Table of Content

1. **Introduction**
   - 1.1. **Project Description:** **1**
   - 1.2. **Technology Used** **2**
     - 1.2.1. **J2EE 1.4.1** **2**
     - 1.2.2. **JDBC** **4**
     - 1.2.3. **Hyper Text Markup Language** **8**
     - 1.2.4. **Java Script** **9**
     - 1.2.5. **ORACLE** **10**

2. **Project life cycle**
   - 2.1. **Software Development life cycle** **17**

3. **Requirement Analysis**
   - 3.1. **Overview** **21**
   - 3.2. **Objective of requirement analysis** **22**
   - 3.3. **Software requirement specification** **23**
     - 3.3.1. **Introduction** **23**
     - 3.3.2. **Product description** **24**
     - 3.3.3. **Usage scenario** **25**
     - 3.3.4. **Data objects and description** **26**
     - 3.3.5. **Design constraints** **27**

4. **System Designing**
   - 4.1. **Overview** **28**
     - 4.1.1. **Database Design** **28**
     - 4.1.2. **Human – Machine Interface Design** **28**
   - 4.2. **Software Design specification** **30**
     - 4.2.1. **Database Design** **30**
   - 4.3. **Process Model** **38**
     - 4.3.1. **Functional Decomposition Diagram** **38**
     - 4.3.2. **Data Flow Diagram** **39**

5. **Coding** **46**

6. **Snapshots**
   - 6.1. **Visitor interfaces** **110**
   - 6.2. **Administrator interfaces** **122**

7. **Testing and implementation**
   - 7.1. **Testing** **130**

7.1.1. **Testing Objectives** **130**
7.1.2. **Testing Principles** **130**
7.1.3. **Unit Testing** **131**
7.1.4. **Integration Testing** **134**
7.1.5. **System Testing** **136**
7.1.6. **Optimization Points** **138**

8. **Limitations and Future Developments** **139**
8.1. **Conclusion** **139**

9. **Bibliography**
9.1. **Books & Authors** **142**
9.2. **Consulted Website** **142**

a.

1 *Introduction*

1.1. ***Project Description***

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that online shopping system project report the process of shopping on the web is becoming commonplace.

The objective of this online shopping system project project is to develop a general purpose e-commerce store where any product (such as books, CDs, computers, mobile phones, electronic items, and home appliances) can be bought from the comfort of home through the Internet.

However, for implementation purposes, this online shopping system project paper will deal with an online book store.
An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that online shopping system project report time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

1.2. Technologies Used

1.2.1. ***Java Server Pages (JSP)***

With the advent of Internet, the monolithic application architecture changed to the multi-tiered client/server architecture. The need for server-side scripting gradually began to dominate aspects of Web Programming. Microsoft introduced Active Server Pages (ASP) to capture the market demand for server-side scripting. Working on similar lines, Sun Microsystems released Java Server Pages (JSP) to add server side programming functionalities to Java.

A typical web application consists of the presentation logic representing the static content used to design the structure of a web page in terms of the page layout, color, and text. The business logic or the dynamic content involves application of business intelligence and diagnostics in terms of financial and business calculations. When developing web applications, time is often lost in situations where the developer is required to code for the static content.

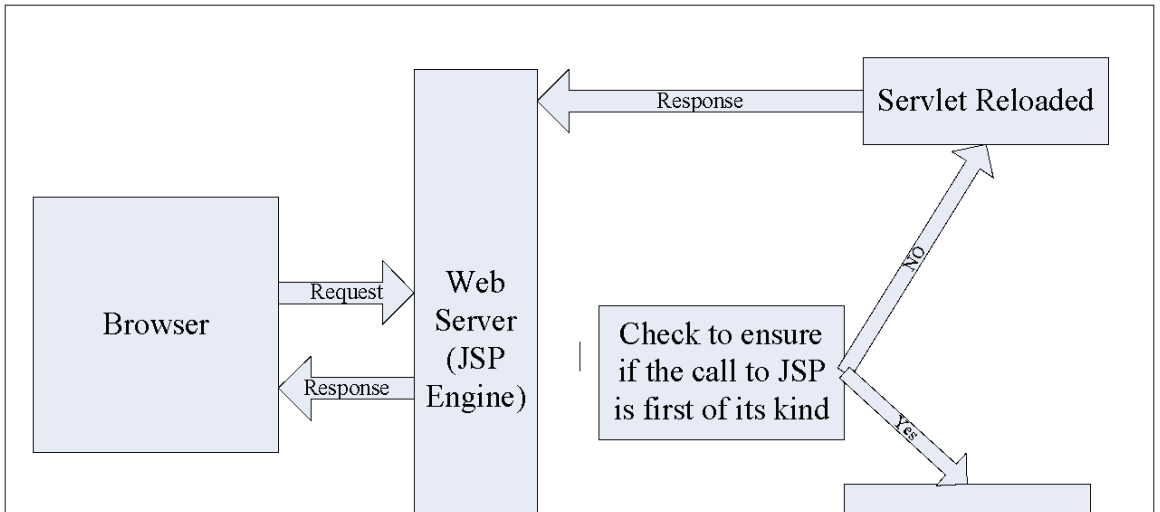
JSP Technology has facilitated the segregation of work profiles of a web designer and a web developer. A Web Designer can design and formulate the layout for the web page by using HTML. On the other hand, a Web Developer working independently can use Java code and other JSP specific tags to code for the business logic. The simultaneous construction of static and dynamic content facilitates development of quality applications with increased productivity.

The JSP Request-Response Cycle:

JSP files are stored on the web server with an extension of .jsp. When the client/browser requests for a particular JSP page, the server in turn sends a request to the JSP Engine.

