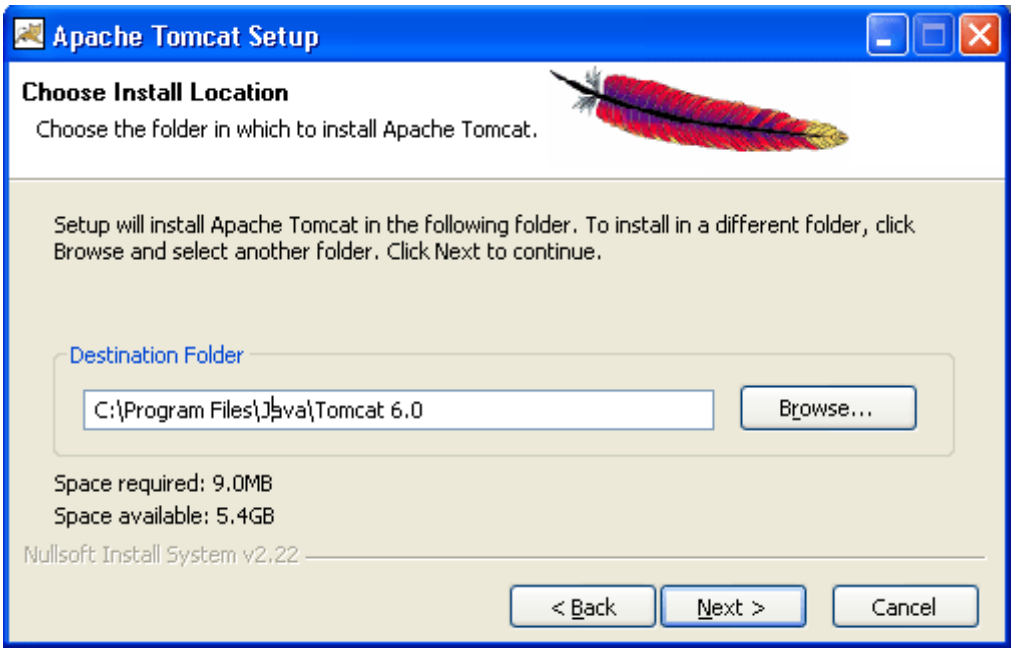
A Welcome screen shot appears that shows the beginning of installation process. Just click on the 'Next' button to proceed the installation process.

Steps 4: A screen of 'License Agreement' displays.



Click on the 'I Agree' button.

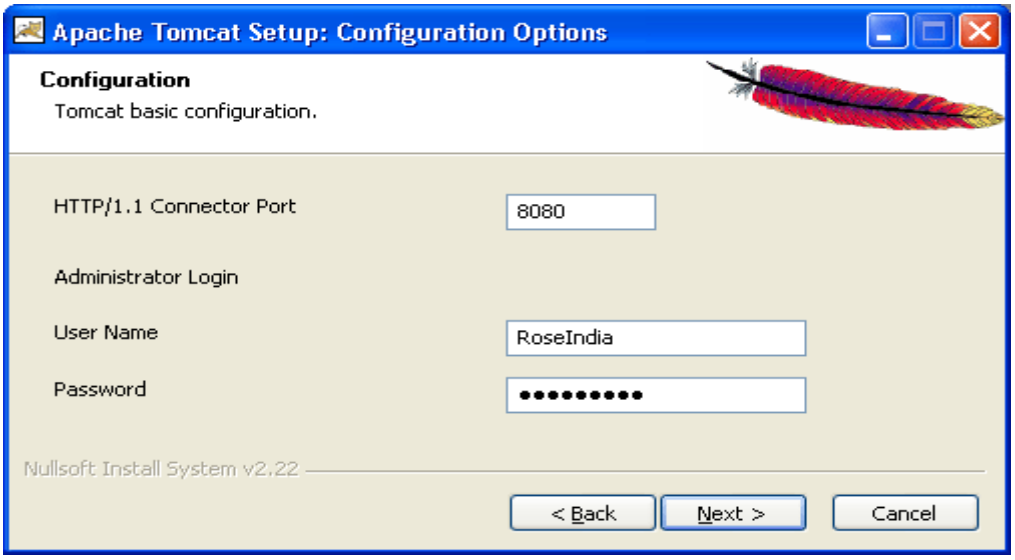
Step 5: A screen shot appears asking for the 'installing location'



Choose the default components and click on the 'Next' button.

Step 6:

A screen shot of 'Configuration Options' displays on the screen. Choose the location for the Tomcat files as per your convenience. You can also opt the default Location

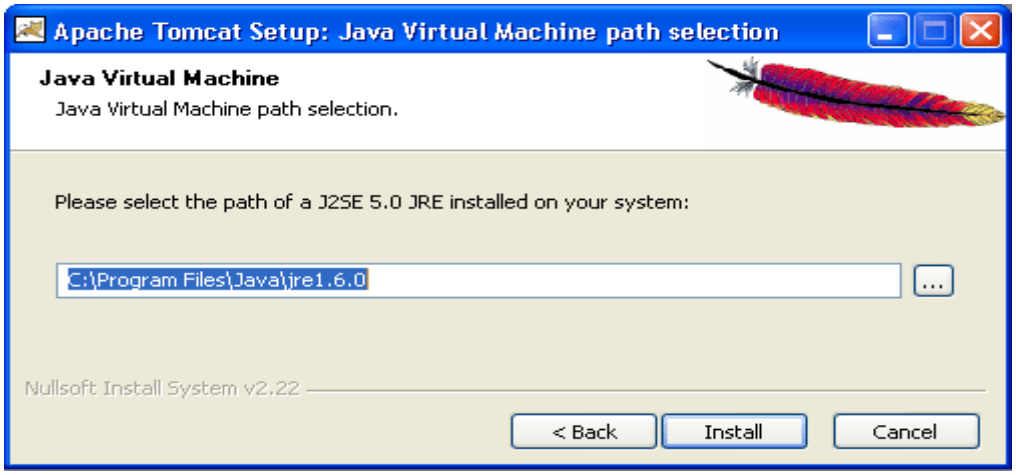


The port number will be your choice on which you want to run the tomcat server. The port number 8080 is the default port value for tomcat server to proceed the HTTP requests. The user can also change the 'port number' after completing the process of installation; for this, users have to follow the following tips.

Go to the specified location as " **Tomcat 6.0 \conf \server.xml** ". Within the server.xml file choose "Connector" tag and change the port number.

Now, click on the 'Next' button to further proceed the installation process.

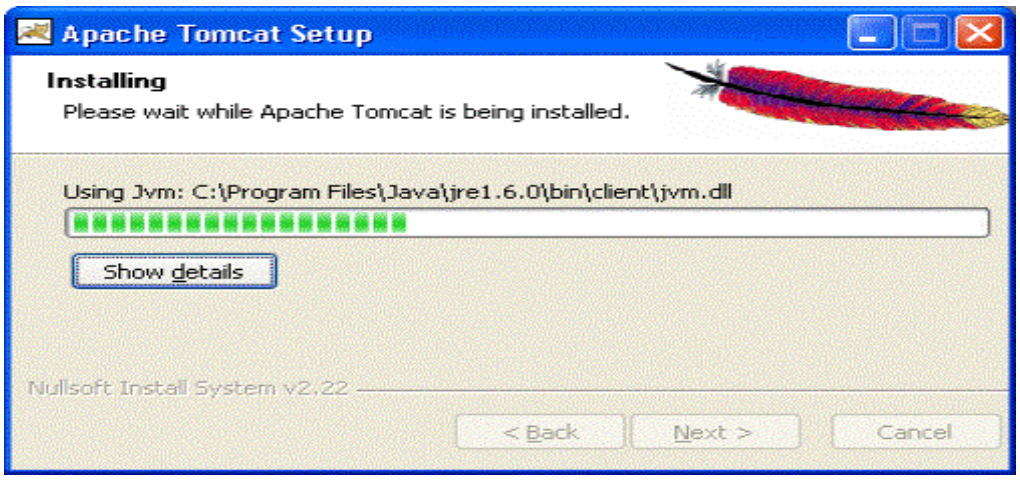
Step 7: A Window of Java Virtual Machine displays on the screen



This window asks for the location of the installed Java Virtual Machine. Browse the location of the JRE folder and click on the Install button. This will install the Apache tomcat at the specified location.

Step 8:

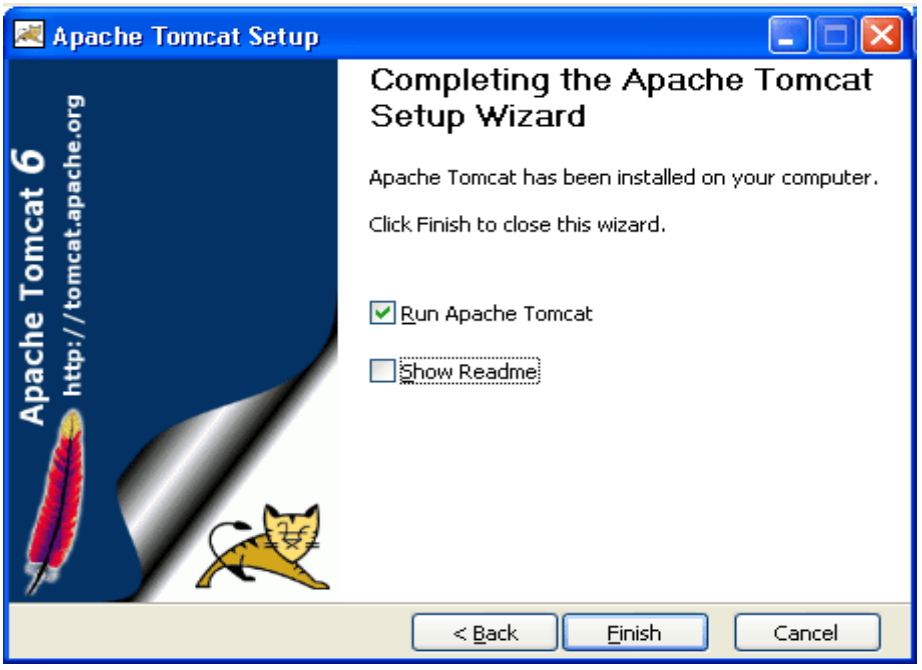
A processing window of installing displays on the screen.



To get the information about installer click on the "Show details" button

Step 9:

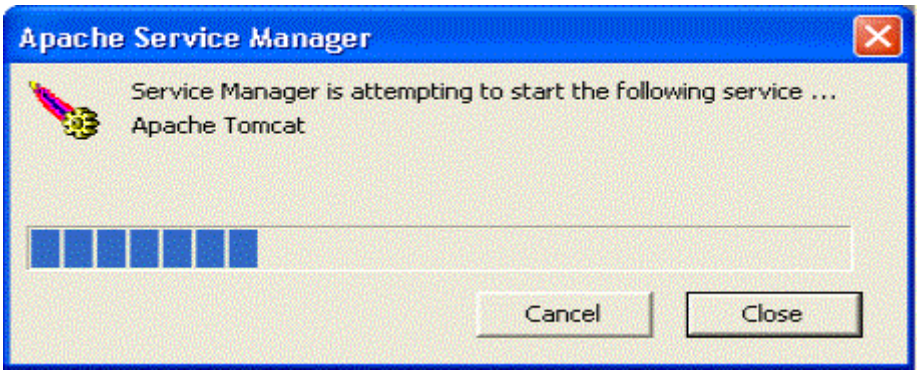
A screen shot of 'Tomcat Completion' displays on the screen.



Click on the 'Finish' button.

Step 10:

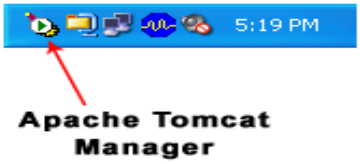
A window of Apache Service Manager appears with displaying the running process.



Let the running process goes on.

Step 11:

After completing the installation process, the Apache Tomcat Manager appears on the toolbar panel like shown in the below picture.

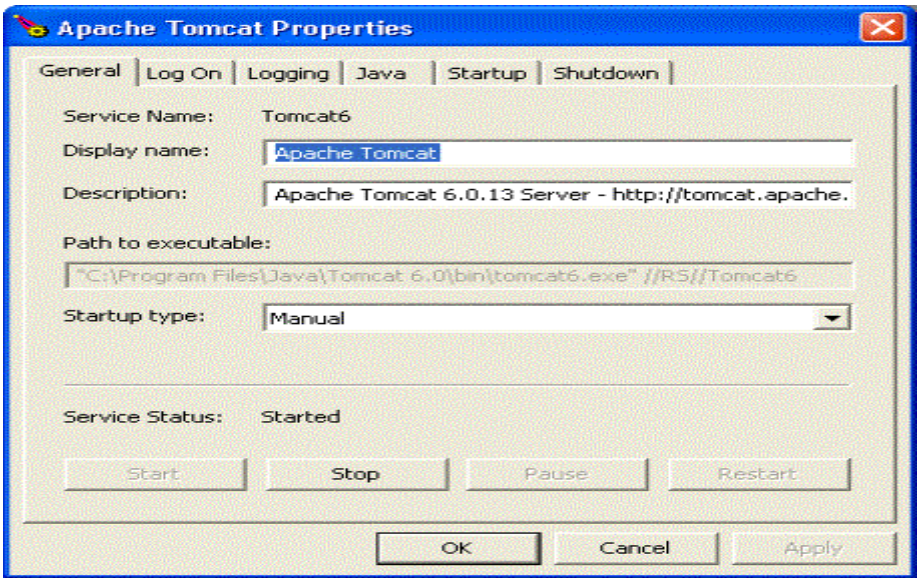


Configuring Tomcat Manager

To Configure the Tomcat Manager, there are two ways; either user can configure Tomcat directly from the toolbar panel or can configure it from Control Panel Section.

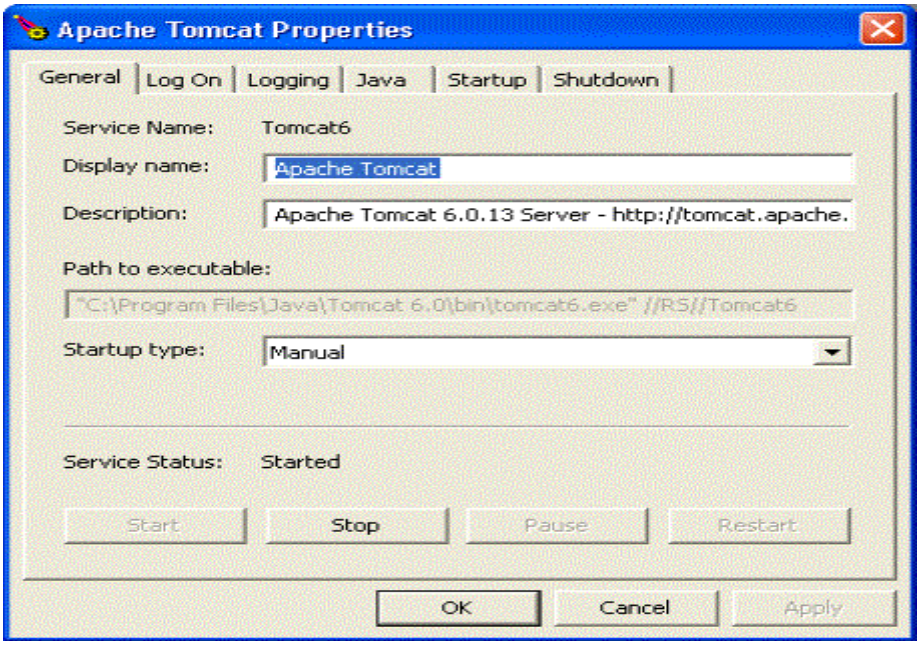
i-) Configuring from toolbar Panel

To Configure Apache Tomcat web server from the toolbar panel, you have to press 'double click' on the appeared icon.



A configured window appears on the desktop. Now, just click on the Startup button. The installation process will be completed.

Double click on the Apache Tomcat. The window of Apache Tomcat Properties appears on the screen.



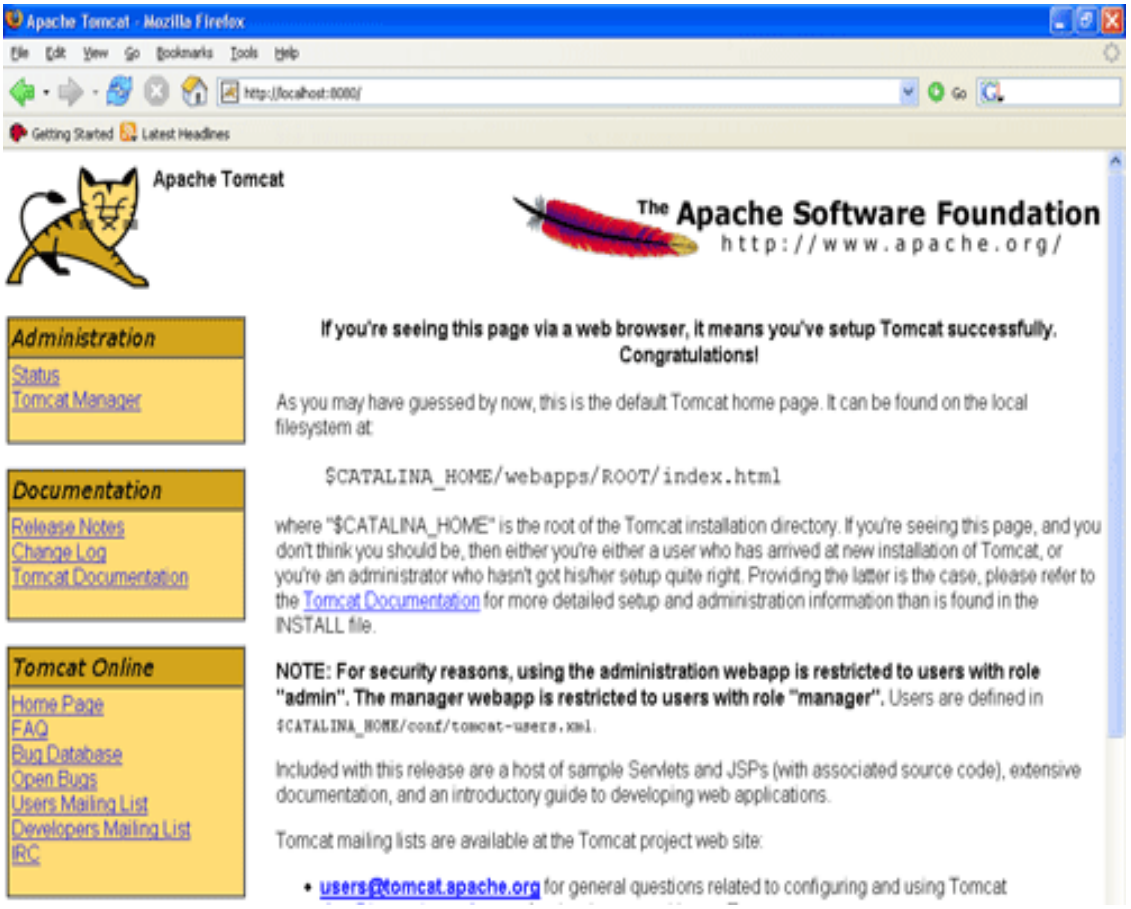
Now, Click on the start up button. The Apache Tomcat is now ready to function.