The following figure represents the process of the flow of events that online shopping system project report occur after a client requests for a JSP page.

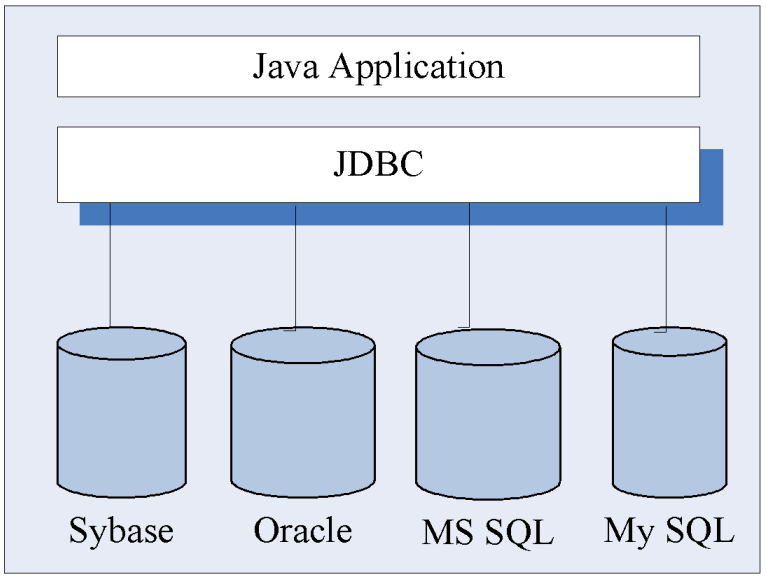
The Request-Response Cycle for a JSP Page



The request-response cycle essentially comprises of two phases, namely the translation phase and request-processing phase. The translation phase is implemented by the JSP engine and involves generation of a servlet. Internally, these results in the creation of a class file for the JSP page that online shopping system project report implements the servlet interface. During the request-processing phase, the response is generated according to the request specification. The servlet then sends back a response corresponding to the request received. After the servlet is loaded for the first time, it remains active and possesses all the subsequent requests with responses, saving time that online shopping system project report would otherwise be lost in reloading a servlet at each request.

1.2.2 ***JDBC***

JDBC (Java Database Connectivity) is an application program interface (API) specification for connecting programs written in Java to the data in popular database. It is provided by Sun Microsystems, the application program interface lets you encode access request statements in structured query language (SQL) that online shopping system project report are then passed to the program that online shopping system project report manages the database. It returns the results through a similar interface.
From the user's point of view, Java application looks something like this online shopping system project:



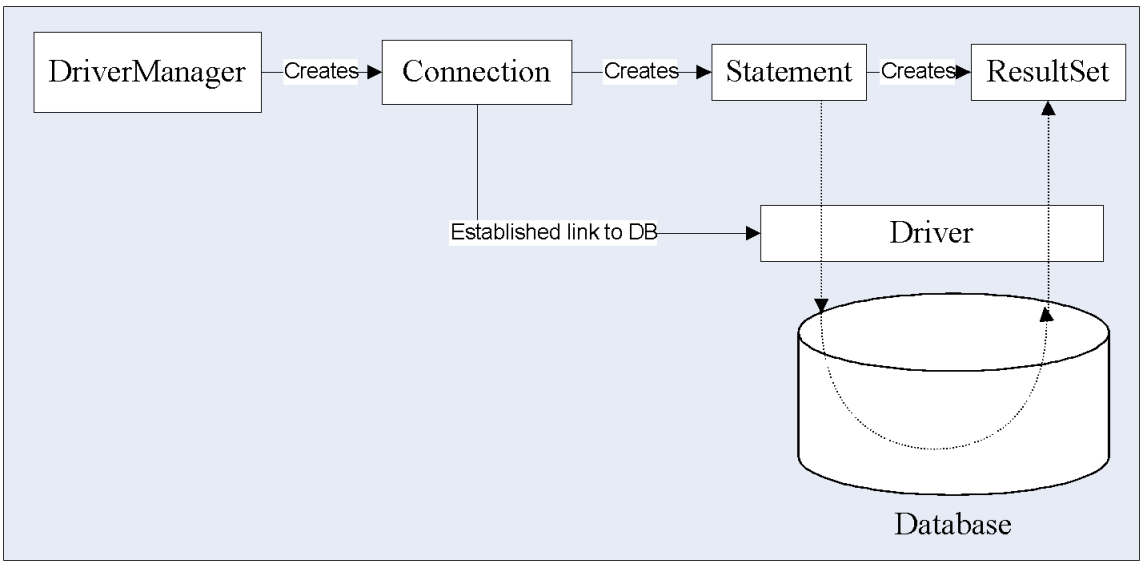
Features of JDBC API:

Java Database Connectivity (JDBC) provides a database programming API for Java programs. Some of the features of JDBC API are as follows:

- Contains a set of classes and interfaces that online shopping system project report are used to connect to a database built using any DBMS/RDBMS, submit SQL queries to a database, and retrieve and process the results of SQL queries.
- Is a low-level interface in which SQL *select* and *update* statements are called directly from within Java programs.
- Can be used with both two-tier and three-tier database architectures. In two-tier architecture, a Java program invokes the methods of JDBC API, which in turn communicates with the database server. In three-tier architecture, a Java applet or an HTML form submits SQL queries to a middle-tier server. Middle-tier server in turn uses JDBC API to communicate with the database server.

JDBC Architecture:

The JDBC architecture is based on a collection of Java interfaces and classes that online shopping system project report together enables you to connect to data sourced, to create and execute SQL statements, and to retrieve and modify sata in a database. These operations are illustrated in the figure below:



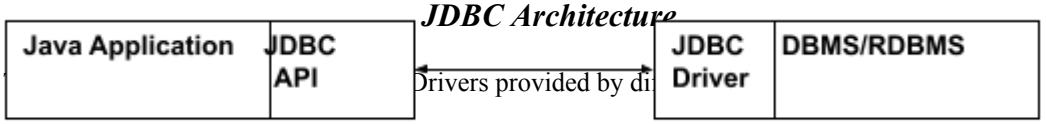
Each of the boxes in the illustration represents a JDBC class or interface that online shopping system project report has fundamental role in accessing a relational database.

JDBC Drivers:

JDBC API takes care of converting Java commands to generic SQL statements. However, to address specific database issues, each database vendor provides a driver along with the database. Java Applications invoke the methods JDBC API. JDBC API in turn uses a driver to communicate with a specific database.

JDBC API submits queries to the JDBC driver. The JDBC driver converts queries to a form that online shopping system project report a particular DBMS/RDBMS can understand. The JDBC driver also retrieves the results of SQL queries, converts it into equivalent JDBC API classes and objects that online shopping system project report can be used by the application. Since the JDBC Driver only takes care of the interactions with database, any change made to the database does not affect the application.

JDBC Architecture



- **JDBC-ODBC Bridge Driver:** The first category of JDBC drivers provides a bridge between the JDBC API and the ODBC API. There are several DBMS/RDBMS, such as MS Access and SQL Server that online shopping system project report contain the ODBC Driver embedded into them. Since the ODBC API is written in the C language and makes use of pointers and other constructs that online shopping system project report Java does not support, a Java program cannot directly communicate with an ODBC Driver. The bridge translates the standard JDBC calls to corresponding ODBC calls, and sends them to ODBC data source via ODBC libraries.

- **Native API Partly Java Driver:** These drivers use a mixture of Java implementation and vendor specific native APIs to provide data access. JDBC database calls are translated into vendor specific API calls. The database will process the request and sends the result back through the API, which will in turn forward them back to the JDBC driver. The JDBC driver will translate the result to the JDBC standard and return them to the Java application. There is one layer fewer to go through than for a type 1 driver and so in general a type 2 driver will be faster than a type 1 driver.
  Some DBMS/RDBMS such as DB2 and Informix contain a JDBC driver supplied by the database vendor.

- **Intermediate Database Access Server:** Type 3 drivers use an intermediate database server that online shopping system project report has the ability to connect multiple Java clients to multiple database servers. Clients connect to database server via an intermediate server component that online shopping system project report acts as a gateway for multiple database servers. The java client application sends a JDBC call through a JDBC driver to the intermediate data access server,

which completes the request to the data sourcing using another driver (for example, a type 2 driver).

BEA WebLogic includes a type 3 driver. One of the benefit of using a type 3 driver is that online shopping system project report it allows flexibility on the architecture of the application, as the intermediate server can can abstract details of connection to database servers.

- **Native Protocol Pure Java Driver/JDBC-Net Pure Java Driver:** These drivers convert the JDBC API calls to direct network calls using vendor specific networking protocol. They do this online shopping system project by making direct socket connections with the database. Type 4 drivers offer better performance than others.

1.2.3. Hyper Text Markup Language (HTML)

Hypertext Markup Language (HTML) is a language for describing how pages of text, graphics, and other Information are organized, formatted, and linked together. It is not really a programming language in the sense of COBOL or Visual Basic, but it does provide powerful capabilities for text formatting and output display. The original purpose of HTML was primarily as a tool for making text documents readily available on the Internet. Basic graphic support was added, and the things started to take off. Now we have sound, live video, retail catalogs, and much, much more available to us.

HTML pages are the standard interface to the Internet. This online shopping system project basic language provides the necessary nuts and bolts for building Web pages.

HTML stands for Hyper Text Markup Language and was developed specifically for use on the system of links, which can be non sequential in order. HTML is not a programming language. It is a marking up language used for linking one piece of information to another.

Using a markup language means that online shopping system project report tags can be added to the words in the document and web enable them. A tag is a set of descriptive formatting codes used in HTML document that online shopping system project report instructs a web browser how to display text and graphics on a web page.

HTML 2.0 is the first standard of HTML specification, released in 1996. HTML 3.6 was the next specification to be developed. It was a joint effort of many software companies like Microsoft, IBM, Sun Microsystems, Novell and others.

HTML 4.0 is the latest version of HTML, which includes CSS and dynamic HTML features.

1.2.4. JavaScript

The project uses JavaScript as the client side scripting language for JSP/HTML pages in the project.
JavaScript is an easy to use object-scripting language designed for creating live online applications that online shopping system project report link together resources on both clients and servers. JavaScript is designed for use by HTML page authors and enterprise application developers to dynamically script the behavior of objects running on either the client or the servers. JavaScript ‘s design and concept represent the next generation of software for the Internet and is:

- Designed for creating network centric applications
- Complementary to and integrated with Java
- Complementary to and integrated with HTML
- Open and cross platform

JavaScript is a platform-independent, event driven, interpreted programming language developed by Netscape Communications Corp. and Sun Microsystems. Originally called Livescript, JavaScript is a programming language that online shopping system project report can be included on web pages to make them more interactive.

JavaScript is easier to understand, less complex version of its distant cousin, Java. It is a text-based language that online shopping system project report must be placed within HTML that online shopping system project report must be placed within, HTML, to be read by the browser and interpreted so the instructions can be performed.

JavaScript is a preferred language for client side scripting. This online shopping system project is mainly because when client side scripting is done browser compatibility is an issue of concern and both major browsers support JavaScript.