To operate it, follow the below steps of processing.

Start the Tomcat Server:

1. Start the tomcat server from the bin folder of Tomcat 6.0 directory by double clicking the "tomcat6.exe" file.
   OR create a shortcut of this .exe file at your desktop.
2. 2. Now Open web browser and type URL http://localhost:8080 in the address bar to test the server.
3. 3. To Stop the Tomcat Server: Stop the server by pressing the "Ctrl + c" keys.

The screen of Apache Tomcat software looks like this:



Home page:

Main.html:

```
<html>
<body>
<br /><br /><br /><br />
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1><br /><br /><br />
<h2 align="center"><pre>
<b>Welcome to online book storage.
Press LOGIN if you are having id
otherwise press REGISTRATION
</b></pre></h2>
<br /><br /><pre>
<div align="center"><a href="login.html">LOGIN</a><a href="reg.html">
REGISTRATION</a></div>
```

```
</pre>
</body>
</html>
```

Login page:

Login.html:

```
<html>
<body><br /><br /><br />
<form name="myform" method="post" action="login">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br /> PASSWORD :<input type="password"
name="pwd"
/></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Registration page:

Reg.html:

```
<html>
<body><br /><br />
<form name="myform" method="post" action="reg">
<table align="center" >
<tr>
<td>NAME</td>
<td>:<input type="text" name="name" /></td>
</tr>
<tr>
<td>ADDRESS</td>
<td>:<input type="text" name="addr" /></td>
</tr>
<tr>
<td>CONTACT NUMBER</td>
<td>:<input type="text" name="phno" /></td>
</tr>
<tr>
<td>LOGINID</td>
<td>:<input type="text" name="id" /></td>
</tr>
<tr>
<td>PASSWORD</td>
```

```
<td>:<input type="password" name="pwd" /></td>
</tr>
</table>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

User profile page:

Profile.html:

```
<html>
<body><br /><br /><br />
<form name="myform" method="post" action="profile">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Books catalog page:

Catalog.html:

```
<html>
<body><br /><br /><br />
<form method="post" action="catalog">
<div align="center"><pre>
BOOK TITLE :<input type="text" name="title" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"
name="button1"/>      
<input type="reset" value="clear" name="button2"/>
</div>
</form>
</body>
</html>
```

Shopping cart, Payment by credit card, Order

Conformation page:

Order.html:

```
<html>
<body><br /><br />
<form method="post" action="order">
<div align="center"><pre>
ID
:<input type="text" name="id" /><br />
PASSWORD
:<input type="password" name="pwd" /><br/> TITLE
:<input type="text" name="title" /><br /> NO. OF BOOKS
:<input type="text" name="no" /><br />
DATE
:<input type="text" name="date" /><br />
CREDIT CARD NUMBER :<input type="password" name="cno" /><br /></pre><br
/><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" name="button1"/>
      <input type="reset" value="clear"
name="button2"/>
</div>
</form></body></html>
```

Login servlet :

Login.java:

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body>");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger
");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
```

```
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
pw.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
pw.println("<a href=\"login.html\">press LOGIN to
RETRY</a>");
}
else
{
pw.println("<br><br>VALID LOGIN ID<br><br>"); 58
pw.println("<h3><ul>");
pw.println("<li><a
href=\"profile.html\"><fontcolor=\"black\">USER
PROFILE</font>
</a></li><br><br>");
pw.println("<li><a
href=\"catalog.html\"><fontcolor=\"black\">BOOKS
CATALOG</font></a></li><br><br>");
pw.println("<li><a
href=\"order.html\"><fontcolor=\"black\">ORDER
CONFIRMATION</font>
</a></li></ul><br><br>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Registration servlet:

Reg.java :

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class reg extends HttpServlet
```

```
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
resp.setContentType("text/html");
pw.println("<html><body>");
String name=req.getParameter("name");
String addr=req.getParameter("addr");
String phno=req.getParameter("phno");
String id1=req.getParameter("id");
String pwd1=req.getParameter("pwd");
int no=Integer.parseInt(phno);
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger
");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id1.equals(rs.getString(1))&&pwd1.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==1)
{
pw.println("<br><br>SORRY INVALID ID ALREADY
EXITS TRY AGAIN WITH NEW ID<br><br>");
pw.println("<a href=\"reg.html\">press REGISTER to
RETRY</a>");
}
else
{
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insert into login
values('"+name+"','"+addr+"',"+no+",'"+id1+"','"+pwd1+"')
;");
pw.println("<br><br>YOUR DETAILS ARE
ENTERED<br><br>");
pw.println("<a href=\"login.html\">press LOGIN to
login</a>");
```

```
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Profile servlet:

Profile.java:

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class profile extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body>");
String id=req.getParameter("id");
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger
");
Statement stmt=con.createStatement();
String sqlstmt="select * from login where id="+id+""; ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
pw.println("<br><br><br>");
while(rs.next())
{
pw.println("<div align=\"center\">");
pw.println("NAME :"+rs.getString(1)+"<br>");
pw.println("ADDRESS
:"+rs.getString(2)+"<br>");
pw.println("PHONE NO
:"+rs.getString(3)+"<br>");
pw.println("</div>");
flag=1;
}
if(flag==0)
{
```

```
pw.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
pw.println("<a href=\"profile.html\">press HERE to
RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Catalog servlet:

Catalog.java:

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class catalog extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body>");
String title=req.getParameter("title");
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger
");
Statement stmt=con.createStatement();
String sqlstmt="select * from book where title=\'"+title+"\'"; ResultSet
rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
pw.println("<div align=\"center\">");
pw.println("TITLE:"+rs.getString(1)+"<br>");
pw.println("AUTHOR :"+rs.getString(2)+"<br>"); pw.println("VERSION
:"+rs.getString(3)+"<br>");
pw.println("PUBLISHER :"+rs.getString(4)+"<br>");
pw.println("COST:"+rs.getString(5)+"<br>");
pw.println("</div>");
flag=1;
```

```
}
if(flag==0)
{
pw.println("<br><br>SORRY INVALID TITLE TRY
AGAIN <br><br>");
pw.println("<a href=\"catalog.html\">press HERE to
RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Order servlet:

Order.java:

```
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class order extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
int count;
PrintWriter pw=resp.getWriter();
pw.println("<html><body>");
String id=req.getParameter("id");
String pwd=req.getParameter("pwd");
String title=req.getParameter("title");
String count1=req.getParameter("no");
String date=req.getParameter("date");
String cno=req.getParameter("cno");
try
{
count=Integer.parseInt(count1);
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger
");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0,amount,x;
```

```
while(rs.next())
{
if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
pw.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
pw.println("<a href= \" order.html \" >press HERE to RETRY</a>");
}
else
{
Statement stmt2=con.createStatement();
String s="select cost from book where title=\'"+title+"\'"; ResultSet rs1=stmt2.executeQuery(s);
int flag1=0;
while(rs1.next())
{
flag1=1;
x=Integer.parseInt(rs1.getString(1));
amount=count*x;
pw.println("<br><br>AMOUNT
:"+amount+"<br><br><br><br>");
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insert into details
values('"+id+"','"+title+"',"+amount+",'"+cno+"');"); pw.println("<br>YOUR ORDER has
taken<br>");
}
if(flag1==0)
{
pw.println("<br><br><br>SORRY INVALID ID
TRY AGAIN ID<br><br>");
pw.println("<a href=\"order.html\">press HERE to
RETRY</a>");
}
}
pw.println("</body></html>");
con.close();
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Web.xml:

```
<?xml version="1.0"?>
<web-app >
<servlet>
<servlet-name>login</servlet-name>
<servlet-class>login</servlet-class>
</servlet>
<servlet>
<servlet-name>reg</servlet-name>
<servlet-class>reg</servlet-class>
</servlet>
<servlet>
<servlet-name>profile</servlet-name>
<servlet-class>profile</servlet-class>
</servlet>
<servlet>
<servlet-name>order</servlet-name>
<servlet-class>order</servlet-class>
</servlet>
<servlet>
<servlet-name>catalog</servlet-name>
<servlet-class>catalog</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>login</servlet-name>
<url-pattern>/login</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>reg</servlet-name>
<url-pattern>/reg</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>profile</servlet-name>
<url-pattern>/profile</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>order</servlet-name>
<url-pattern>/order</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>catalog</servlet-name>
<url-pattern>/catalog</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>main.html</welcome-file></welcome-file-list>
</web-app>
```

9. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

PROCEDURE:
1) Create your own directory under tomcat/webapps (e.g. tr1)
2) Copy the html files in tr1
3) Copy the jsp files also into tr1
4) Start tomcat give the following command
Catalina.bat run
At install-dir/bin
5) at I.E give url as http://localhost:8081/tr1/main.html

Home page:
Main.html:

```
<html>
<body>
<br><br><br><br><br><br>
<h1 align="center"><u>ONLINE BOOK STORAGE</u></h1><br><br><br>
<h2 align="center"><PRE>
<b> Welcome to online book storage.
Press LOGIN if you are having id
Otherwise press REGISTRATION
</b></PRE></h2>
<br><br><pre>
<div align="center"><a href="login.html">LOGIN</a>
<a href="reg.html">REGISTRATION</a></div></pre>
</body></html>
```