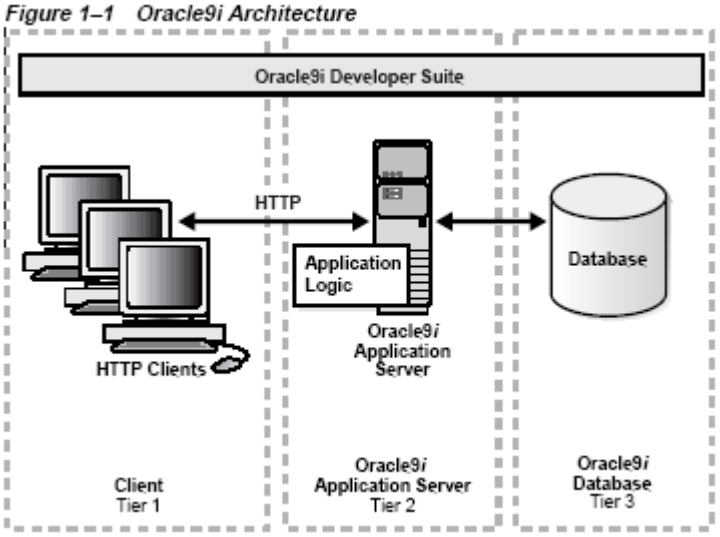
1.2.6 Oracle :

Oracle takes business where it needs to be: meeting and exceeding stringent demands for high-quality service in a service-driven marketplace. Oracle is designed to optimize traditional, internet and intranet applications, and to stimulate the emerging hosted application market on the internet.
Oracle builds on historic strengths to offer the first complete *and* simple software infrastructure for the internet's next generation of intelligent, collaborative applications. The Oracle new features expedite delivery of critical performance, scalability, and availability essential to providing hosted service software for anyone, anywhere, anytime. Oracle architecture is shown in figure:

Oracle components include the following:

- Oracle Database

Figure 1–1 Oracle9i Architecture



- Oracle Application Server
- Oracle Developer Suite

Oracle Database:

The Oracle Database introduces the following advanced and automated design features that online shopping system project report refine Oracle Application Server and Oracle Developer Suite to optimize performance for traditional applications and the emerging hosted application market.

- **Oracle Real Application Clusters:**

The next evolutionary step after Oracle Parallel Server, Oracle Real Application Clusters provides out-of-the-box, linear scaling transparency, compatibility with all applications without redesign, and the ability to rapidly add nodes and disks.

- **Systems Management**

Integrated system management products create a complete view of all critical components that online shopping system project report drive e-business processes. From the client and application server to the database and host, Oracle quickly and completely assesses the overall health of an e-business infrastructure.

- **High Availability**

Setting a new standard for high availability, Oracle introduces powerful new functionality in areas of disaster recovery, system fault recovery, and planned downtime.

- **High Security**

Oracle offers the most secure internet platform for protecting company information through multiple layers of security for data, users, and companies. Included are features for building internet-scale applications, for providing security for users, and for keeping data from different hosted user communities separate.

Oracle Application Server:

Recognized as the leading application server for database-driven Web sites, Oracle Application Server offers the industry's most innovative and comprehensive set of middle-tier services.

- **Comprehensive Middle-tier Services**

Continued innovation within comprehensive middle-tier services, ranging from self service enterprise portals, to e-stores and supplier exchange, sustains the Oracle Application Server as the industry's preferred application server for database-driven Web sites.

- **New Caching Technology**

The new caching technology in Oracle can dramatically increase Web site performance, scalability, and availability. Greater numbers of users can be provided with more personalized, dynamic Web content without adding more application or database servers.

- **Scalability and Performance**

Superb scalability and performance now is made available for all Web applications. Oracle Portal services make it easy for Web site developers to deploy enterprise portals with centralized management and unified security. Standard Java, with rich XML and content management support, as well as back-office transactional applications built using Oracle Forms Developer, can easily be deployed.

- **Wireless Device Access**

Information in any database or internet application is easily available through Oracle. Support for each wireless device's specific markup language is no longer necessary.

- **Business Intelligence**

Oracle Application Server has built-in reporting and ad hoc query functionality to derive business intelligence after Web site deployment.

Oracle Developer Suite:

Oracle Internet Developer Suite combines leading Oracle applications development tools, business intelligence tools, and enterprise portal building tools in a single, integrated product. Built on internet standards such as Java, XML, CORBA, and HTML, the Oracle Internet Developer Suite provides a high-performance development environment with tools needed to respond to rapidly changing markets and user demands.

- **Extensive Development Alternatives**

Oracle Internet Developer Suite tools are suitable for any kind of development approach, including component-based development, Java coding, and visual modeling, and offer rapid application development based on 4GL. Applications can be developed for all clients, including High Productive Java Client, Universal HTML Client, and Anywhere Mobile Client.

- **Tools**

Oracle Internet Developer Suite tools include: Oracle Forms Developer; Oracle Designer; Oracle JDeveloper and Business Components for Java; Oracle Reports Developer; and Oracle Discoverer.

- **Other Features**

All applications are single, middle-tier deployable with an Oracle Application Server. Oracle Portal is available for an integrated, personalized portal view with all applications. Integrated business intelligence for enterprise reporting and ad-hoc query is available.

Oracle Database New Features:

Important new features are introduced with the release of the Oracle Database, including advancements in Real Application Clusters, systems management, availability, scalability, and security.