Login page:
Login.html:

```
<html>
<body><br /><br /><br />
<form name="myform" method="post" action="login.jsp">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br /> PASSWORD :<input type="password"
name="pwd"
/></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Registration page:

Reg.html:

```
<html>
<body><br /><br />
<form name="myform" method="post" action="reg.jsp">
<table align="center" >
<tr>
<td>NAME</td>
<td>:<input type="text" name="name" /></td>
</tr>
<tr>
<td>ADDRESS</td>
<td>:<input type="text" name="addr" /></td>
</tr>
<tr>
<td>CONTACT NUMBER</td>
<td>:<input type="text" name="phno" /></td>
</tr>
<tr>
<td>LOGINID</td>
<td>:<input type="text" name="id" /></td>
</tr>
<tr>
<td>PASSWORD</td>
<td>:<input type="password" name="pwd" /></td>
</tr>
</table>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
```

```
</div>
</form>
</body>
</html>
```

User profile page:

Profile.html:

```
<html>
<body><br /><br /><br />
<form name="myform" method="post" action="profile.jsp">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
      <input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```

Books catalog page:

Catalog.html:

```
<html>
<body><br /><br /><br />
<form method="post" action="catalog.jsp">
<div align="center"><pre>
BOOK TITLE :<input type="text" name="title" /><br />
</pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"
name="button1"/>      
<input type="reset" value="clear" name="button2"/>
</div>
</form>
</body> </html>
```

Shopping cart, Payment by credit card, Order Conformation page:

Order.html:

```
<html>
<body><br /><br />
<form method="post" action="order.jsp">
<div align="center"><pre>
ID
:<input type="text" name="id" /><br />
```

```
PASSWORD
:<input type="password" name="pwd" /><br/> TITLE
:<input type="text" name="title" /><br /> NO. OF BOOKS :<input type="text" name="no"
/><br />
DATE
:<input type="text" name="date" /><br />
CREDIT CARD NUMBER :<input type="password" name="cno" /><br
/></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" name="button1"/>
      <input type="reset" value="clear"
name="button2"/>
</div>
</form>
</body>
</html>
```

Login JSP:

Login.jsp:

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<% out.println("<html><body>");
String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger"); Statement
stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
out.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
out.println("<a href=\"login.html\">press LOGIN to
RETRY</a>");
}
else
```

```
{
out.println("<br><br>VALID LOGIN ID<br><br>"); out.println("<h3><ul>");
out.println("<li><a
href=\"profile.html\"><fontcolor=\"black\">USER
PROFILE</font>
</a></li><br><br>");
out.println("<li><a
href=\"catalog.html\"><fontcolor=\"black\">BOOKS
CATALOG</font></a></li><br><br>");
out.println("<li><a
href=\"order.html\"><fontcolor=\"black\">ORDER
CONFIRMATION</font>
</a></li></ul><br><br>");
}
out.println("</body></html>");
con.close();
%>
```

Registration JSP:

Reg.jsp :

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<% response.setContentType("text/html");
out.println("<html><body>");
String name=request.getParameter("name");
String addr=request.getParameter("addr");
String phno=request.getParameter("phno");
String id1=request.getParameter("id");
String pwd1=request.getParameter("pwd");
int no=Integer.parseInt(phno);
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger"); Statement
stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{ if(id1.equals(rs.getString(1))&&pwd1.equals(rs.getString(2)))
{
flag=1;
}
}
```

```
if(flag==1)
{ out.println("<br><br>SORRY INVALID ID ALREADY
EXITS TRY AGAIN WITH NEW ID<br><br>");
out.println("<a href=\"reg.html\">press REGISTER to
RETRY</a>");
}
else
{ Statement stmt1=con.createStatement();
stmt1.executeUpdate("insert into login
values('"+name+"','"+addr+"','"+no+"','"+id1+"','"+pwd1+"');"); out.println("<br><br>YOUR
DETAILS
ARE
ENTERED<br><br>");
out.println("<a href=\"login.html\">press LOGIN to
login</a>");
}
out.println("</body></html>");
con.close();
%>
```

Profile JSP:

Profile.jsp:

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<% out.println("<html><body>");
String id=request.getParameter("id");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger"); Statement
stmt=con.createStatement();
String sqlstmt="select * from login where id="+id+""; ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
out.println("<br><br><br>");
while(rs.next())
{
out.println("<div align=\"center\">");
out.println("NAME :"+rs.getString(1)+"<br>");
out.println("ADDRESS
:"+rs.getString(2)+"<br>");
out.println("PHONE NO
:"+rs.getString(3)+"<br>");
out.println("</div>");
flag=1;
}
if(flag==0)
{
out.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
```

```
out.println("<a href=\"profile.html\">press HERE to
RETRY</a>");
}
out.println("</body></html>");
con.close(); %>
```

Catalog JSP:

Catalog.jsp:

```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<% out.println("<html><body>");
String title=request.getParameter("title");
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger"); Statement
stmt=con.createStatement();
String sqlstmt="select * from book where title=\'"+title+"\'"; ResultSet
rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
out.println("<div align=\"center\">");
out.println("TITLE
:"+rs.getString(1)+"<br>");
out.println("AUTHOR :"+rs.getString(2)+"<br>"); out.println("VERSION
:"+rs.getString(3)+"<br>");
out.println("PUBLISHER :"+rs.getString(4)+"<br>"); out.println("COST
:"+rs.getString(5)+"<br>");
out.println("</div>");
flag=1;
}
if(flag==0)
{
out.println("<br><br>SORRY INVALID TITLE TRY
AGAIN <br><br>");
out.println("<a href=\"catalog.html\">press HERE to
RETRY</a>");
}
out.println("</body></html>");
con.close();
%>
```

Order servlet:

Order.java:

```
<%@page import="java.sql.*"%>
```

```
<%@page import="java.io.*"%>
<% int count;
out.println("<html><body>");
String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
String title=request.getParameter("title");
String count1=request.getParameter("no");
String date=request.getParameter("date");
String cno=request.getParameter("cno");
count=Integer.parseInt(count1);
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection
con=DriverManager.getConnection("jdbc:odbc:orcl","scott","tiger"); Statement
stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0,amount,x;
while(rs.next())
{
if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
{
flag=1;
}
}
if(flag==0)
{
out.println("<br><br>SORRY INVALID ID TRY AGAIN
ID<br><br>");
out.println("<a href= \"order.html \" >press HERE to RETRY</a>");
}
else
{
Statement stmt2=con.createStatement();
String s="select cost from book where title=\'"+title+"\'"; ResultSet rs1=stmt2.executeQuery(s);
int flag1=0;
while(rs1.next())
{
flag1=1;
x=Integer.parseInt(rs1.getString(1));
amount=count*x;
out.println("<br><br>AMOUNT
:"+amount+"<br><br><br><br>");
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insert into details
values('"+id+"','"+title+"',"+amount+",'"+cno+"');"); out.println("<br>YOUR ORDER has
taken<br>");
}
```

```
if(flag1==0)
{
out.println("<br><br><br>SORRY INVALID ID
TRY AGAIN ID<br><br>");
out.println("<a href=\"order.html\">press HERE to
RETRY</a>");
}
}
out.println("</body></html>");
con.close();
%>
```

Output:

ADDITIONAL PROGRAMS LIST

TO CREATE TIMETABLE:

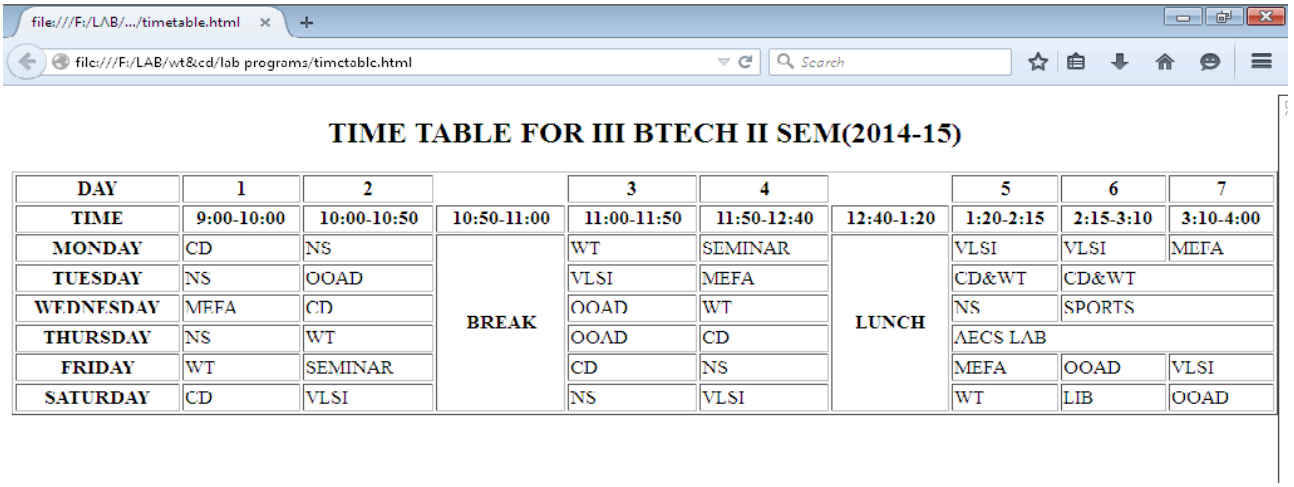
```
<html>
```