1. **Availability:**

Oracle dramatically extends Oracle's leadership in Internet database availability, which is critical for any e-business application. Key areas in Oracle include:

- Enhanced Disaster Recovery Environment
- Online Data Evolution
- Precision Database Repair

2. **Scalability:**

Oracle allows e-business to scale the most demanding e-business applications with intensive transactions loads. Key areas include:

- Increased Transaction Throughput on Oracle Real Application Clusters:
- Scalable Session State Management:
- Fine-Grained, Automatic Resource Management:

3. **Performance:**

Oracle is the continued performance leader in all relevant areas. Measurement and tuning projects continue to improve Oracle runtime performance. Major focus performance improvements in Oracle are in areas critical for e-business.

- Native Compilation and Improved PL/SQL Optimization:
- Latch Contention Improvements:
- Enhanced Java Performance:
- Network and Distributed Database Performance:

4. **Security:**

Oracle continues to provide the most secure application development and deployment platform in the industry. Key areas include:

- Strong Three-Tier Security:
- Security for Hosting Environments:
- Standards-Based Public Key Infrastructure (PKI):
- Improved User and Security Policy Management:
- Data Encryption and Label Security

SQL PLUS:

SQLPLUS is usually thought of as a kind of interactive report writer. It uses SQL to get information from the Oracle database, and lets you create reports by giving you easy control over titles, column headings, subtotals and totals, reformatting of numbers and text, and much more. It also can be used to change the database by using insert, update, and delete commands in SQL. SQLPLUS can even be used as a code generator, where a series of commands in SQLPLUS can dynamically build a program and then execute it.

In most production applications, more advanced report writers are used—such as Web-based parameter-driven reports. SQLPLUS is most commonly used for simple queries and printed reports. Getting SQLPLUS to format information in reports according to your taste and needs requires only a handful of commands, or keywords that online shopping system project report instruct SQLPLUS how to behave.

You can write SQLPLUS reports while working interactively with SQLPLUS—that online shopping system project report is, you can type commands about page headings, column titles, formatting, breaks, totals, and so on, and then execute a SQL query, and SQLPLUS will immediately produce the report formatted to your specifications. For quick answers to simple questions that online shopping system project report aren't likely to recur, this online shopping system project is a fine approach. More common, however, are complex reports that online shopping system project report need to be produced periodically, and that online shopping system project report you'll want to print rather than just view on the screen. Unfortunately, when you quit SQLPLUS, it promptly forgets every instruction you've given it. If you were restricted to using

SQLPLUS only in this online shopping system project interactive way, then running the same report at a later time would require typing everything all over again.

The alternative is very straightforward. You simply type the commands, line by line, into a file. SQLPLUS can then read this online shopping system project file as if it were a script, and execute your commands just as if you were typing them. In effect, you create a report program, but you do it without a programmer or a compiler. You create this online shopping system project file using any of the popular editor programs available or even (given certain restrictions) a word processor.

The editor is not a part of Oracle. Editors come in hundreds of varieties, and every company or person seems to have a favorite. Oracle realized this online shopping system project, and decided to let you choose which editor program to use, rather than packaging a program with Oracle and forcing you to use it. When you're ready to use your editor program, you suspend SQLPLUS, jump over to the editor program, create or change your SQLPLUS report program (also called a start file), and then jump back to SQLPLUS right at the spot you left and run that online shopping system project report report (see Figure 6-1).

SQLPLUS also has a built-in editor of its own, sometimes called the command line editor, which allows you to quickly modify a SQL query without leaving SQLPLUS.

Difference between SQL and SQL Plus:

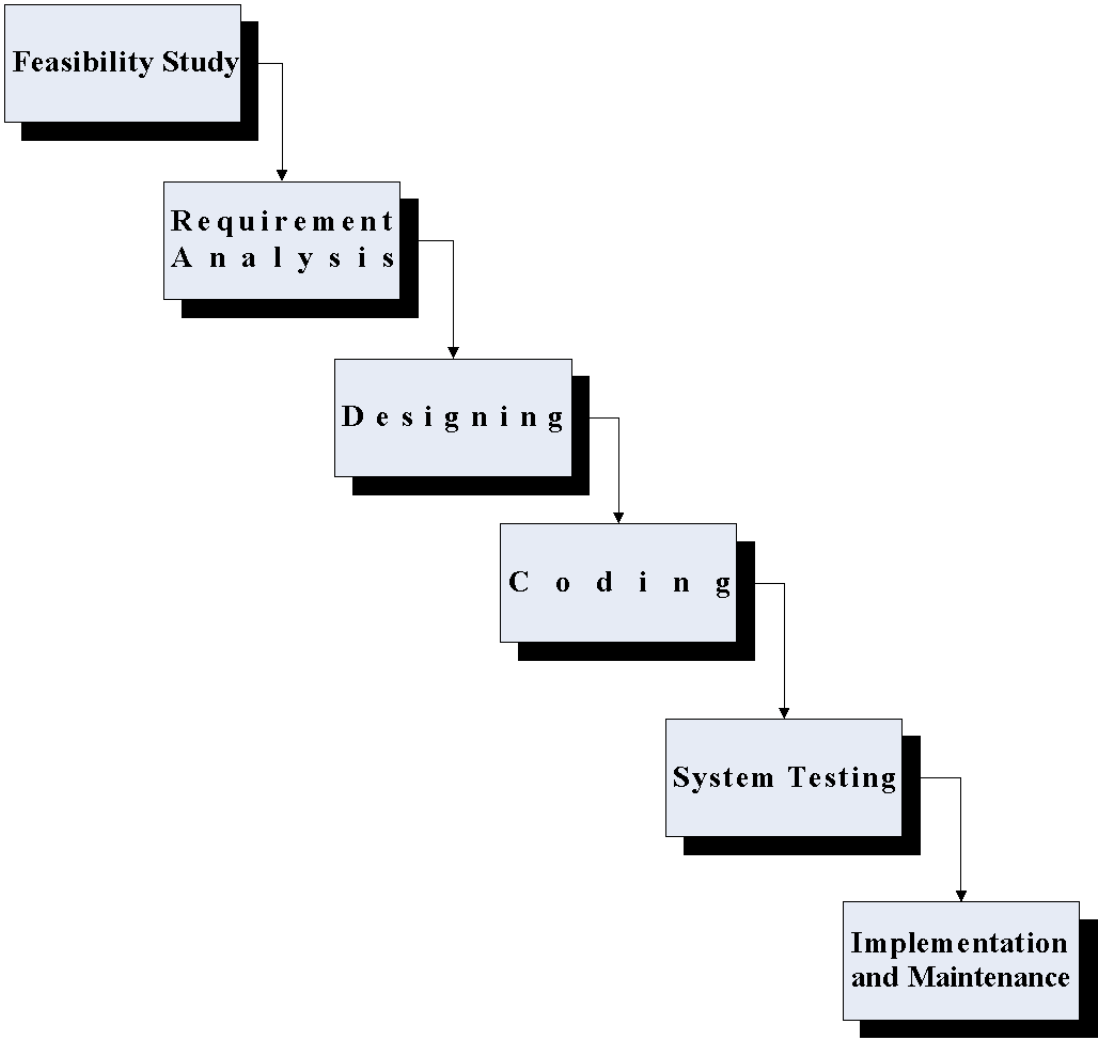
SQL	***SQL PLUS***
SQL is a language for communicating with the oracle server to access data.	SQL *PLUS recognizes SQL statements and sends them to the server.
SQL is based on American National Standards Institute (ANSI) standard SQL.	SQL *PLUS is the Oracle proprietry interface for executing SQl statements.

SQL manipulates data and table definitions in the database.	SQL *PLUS does not allow manipulation of values in the database.
SQL does not have a continuation character.	SQL *PLUS uses a dash(-) as a continuation character if the command is longer than one line.
It can not be abbreviated.	It can be abbreviated.
SQL uses a termination character to execute commands immediately.	SQL *PLUS does not require termination character; executes commands immediately.
SQL uses functions to perform some formatting.	SQL *PLUS uses commands to format data.

2 Project Life Cycle

2.1 Software Development Life Cycle:

Software development organization follows some process when developing a software product. A key component of any software development process is the life cycle model on which the process is based. The particular life cycle model can significantly affect overall life cycle costs associated with a software product. Life cycle of the software starts from concept exploration and at the retirement of the software.



Software Development Life Cycle

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE:

The system development life cycle is classically thought of as the set of activities that online shopping system project report analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation.
- Requirement Analysis.
- System Designing.
- Coding.
- System Testing.
- Implementation and Maintenance.

Preliminary Investigation: -

An important outcome of the preliminary investigation is the determination that online shopping system project report system is feasible or not. In the conduct of feasibility study, there are three major distinct and interrelated areas were taken into consideration. They are as follows:

1. Technical Feasibility :

The System of operation which was functioning earlier was totally manual, with no kind of automation or computerization. All the departments were maintaining separate registers for keeping various records. Due to expansion of schools more workspace and it appears a tedious task to maintain with specifying equipment and software that online shopping system project report will successfully support the tasks required. As a result the computerized system is technically feasible as it is efficient, less time consuming, can produce outputs faster, can input large amount of data in limited time scale and easier to use in operation

2. Operational Feasibility : -

The ultimate users i.e. the people who are supposed to use the system are trained for a period of one month so as to get familiar with the new system and its operation. They are taught about the new skills and the new technology and how the technology will be useful to them in their functioning. Operational feasibility is concerned with human, organizational and political aspects. General impression of these factors is gained from the corporate appraisal.

3. Economical Feasibility :

The computerized system is economically feasible in the sense the cost of the hardware and software and the cost to training of personnel of the company to operate the system and the installation cost is less than the cost of maintaining the registers. This online shopping system project may not be a big sum in the long run of the school business. Also the time taken for the entire process of formulation, checking, studying and installation of the project has been equal to one working month of the school. As a result, there has been no hesitation on pert of the management in adopting the new system.

Requirements analysis:-

Analysis of requirements includes studying the existing system and collecting data. During analysis, data are collected on the available files, decision points and transaction handled by the present system. Once the structured analysis is completed, the analyst has affirmed understanding of what is to be done.

System Designing:

The design of an information system produces the details that online shopping system project report clearly describe how a system will meet the requirements identified during system analysis. System analysts begin the design process by identifying reports and other outputs system will produce. The system design also describes the data to be input, calculated or stored.

Coding: -

This online shopping system project is the phase in which computer based system is constructed from the specifications prepared in the design phase. Equipment is acquired and installed during the development phase. All necessary procedure, manuals software specifications, and other documentation are completed. The staff is trained.

System Testing:-

During system testing, the system is used experimentally to ensure that online shopping system project report the software does not fail. In other words we can say that online shopping system project report it will run according to its specifications and in the way users expect. Special test data are input for processing, and the result examined.

Implementation, Evaluation and Maintenance:-

Implementation is the process of having systems personnel check out and put new equipments into use, train users, install the new application and construct any files of data needed to use it.