```
<body>
  <table>
    <table border=1 width="100%"align="left">
      <caption align="top"><h2>TIME TABLE FOR III BTECH II
          SEM(2014-15)</h2></caption>
        <tr>
      <th>DAY</th>
          <th>1</th>
          <th>2</th>
          <th></th>
          <th>3</th>
          <th>4</th>
          <th></th>
          <th>5</th>
          <th>6</th>
          <th>7</th>
    </tr>
    <tr>
          <th>TIME</th>
            <th>9:00-10:00</th>
            <th>10:00-10:50</th>
            <th>10:50-11:00</th>
            <th>11:00-11:50</th>
           <th>11:50-12:40</th>
          <th>12:40-1:20</th>
          <th>1:20-2:15</th>
```

```
            <th>2:15-3:10</th>
              <th>3:10-4:00</th>
        </tr>
        <tr>
            <th>MONDAY</th>
            <td>CD</td>
            <td>NS</td>
            <th rowspan=6>BREAK</th>
            <td>WT</td>
            <td>SEMINAR</td>
            <th rowspan=6>LUNCH</th>
            <td>VLSI</td>
            <td>VLSI</td>
            <td>MEFA</td>
        </tr>
          <tr>
            <th>TUESDAY</th>
            <td>NS</td>
            <td>OOAD</td>
            <td>VLSI</td>
            <td>MEFA</td>
            <td>CD&WT</td>
            <td colspan=2>CD&WT</td>
        </tr>
            <tr>
```

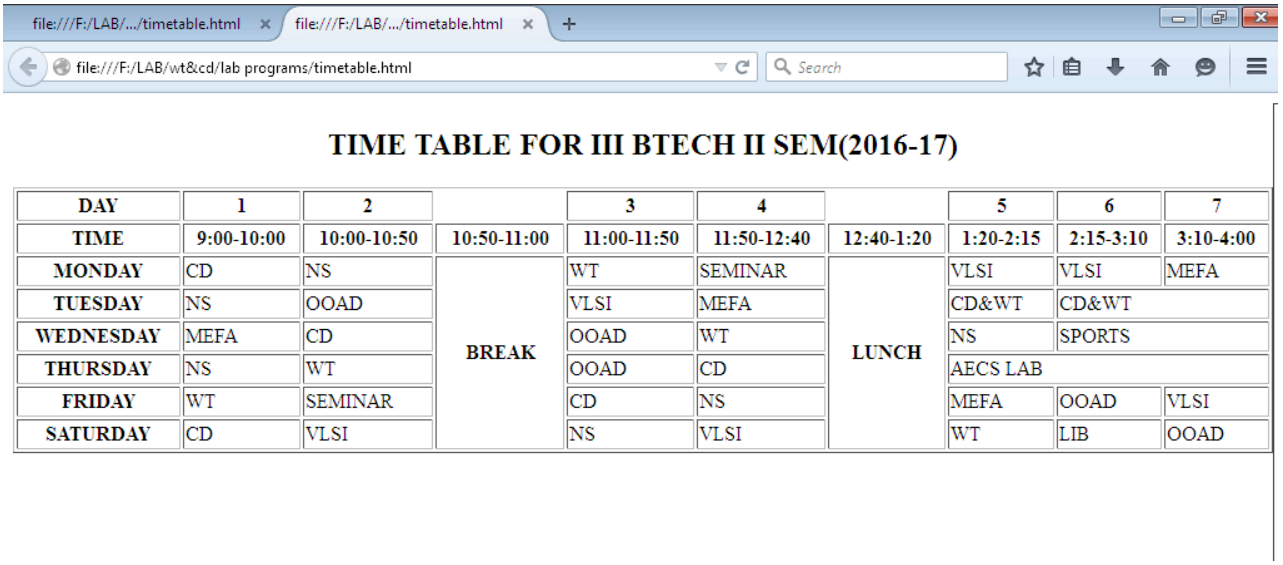
```
<th>WEDNESDAY</th>
<td>MEFA</td>
<td>CD</td>
<td>OOAD</td>
<td>WT</td>
<td>NS</td>
<td colspan=2>SPORTS</td>
</tr>
<tr>
<th>THURSDAY</th>
<td>NS</td>
<td>WT</td>
<td>OOAD</td>
<td>CD</td>
<td colspan=3>AECS LAB</td>
</tr>
<tr>
<th>FRIDAY</th>
<td>WT</td>
<td>SEMINAR</td>
<td>CD</td>
<td>NS</td>
<td>MEFA</td>
<td>OOAD</td>
<td>VLSI</td>
```

```
                </tr>
                 <tr>
                   <th>SATURDAY</th>
                   <td>CD</td>
                   <td>VLSI</td>
                   <td>NS</td>
                   <td>VLSI</td>
                   <td>WT</td>
                   <td>LIB</td>
                   <td>OOAD</td>
                 </tr>
        <img style="position:absolute;z-index:-1"src="C:\Users\Public\Pictures\Sample Pictures\tulips.jpg" height="500" width="1500">
    </body>
</html>
```



TIME TABLE FOR III BTECH II SEM(2014-15)

DAY	1	2		3	4		5	6	7
TIME	9:00-10:00	10:00-10:50	10:50-11:00	11:00-11:50	11:50-12:40	12:40-1:20	1:20-2:15	2:15-3:10	3:10-4:00
MONDAY	CD	NS	BREAK	WT	SEMINAR	LUNCH	VLSI	VLSI	MEFA
TUESDAY	NS	OOAD		VLSI	MEFA		CD&WT	CD&WT	
WEDNESDAY	MEFA	CD		OOAD	WT		NS	SPORTS	
THURSDAY	NS	WT		OOAD	CD		AECS LAB		
FRIDAY	WT	SEMINAR		CD	NS		MEFA	OOAD	VLSI
SATURDAY	CD	VLSI		NS	VLSI		WT	LIB	OOAD



TIME TABLE FOR III BTECH II SEM(2016-17)

DAY	1	2		3	4		5	6	7
TIME	9:00-10:00	10:00-10:50	10:50-11:00	11:00-11:50	11:50-12:40	12:40-1:20	1:20-2:15	2:15-3:10	3:10-4:00
MONDAY	CD	NS	BREAK	WT	SEMINAR	LUNCH	VLSI	VLSI	MEFA
TUESDAY	NS	OOAD		VLSI	MEFA		CD&WT	CD&WT	
WEDNESDAY	MEFA	CD		OOAD	WT		NS	SPORTS	
THURSDAY	NS	WT		OOAD	CD		AECS LAB		
FRIDAY	WT	SEMINAR		CD	NS		MEFA	OOAD	VLSI
SATURDAY	CD	VLSI		NS	VLSI		WT	LIB	OOAD

TO CREATE REGISTRATION FORM:

```
<html>

<head>

<title>

Registration Form

</title>

</head>

<body bgcolor="pink">

<form name="regform">


<h2>Registration Form</h2>

<h4>Student Name: <input type="text" name="sname" size="20"></h4>

<h4>Father Name      : <input type="text" name="fname"  size="20"></h4>

<h4>Mother Name     : <input type="text" name="mname" size="20"></h4>

<h4>Date Of Birth: <input type="Date" name="bday" size="20"></h4>
```

```
<h4>Address          : <Textarea cols="20" rows="5" name="taname" value="" size="20"></textarea></h4>