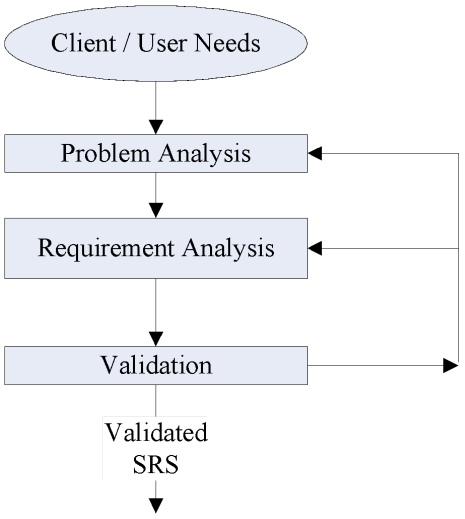
Evaluation of the system is performed to identify its strength and weaknesses. Maintenance is necessary to eliminate errors in the working system during its working

life and to tune the system to any variations in its working environment. The importance of maintenance is to continue to bring the new system to standards.

Requirement Analysis

3.1 Overview:

Analysis is a Fact Finding Technique where studies like User's need, System Requirement Specifications, Feasibility Analysis and Cost-Benefit Analysis are carried out.



This online shopping system project is the most important step in a software project where we get a general idea about the needs of the customers or end users by having man to man conversation with them and about the various conditions and restrictions that online shopping system project report have to be taken care of while developing the software application.

The purpose of this online shopping system project phase is to identify, analyze and document the exact requirements for the system. The developer, customer, a marketing organization, or any combination of the three may perform such study. It is extremely important that online shopping system project report the developers of the system study the existing system thoroughly otherwise it is impossible to satisfy the needs of the user. The requirements at this online shopping system project stage are in end-user terms.

During the Requirement Analysis Phase, the development team analyzes the requirements to be fulfilled by the Online Shoping Cart website and identifies the probable approach for meeting these requirements. To identify the requirements needed by the website, we decided to study the existing Shopping Cart process like Searching Book, Adding to Cart, money transfer. In this online shopping system project phase we

have also collect necessary information regarding the details to be stored Registered customer.

3.2 Objective of Requirement Analysis:

Requirement analysis was conducted with the following objectives in mind:

Identification of need

Information Gathering

Evaluate the system concept of feasibility

1. Identification of Need:

The success of the system depends largely on how accurately a problem is defined, thoroughly investigated, and properly carried out through the choice of solution. Users need identification and analysis is concerned with what the user needs rather then what he/she wants. Until the problem has been identified, defined, and evaluated the analyst shouldn't think about solutions and whether the problem is worth solving or not.

2. Information Gathering:

A key part of system development is gathering information. The analyst must know what information to get, where to find it, how to collect it, and how to make use of it. The proper use of tools for gathering information is the key to successful analysis.

3. Feasibility Study:

Feasibility study is carried out to test if the proposed system is feasible in terms of economy, technology, resource availability etc. As such, given unlimited resources and infinite time, all projects are feasible. Unfortunately, such results and time are not possible in real life situations. Hence it is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time in order to avoid unnecessary wastage of time, effort and professional embarrassment over an ill conceived system.

3.3 Software Requirements Specification (SRS):

3.3.1 Introduction:

A. *Purpose:*

This online shopping system project document completely describes *what* the "Shopping Cart" should do without describing *how* the software will do it. The basic goal of the requirement phase is to produce the SRS, which describing the complete external behavior of the purposed software.

B. *Scope:*

This online shopping system project document is the only one that online shopping system project report describes the requirements of the system. It is meant for use by the developer and will be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal changes approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alteration without the permission of the client.

C. *Developer's responsibility:*

The developer is responsible for:

(a) Developing the system.

(b) Installing the software on the client's hardware.

(c) Conducting any user training that online shopping system project report might be needed for using the system.

(d) Maintaining the system for a period of one year after installation.

3.3.2 Product description:

This online shopping system project section provides an overview of the software. This online shopping system project section describes the goal and objective of the software. This online shopping system project section also briefly describes the general requirements of the software. This online shopping system project section is very important for the verification of the software after the completion whether the objective and requirements of the software will met or not.

A. *Goals and objective:*

The main purpose of "Online shopping Cart" is to provide the Shopping related services on the Internet. This online shopping system project software also helps to automate the process of ordering the books in home using internet . The goals of " Online Shopping Cart" are:

- To automate the time consuming process to go to book store and purchases books .
- To advertise the new books available in Internet.
- To manage the records of customers, Books Details, Stock Details.
- To provide a searchable database of all customers and accounts.
- To minimize the amount of paper work required in the daily services.
- To provide a secure interface for the banking transactions.
- To provide an interface so that online shopping system project report user can take advantage of anytime, anywhere Shopping.

B. ***General requirements:***

During the Requirement Analysis Phase, the development team analyzes the requirements to be fulfilled by the Online Shopping website and identifies the probable approach for meeting these requirements. To identify the requirements needed by the website, we decided to study the existing Shoping process like Searching Books , money transfer. In this online shopping system project phase we have also collect necessary information regarding the details to be stored by the database for opening an account.
Finally, it was identified that online shopping system project report the Shopping Website should:

- Enable the visitors to fill Registration form.
- Provide details of the various Books available in Stores.
- Provide the information about the rate of the available books.
- Be secure enough against the malicious security attack, identity verification of the registered user and authorization.
- Be able to handle various run time exceptions and errors.
- It should provide proper interfaces to manage and view details.
- The web pages should be user friendly and well design to attract visitors.

3.3.3 Usage Scenario:

This online shopping system project section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

A. User Profiles:

There will be three levels of users:

- Administrator level (Employee)
- User level (Account Holder)

B. Use-cases:

- **Administrator level:**

This online shopping system project level of users will be able to insert new Books, new Category and Price information,. They will also be able to generate

- **User Level:**

This online shopping system project level of users will be able to Search the books, Order the books etc. They can view their records.