<h4>E-Mail           : <input type="email" name="email" value="" size="20"></h4>

<h4>Phone No         : <input type="text" name="phno" value="" size="20"></h4>

<h4>Select Gender:</h4>

        <input type="radio" name="gender" value="Male">Male

        <input type="radio" name="gender" value="FeMale">FeMale

        <input type="radio" name="gender" value="Others">Others

<h4>Educational Qualification: </h4>

<h4><input type="checkbox"  name="option1" value="SSC" size="20">SSC

        <input type="checkbox"  name="option2" value="Inter" size="20">Inter

        <input type="checkbox"  name="option3" value="Degree" size="20">Degree

        <input type="checkbox"  name="option4" value="Others" size="20">Others

<h4>Nationality:</h4>

        <select name="My_menu" value="My_menu">

                <option value="My_menu">select</option>

                <option value="Hindu">Hindu</option>

                <option value="Christian">Christian</option>

                <option value="Muslim">Muslim</option>

                <option value="Others">Others</option>

        </select><br><br>

<input type="submit" value="Submit">

<input type="reset" value="Reset">

</form>
```

</body>

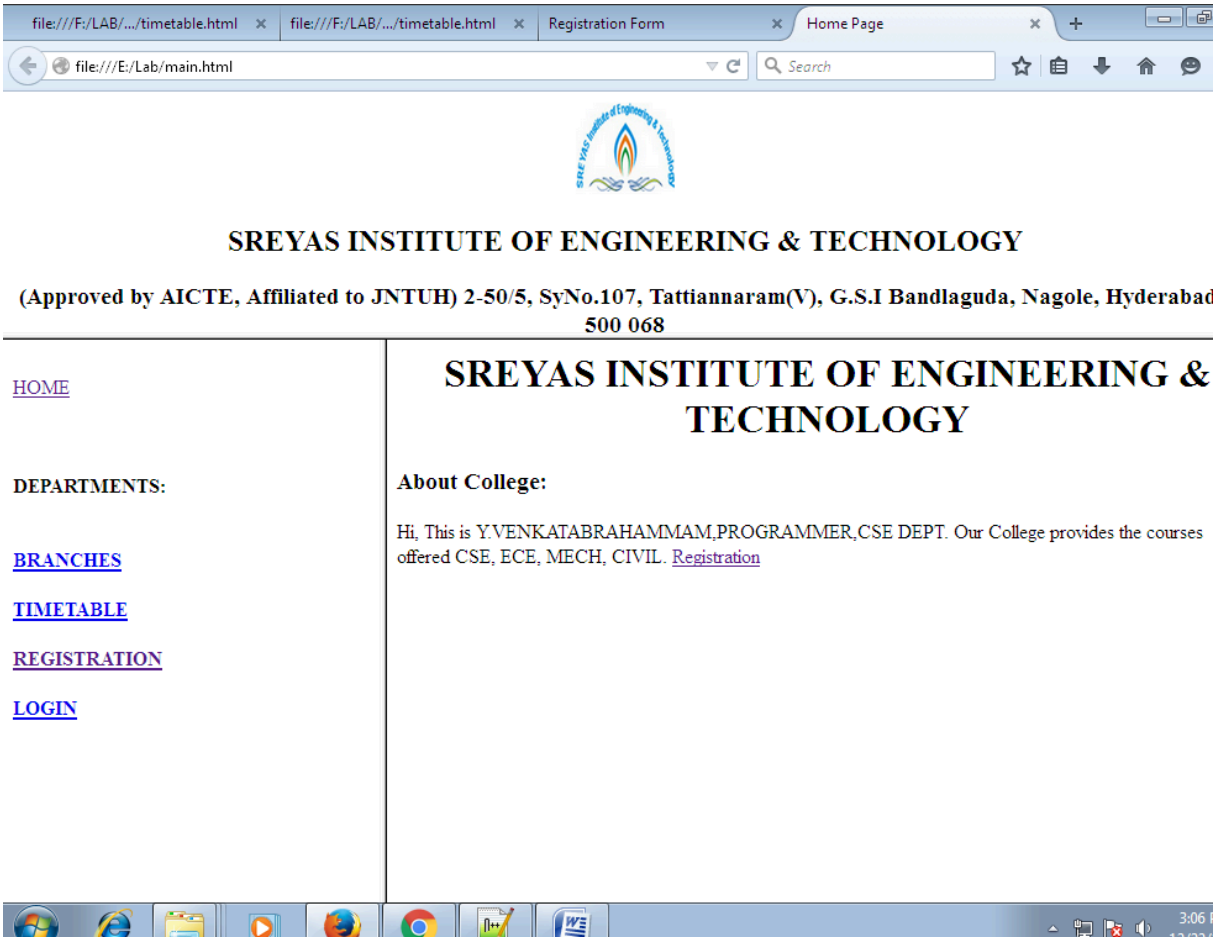
</html>

OUTPUT:

TO CREATE FRAMES:

```
<html>
<head>
<title>
Home Page
</title>
</head>
<frameset rows="30%,70%">
<frame src="title.html" name="name1">
<frameset cols="30%,70%">
<frame src="menu.html" name="name2">
<frame src="home.html" name="name3">
</frameset>
</frameset>
</html>
```

OUTPUT:



WEB TECHNOLOGIES VIVA QUESTIONS

HTML

1. **What is HTML?**

HTML (HyperText Markup Language) is a Universal language which allows an individual using special code to create web pages to be viewed on the Internet.

2. **What is a tag?**

In HTML, a tag tells the browser what to do. When you write an HTML page, you enter tags for many reasons; to change the appearance of text, to show a graphic, or to make a link to another page.

3. **What is the simplest HTML page?**

   HTML Code:

   <HTML><HEAD><TITLE>This is my page title! </TITLE></HEAD>

   <BODY> This is my message to the world! </BODY></HTML>

   Browser Display: This is my message to the world!

4. **How do I create frames? What is a frameset?**

   Frames allow an author to divide a browser window into multiple (rectangular) regions. Multiple documents can be displayed in a single window, each within its own frame. Graphical browsers allow these frames to be scrolled independently of each other, and links can update the document displayed in one frame without affecting the others.

   You can't just "add frames" to an existing document. Rather, you must create a frameset document that defines a particular combination of frames, and then display your content documents inside those frames. The frameset document should also include alternative non-framed content in a NOFRAMES element.

5. **How can I include comments in HTML?**

   An HTML comment begins with "<!—", ends with "--> ", and does not contain "--" or ">" anywhere in the comment.

6. **What is a Hypertext link?**

   A hypertext link is a special tag that links one page to another page or resource. If you click the link, the browser jumps to the link's destination.

7. **What is a DOCTYPE? Which one do I use?**

   According to HTML standards, each HTML document begins with a DOCTYPE declaration that specifies which version of HTML the document uses. Many browsers use

the document's DOCTYPE declaration to determine whether to use a stricter, more standards-oriented layout mode, or to use a "quirks" layout mode that attempts to emulate older, buggy browsers.

8. **Can I nest tables within tables?**

   Yes, a table can be embedded inside a cell in another table. The main caveat about nested tables is that older versions of Netscape Navigator have problems with them if you don't explicitly close you TR, TD, and TH elements. To avoid problems, include closing tags for you TR, TD, an TH even though the HTML specifications don't require them.

9. **How do you align a table to the right (or left)?**

   You use the <table align="right"> property to float a table to the right. Put left in place of right to float right.

10. **How can I use tables to structure forms?**

    Small forms are sometimes placed within a TD element within a table. This can be useful for positioning a form relative to other content, but it doesn't help position the form-related elements relative to each other. The table must be within the form and then use the table to position the form elements.

11. **How do I open a link into a new window?**

    Add target="_blank" to your link syntax.

JavaScript:

1. **What's relationship between JavaScript and ECMAScript?**

   ECMAScript is yet another name for JavaScript (other names include LiveScript). The current JavaScript that you see supported in browsers is ECMAScript revision

2. **What are JavaScript types?**

   Number, String, Boolean, Function, Object, Null, Undefined.

3. **How do you convert numbers between different bases in JavaScript?**

   Use the parseInt() function, that takes a string as the first parameter, and the base as a second parameter. So to convert hexadecimal 3F to decimal, use parseInt ("3F", 16).

4. **What does isNaN function do?**

   Return true if the argument is not a number.

5. **What is negative infinity?**

   It's a number in JavaScript, derived by dividing negative number by zero.

6. **What boolean operators does JavaScript support?**

   &&, || and !

7. **What does "1"+2+4 evaluate to?**

   Since 1 is a string, everything is a string, so the result is 124.

8. **How about 2+5+"8"?**

   Since 2 and 5 are integers, this is number arithmetic, since 8 is a string, it's concatenation, so 78 is the result.

9. **What looping structures are there in JavaScript?**

   for, while, do-while loops, but no foreach.

10. **How do you create a new object in JavaScript?**
    var obj = new Object(); or var obj = {};

11. **How do you assign object properties?**

    obj["age"] = 17 or obj.age = 17.

12. **What's a way to append a value to an array?**

    arr[arr.length] = value;

13. **What is this keyword?**

It refers to the current object.

14. JavaScript gives HTML designers a programming tool

HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages

15. JavaScript can react to events

A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.

16. JavaScript can read and write HTML elements

A JavaScript can read and change the content of an HTML element

17. JavaScript can be used to validate data

A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing

18. JavaScript can be used to detect the visitor's browser

A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser.

19. JavaScript can be used to create cookies - A JavaScript can be used to store and retrieve information on the visitor's computer.

CSS Interview Questions

1. **What are Cascading Style Sheets?**

Called (CSS) is a list of statements (or rules) that can assign various rendering properties to HTML elements. Style rules can be specified for a single element occurrence, multiple elements, and entire document, or even multiple documents at once.

2. **What is class?**

A group of instances of the same element to which a unique style can be attached.

3. **What is grouping?**

Gathering into a comma separated list two or more selectors that share the same style or into a semicolon separated list two or more declarations that are attached to the same selector.

4. **What is ID selector?**

ID selector is an individually identified (named) selector to which a specific style is declared. Using the ID attribute, the declared style can then be associated with one and only one HTML element per document as to differentiate it from all the other elements. They use the # character followed by a name.

5. **What is contextual selector?**

A contextual selector addresses a specific occurrence of an element. It is a string of individual selectors separated by white space, a search pattern, where only the last element in the pattern is addressed providing it matches the specified context. Example: " td li {color: red} "

6. **What does \ABCD (and \ABCDE) mean?**

CSS allows Unicode characters to be entered by number. For example, if a class value in some Russian document contains Cyrillic letters EL PE (Unicode numbers 041B and 041F) and you want to write a style rule for that class, you can put that letter into the style sheet by writing:

.\041B\041F {font-style: italic;}

This works on all keyboards, so you don't need a Cyrillic keyboard to write class names in Russian or another language that uses that script.

7. **What are the advantages / disadvantages of various style methods?**

External Style Sheet Advantages

- Can control styles for multiple documents at once.
- Classes can be created for use on multiple HTML element types in many documents.
- Selector and grouping methods can be used to apply styles under complex contexts.

External Style Sheet Disadvantages

- An extra download is required to import style information for each document.
- The rendering of the document may be delayed until the external style sheet is loaded.

Embedded Style Sheet Advantages

- Classes can be created for use on multiple tag types in the document.
- Selector and grouping methods can be used to apply styles under complex contexts
- No additional downloads necessary to receive style information.

Inline Style Disadvantages

- Does not distance style information from content.
- Cannot control styles for multiple documents at once.
- Author cannot create or control classes of elements to control multiple element types within the document.
- Selector grouping methods cannot be used to create complex element addressing scenarios.

8. **What is property?**

Property is a stylistic parameter (attribute) that can be influenced through CSS, e.g. font or width. There must always be a corresponding value or values set to each property.

9. **What is the CSS clear property?**

The clear property specifies which sides of an element where other floating elements are not allowed.

- This method cannot control styles for multiple documents at once.

Inline Style Advantages

- Useful for small quantities of style definitions.
- Can override other style specification methods at the local level so only exceptions need to be listed in conjunction with other style methods.

10. **What are the necessities of using HTML forms?**
    1. Gathering user information
    2. Conducting Surveys
    3. Interactive services

11. **What are the sequences of steps for each HTTP request from a client to the server?**
    1. Making the connection
    2. Making a request
    3. The response
    4. Closing the connection

What is XML?

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML is a W3C Recommendation

The Difference Between XML and HTML

XML is not a replacement for HTML.

XML and HTML were designed with different goals:

- XML was designed to transport and store data, with focus on what data is
- HTML was designed to display data, with focus on how data looks
- HTML is about displaying information, while XML is about carrying information.

Some advantages of XML are:

- It is a platform independent language.
- It is as easy as HTML.
- XML is fully compatible with applications like JAVA, and it can be combined with any application which is capable of processing XML irrespective of the platform it is being used on.
- XML is an extremely portable language to the extent that it can be used on large networks with multiple platforms like the internet, and it can be used on handhelds or palmtops or PDAs.
- XML is an extendable language, meaning that you can create your own tags, or use the tags which have already been created.
- It can be deployed on any network if it is amicable for usage with the application in use.
- If the application can work along with XML, then XML can work on any platform and has no boundaries.
- It is also vendor independent and system independent. While data is being exchanged using XML, there will be no loss of data even between systems that use totally different formats.

Servlets:

Q: ***Explain the life cycle methods of a Servlet.***

A: *The javax.servlet.Servlet interface defines the three methods known as life-cycle method.*
public void init(ServletConfig config) throws ServletException
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException

public void destroy()

*First the servlet is constructed, then initialized wih the **init()** method.*
*Any request from client are handled initially by the **service()** method before delegating to the **doXxx()** methods in the case of HttpServlet.*

*The servlet is removed from service, destroyed with the **destroy()** methid, then garbaged collected and finalized.*

Q: What is the difference between the getRequestDispatcher(String path) method of javax.servlet.ServletRequest interface and javax.servlet.ServletContext interface?

A: The getRequestDispatcher(String path) method of javax.servlet.ServletRequest interface accepts parameter the path to the resource to be included or forwarded to, which can be relative to the request of the calling servlet. If the path begins with a "/" it is interpreted as relative to the current context root.

The getRequestDispatcher(String path) method of javax.servlet.ServletContext interface cannot accepts relative paths. All path must sart with a "/" and are interpreted as relative to curent context root.

Q: Explain the directory structure of a web application.

A: The directory structure of a web application consists of two parts.
A private directory called WEB-INF
A public resource directory which contains public resource folder.

WEB-INF folder consists of
1. web.xml

2. classes directory
3. lib directory

Q: What are the common mechanisms used for session tracking?

A: Cookies
SSL sessions
URL- rewriting

Q: Explain ServletContext.

A: ServletContext interface is a window for a servlet to view it's environment. A servlet can use this interface to get information such as initialization parameters for the web applicationor servlet container's version. Every web application has one and only one ServletContext and is accessible to all active resource of that application.

Q: What is preinitialization of a servlet?

A: A container doesnot initialize the servlets ass soon as it starts up, it initializes a servlet when it receives a request for that servlet first time. This is called lazy loading. The servlet specification defines the <load-on-startup> element, which can be specified in the deployment descriptor to make the servlet container load and initialize the servlet as soon as it starts up. The process of loading a servlet before any request comes in is called preloading or preinitializing a servlet.

Q: What is the difference between Difference between doGet() and doPost()?

A: A doGet() method is limited with 2k of data to be sent, and doPost() method doesn't have this limitation. A request string for doGet() looks like the following:
http://www.allapplabs.com/svt1?p1=v1&p2=v2&...&pN=vN
doPost() method call doesn't need a long text tail after a servlet name in a request. All

parameters are stored in a request itself, not in a request string, and it's impossible to guess the data transmitted to a servlet only looking at a request string.

Q: What is the difference between HttpServlet and GenericServlet?

A: A GenericServlet has a service() method aimed to handle requests. HttpServlet extends GenericServlet and adds support for doGet(), doPost(), doHead() methods (HTTP 1.0) plus doPut(), doOptions(), doDelete(), doTrace() methods (HTTP 1.1).
Both these classes are abstract.

Q: What is the difference between ServletContext and ServletConfig?

A: ServletContext: Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.The ServletContext object is contained within the ServletConfig object, which the Web server provides the servlet when the servlet is initialized

ServletConfig: The object created after a servlet is instantiated and its default constructor is read. It is created to pass initialization information to the servlet.

1. **What is the difference between JSP and Servlets ?**

JSP is used mainly for presentation only. A JSP can only be HttpServlet that means the only supported protocol in JSP is HTTP. But a servlet can support any protocol like HTTP, FTP, SMTP etc.

2. **What is difference between custom JSP tags and beans?**

- Custom JSP tag is a tag you defined. You define how a tag, its attributes and its body are interpreted, and then group your tags into collections called tag libraries that can be used in any number of JSP files. To use custom JSP tags, you need to define three separate components: the tag handler class that defines the tag's behavior ,the tag

library descriptor file that maps the XML element names to the tag implementations and the JSP file that uses the tag library

- JavaBeans are Java utility classes you defined. Beans have a standard format for Java classes. You use tags
- Custom tags and beans accomplish the same goals — encapsulating complex behavior into simple and accessible forms. There are several differences:
- Custom tags can manipulate JSP content; beans cannot. Complex operations can be reduced to a significantly simpler form with custom tags than with beans. Custom tags require quite a bit more work to set up than do beans. Custom tags usually define relatively self-contained behavior, whereas beans are often defined in one servlet and used in a different servlet or JSP page. Custom tags are available only in JSP 1.1 and later, but beans can be used in all JSP 1.x versions.

3. **What are the different ways for session tracking?**

Cookies, URL rewriting, HttpSession, Hidden form fields

4. **What mechanisms are used by a Servlet Container to maintain session information?**

Cookies, URL rewriting, and HTTPS protocol information are used to maintain session information

5. **Difference between GET and POST**

- In GET your entire form submission can be encapsulated in one URL, like a hyperlink. query length is limited to 255 characters, not secure, faster, quick and easy. The data is submitted as part of URL.
- In POST data is submitted inside body of the HTTP request. The data is not visible on the URL and it is more secure.

6. **What is session?**

The session is an object used by a servlet to track a user's interaction with a Web application across multiple HTTP requests. The session is stored on the server.

7. **What is servlet mapping?**

The servlet mapping defines an association between a URL pattern and a servlet. The mapping is used to map requests to Servlets.

8. **What is servlet context ?**

The servlet context is an object that contains a information about the Web application and container. Using the context, a servlet can log events, obtain URL references to resources, and set and store attributes that other servlets in the context can use.

9. **What is a servlet ?**

servlet is a java program that runs inside a web container.

10. **Can we use the constructor, instead of init(), to initialize servlet?**

Yes. But you will not get the servlet specific things from constructor. The original reason for init() was that ancient versions of Java couldn't dynamically invoke constructors with arguments, so there was no way to give the constructor a ServletConfig. That no longer applies, but servlet containers still will only call your no-arg constructor. So you won't have access to a ServletConfig or ServletContext.

2. **How many JSP scripting elements are there and what are they?**

There are three scripting language elements: declarations, scriptlets, expressions.

3. **How do I include static files within a JSP page?**

Static resources should always be included using the JSP include directive. This way, the inclusion is performed just once during the translation phase

4. **How can I implement a thread-safe JSP page?**

You can make your JSPs thread-safe adding the directive <%@ page isThreadSafe="false" % > within your JSP page.

5. **What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher()?**

In request.getRequestDispatcher(path) in order to create it we need to give the relative path of the resource. But in resourcecontext.getRequestDispatcher(path) in order to create it we need to give the absolute path of the resource.

6. **What are the lifecycle of JSP?**

When presented with JSP page the JSP engine does the following 7 phases.

- Page translation: -page is parsed, and a java file which is a servlet is created.
- Page compilation: page is compiled into a class file
- Page loading : This class file is loaded.
- Create an instance :- Instance of servlet is created
- jspInit() method is called
- _jspService is called to handle service calls
- _jspDestroy is called to destroy it when the servlet is not required.

7. **What are context initialization parameters?**

Context initialization parameters are specified by the in the web.xml file, these are initialization parameter for the whole application.

8. **What is a Expression?**

Expressions are act as place holders for language expression, expression is evaluated each time the page is accessed. This will be included in the service method of the generated servlet.

9. **What is a Declaration?**

It declares one or more variables or methods for use later in the JSP source file. A declaration must contain at least one complete declarative statement. You can declare any number of variables or methods within one declaration tag, as long as semicolons separate them. The declaration must be valid in the scripting language used in the JSP file. This will be included in the declaration section of the generated servlet.

10. **What is a Scriptlet?**

A scriptlet can contain any number of language statements, variable or expressions that are valid in the page scripting language. Within scriptlet tags, you can declare variables to use later in the file, write expressions valid in the page scripting language, use any of the JSP implicit objects or any object declared with a . Generally a scriptlet can contain any java code that are valid inside a normal java method. This will become the part of generated servlet's service method.

1. **What is a servlet?**

Servlets are modules that extend request/response-oriented servers,such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database. Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface.

2. **Whats the advantages using servlets over using CGI?**

Servlets provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.

3. **What are the general advantages and selling points of Servlets?**

A servlet can handle multiple requests concurrently, and synchronize requests. This allows servlets to support systems such as online
real-time conferencing. Servlets can forward requests to other servers and servlets. Thus servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organizational boundaries.

4. **Which package provides interfaces and classes for writing servlets?**

javax

5. **What's the Servlet Interface?**

The central abstraction in the Servlet API is the Servlet interface. All servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.Servlets > Generic Servlet > HttpServlet > MyServlet.
The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

6. **When a servlet accepts a call from a client, it receives two objects. What are they?**

ServletRequest (which encapsulates the communication from the client to the server) and ServletResponse (which encapsulates the communication from the servlet back to the client). ServletRequest and ServletResponse are interfaces defined inside javax.servlet package.

7. **What information does ServletRequest allow access to?**

Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names
of the remote host that made the request and the server that received it. Also the input stream, as ServletInputStream.Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and GET methods.

8. **What type of constraints can ServletResponse interface set on the client?**

It can set the content length and MIME type of the reply. It also provides an output stream, ServletOutputStream and a Writer through
which the servlet can send the reply data.

1. **Explain servlet lifecycle?**

Each servlet has the same life cycle: first, the server loads and initializes the servlet (init()), then the servlet handles zero or more client requests (service()), after that the server removes the servlet (destroy()). Worth noting that the last step on some servers is done when they shut down.

2. **How does HTTP Servlet handle client requests?**

An HTTP Servlet handles client requests through its service method. The service method supports standard HTTP client requests by dispatching each request to a method designed to handle that request.

JDBC interview questions and answers

1. **What are the steps involved in establishing a JDBC connection?**

This action involves two steps: loading the JDBC driver and making the connection.

2. **How can you load the drivers?**

Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Your driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverXYZ, you would load the driver with the following line of code:

Class.forName("jdbc.DriverXYZ");

3. **What will Class.forName do while loading drivers?**

It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

4. **How can you make the connection?**

To establish a connection you need to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

String url = "jdbc:odbc:Fred";
Connection con = DriverManager.getConnection(url, "Fernanda", "J8?);

5. **How can you create JDBC statements and what are they?**

A Statement object is what sends your SQL statement to the DBMS. You simply create a Statement object and then execute it, supplying the appropriate execute method with the SQL statement you want to send. For a SELECT statement, the method to use is executeQuery. For statements that create or modify tables, the method to use is executeUpdate. It takes an instance of an active connection to create a Statement object. In the following example, we use our Connection object con to create the Statement object

Statement stmt = con.createStatement();

6. **How can you retrieve data from the ResultSet?**

JDBC returns results in a ResultSet object, so we need to declare an instance of the class ResultSet to hold our results. The following code demonstrates declaring the ResultSet object rs.

ResultSet rs = stmt.executeQuery("SELECT COF_NAME, PRICE FROM COFFEES");
String s = rs.getString("COF_NAME");

The method getString is invoked on the ResultSet object rs, so getString() will retrieve (get) the value stored in the column COF_NAME in the current row of rs.

7. **What are the different types of Statements?**

Regular statement (use createStatement method), prepared statement (use prepareStatement method) and callable statement (use prepareCall)

8. **How can you use PreparedStatement?**

This special type of statement is derived from class Statement.If you need a Statement object to execute many times, it will normally make sense to use a PreparedStatement object instead. The advantage to this is that in most cases, this SQL statement will be sent to the DBMS right away, where it will be compiled. As a result, the PreparedStatement object contains not just an SQL statement, but an SQL statement that has been precompiled. This

means that when the PreparedStatement is executed, the DBMS can just run the PreparedStatement's SQL statement without having to compile it first.

9. PreparedStatement updateSales = con.prepareStatement("UPDATE COFFEES SET SALES = ? WHERE COF_NAME LIKE ?");

10. **What does setAutoCommit do?**

When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed. The way to allow two or more statements to be grouped into a transaction is to disable auto-commit mode:

11. **How do you call a stored procedure from JDBC?**

The first step is to create a CallableStatement object. As with Statement an and PreparedStatement objects, this is done with an open
Connection object. A CallableStatement object contains a call to a stored procedure.

12. CallableStatement cs = con.prepareCall("{call SHOW_SUPPLIERS}");

13. ResultSet rs = cs.executeQuery();

JSP interview questions and answers

1. **What is JSP? Describe its concept.**

JSP is a technology that combines HTML/XML markup languages and elements of Java programming Language to return dynamic content to the Web client, It is normally used to handle Presentation logic of a web application, although it may have business logic.

2. **What are the lifecycle phases of a JSP?**

JSP page looks like a HTML page but is a servlet. When presented with JSP page the JSP engine does the following 7 phases.

A. Page translation: -page is parsed, and a java file which is a servlet is created.

B. Page compilation: page is compiled into a class file

C. Page loading : This class file is loaded.

D. Create an instance :- Instance of servlet is created

E. jspInit() method is called

F. _jspService is called to handle service calls

G. _jspDestroy is called to destroy it when the servlet is not required.

3. **What is a translation unit?**

JSP page can include the contents of other HTML pages or other JSP files. This is done by using the include directive. When the JSP engine is presented with such a JSP page it is converted to one servlet class and this is called a translation unit, Things to remember in a translation unit is that page directives affect the whole unit, one variable declaration cannot occur in the same unit more than once, the standard action jsp:useBean cannot declare the same bean twice in one unit.

3. **How is JSP used in the MVC model?**

JSP is usually used for presentation in the MVC pattern (Model View Controller) i.e. it plays the role of the view. The controller deals with calling the model and the business classes which in turn get the data, this data is then presented to the JSP for rendering on to the client.

4. **What are context initialization parameters?**

Context initialization parameters are specified by the <context-param> in the web.xml file, these are initialization parameter for the whole application and not specific to any servlet or JSP.

5. **What is a output comment?**

A comment that is sent to the client in the viewable page source. The JSP engine handles an output comment as un-interpreted HTML text, returning the comment in the HTML output

sent to the client. You can see the comment by viewing the page source from your Web browser.

6. **What is a Hidden Comment?**

   A comment that documents the JSP page but is not sent to the client. The JSP engine ignores a hidden comment, and does not process any code within hidden comment tags. A hidden comment is not sent to the client, either in the displayed JSP page or the HTML page source. The hidden comment is useful when you want to hide or "comment out" part of your JSP page.

7. **What is a Expression?**

   Expressions are act as place holders for language expression, expression is evaluated each time the page is accessed.

8. **What is a Declaration?**

   It declares one or more variables or methods for use later in the JSP source file. A declaration must contain at least one complete declarative statement. You can declare any number of variables or methods within one declaration tag, as long as semicolons separate them. The declaration must be valid in the scripting language used in the JSP file.

9. **What is a Scriptlet?**

   A scriptlet can contain any number of language statements, variable or method declarations, or expressions that are valid in the page scripting language. Within scriptlet tags, you can declare variables or methods to use later in the file, write expressions valid in the page scripting language, use any of the JSP implicit objects or any object declared with a <jsp:useBean>.

10. **What are the implicit objects? List them.**

    Certain objects that are available for the use in JSP documents without being declared first. These objects are parsed by the JSP engine and inserted into the generated servlet. The implicit objects are:

    - request
    - response
    - pageContext
    - session
    - application

- out
- config
- page
- exception

11. **What's the difference between forward and sendRedirect?**

When you invoke a forward request, the request is sent to another resource on the server, without the client being informed that a different resource is going to process the request. This process occurs completely with in the web container And then returns to the calling method. When a sendRedirect method is invoked, it causes the web container to return to the browser indicating that a new URL should be requested. Because the browser issues a completely new request any object that are stored as request attributes before the redirect occurs will be lost. This extra round trip a redirect is slower than forward.

12. **What are the different scope values for the <jsp:useBean>?**

The different scope values for <jsp:useBean> are:

- page
- request
- session
- application

13. **Why are JSP pages the preferred API for creating a web-based client program?**

Because no plug-ins or security policy files are needed on the client systems(applet does). Also, JSP pages enable cleaner and more module application design because they provide a way to separate applications programming from web page design. This means personnel involved in web page design do not need to understand Java programming language syntax to do their jobs.

14. **Is JSP technology extensible?**

Yes, it is. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

15. **What is difference between custom JSP tags and beans?**

Custom JSP tag is a tag you defined. You define how a tag, its attributes and its body are interpreted, and then group your tags into collections called tag libraries that can be used in

any number of JSP files. Custom tags and beans accomplish the same goals — encapsulating complex behavior into simple and accessible forms. There are several differences:

- Custom tags can manipulate JSP content; beans cannot.
- Complex operations can be reduced to a significantly simpler form with custom tags than with beans.
- Custom tags require quite a bit more work to set up than do beans.
- Custom tags usually define relatively self-contained behavior, whereas beans are often defined in one servlet and used in a different servlet or JSP page.
- Custom tags are available only in JSP 1.1 and later, but beans can be used in all JSP 1.x versions.

Classes and Interfaces of Servlet:

The javax.servlet package defines the contract between container and the servlet class. It provides the flexibility to container vendor to implement the API the way they want so long as they follow the specification. To the developers, it provides the library to develop the servlet based applications.

Interfaces of javax.Servlet

Filter	A filter is an object that performs filtering tasks on either the request to a resource (a servlet or static content), or on the response from a resource, or both.
FilterChain	A FilterChain is an object provided by the servlet container to the developer giving a view into the invocation chain of a filtered request for a resource.
FilterConfig	A filter configuration object used by a servlet container to pass information to a filter during initialization.
RequestDispatcher	Defines an object that receives requests from the client and sends them to any resource (such as a servlet, HTML file, or JSP file) on the server.

Servlet	Defines methods that all servlets must implement.
ServletConfig	A servlet configuration object used by a servlet container to pass information to a servlet during initialization.
ServletContext	Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.
ServletContextAttributeListener	Implementations of this interface receive notifications of changes to the attribute list on the servlet context of a web application.
ServletContextListener	Implementations of this interface receive notifications about changes to the servlet context of the web application they are part of.
ServletRequest	Defines an object to provide client request information to a servlet.