3.3.4 Data Objects and description:

During the requirement analysis phase, the development team examines existing Shopping Cart . After examining all process and feasibility we decided to consider following points while designing database:

- It should store information of the Registered user details.
- It should store information of the Books details.
- It should store information of the various accounts.
- It should store information of the various category of the books.

Design constraints:

A. Hardware Requirement:

I. Server (Windows 2000 Server (NT))

Microsoft 2000 Server is based on NT Technology biased to run server side scripts in ASP technology.

Disk Space	3 GB.
Web Server	Microsoft IIS (6.0)

II. Client

Disk Space	1GB.
Processor	Pentium III
Processor Speed	1.13Ghz
Memory	256 MB

B. Software Requirement:

Operating System	:	Windows 2000 / NT / XP
Technologies	**:**	JDBC, Java Beans
Client Side Scripting Language	:	HTML and JSP and Java.
Server Side Scripting Language	**:**	JSP
Markup Language	**:**	HTML
Database Server	:	ORACLE
Web Server	:	Tomcat 4.0

4

System Designing

4.1 ***OVERVIEW:***

System design is a solution, a "HOW TO APPROACH" to the creation of a new system. This online shopping system project important phase is composed of several steps. It provides the understanding and procedural details to implement the system. Design goes through a logical and physical stage of the progress. Logical design reviews the present physical system, prepares input/output specifications, makes audit security and control specifications, detailed implementation plans, and prepares the logical design walkthrough. The physical design makes out the details of the physical system, plans the system implementation and specifies any new hardware and software products.

4.1.1 DATABASE DESIGN:

The collection of data is usually referred to as the database. The database contains the information about one particular enterprise. Database system of data involves both the definitions of structures for the storage of information, processing and mechanism for the manipulation of information. In addition, the database system provides for the safety of information stored in the database despite system crashes or attempts of unauthorized access.

4.1.2 HUMAN-MACHINE INTERFACE DESIGN:

The design of the human machine interface in one of the most important aspects of system design. A good interface design should take into account the following factors: -

User characteristics:

It includes consideration of the kinds of the users who will use the equipment, their diverse backgrounds and skills, the user expectations as well as their physical characteristics. The users who posses high degree of skill often prefer more powerful functions which usually means greater complexity, unskilled operations, on the other hand, would simple functions which are easier to learn and use.

Task Characteristics:

The nature of the users tasks differ and therefore the needs for specific-kinds of service from the system. For example unstructured tasks usually requires a more flexible mode of interaction to meet the varying needs of users as opposed to structured tasks, which are more predictable and repetitive. The sequence and frequency with which certain tasks are performed will also affect the optimal design of the user interface.

Functional Characteristics:

It refers to the various functions required to perform the tasks and the ease with which these functions can be learnt made use of by the users while ascertaining the functional characteristics, the support facilities required to perform the functions, also need to be taken into account. These include facilities like training, on line help, documentation, expert system etc. the other aspect of functional characteristics of a system its performance criteria like response time, fault tolerance etc.

INPUT DESIGN:

The most common cause of errors in data processing is inaccurate input data. Errors entered by data entry operators can be controlled by the input design. Input design is the process of converting user-oriented inputs to computer based formats. The goal of input design to make data entry easy logical and free from errors.

OUTPUT DESIGN:

Computers are the most important source of information to the user. Inputs are fed into computers to acquire the required outputs. The computers can provide valuable information's in the form of well-documented outputs for various values. The major form of output is a hardcopy (reports) from the printer. Reports are around the output requirements of the user.

4.2 *Software Design Specification:*

This online shopping system project section provides an overview of the entire design document. This online shopping system project document describes all data, architectural, interface and component-level design for the software.

4.2.1 Database Design:

A. Tables description:

Table 1:

Table Definition:

Table Name : **Category_Details:**

Purpose : *Store information about book category.*

Primary Key : CategoryID

Foreign Key :

 Columns Definition:

S. NO.	NAME	DATA TYPE	SIZE	NULLS ?	DEFAULT VALUE
1	**CategoryId**	Number	5	No	
2	**CategoryName**	VARCHAR2	25	No	

 Table Description:

S. NO.	NAME	DESCRIPTION
1	**BRANCH_NO**	A system generated number auto posted to this online shopping system project table column.
2	**NAME**	Name of the Category.

Table 2:

 Table Definition:

Table Name : **Book_Details**

Purpose : Stores details about books such as book title, author etc.

Primary Key : BookID

Foreign Key :
CATEGORY_DETAILS.CATEGORYID→BOOK_DETAILS.CATEGORY

 Columns Definition:

S.NO.	NAME	DATA TYPE	SIZE	NULLS ?	DEFAULT VALUE
1	**BookId**	Number	5	No	
2	**CategoryId**	Number	5	No	
3	**Title**	VARCHAR2	50	No	
4	**Author**	VARCHAR2	30	No	
5	**Publisher**	VARCHAR2	30	No	
6	**Edition**	VARCHAR2	5	No	
7	**Price**	Number	7,2	No	
8	**Quantity**	Number	3	No	
9	**Description**	Varchar2	100	Yes	

 Table description:

S.NO.	NAME	DESCRIPTION
1	**BookId**	A system generated number auto posted to this online shopping system project table column.
2	**CategoryID**	Reference to Category_Details
3	**Title**	Title of the Book.
4	**Author**	Author Name of the book.
5	**Publisher**	Store the Book Publisher.
6	**Edition**	Store the edition of the Book.
7	**Price**	Contain the Price Information.
8	**Quantity**	Contain the quantity available in store.
9	**Description**	Details of the book information

Table 3:

 Table Definition:

Table Name : **ODER_DETAILS**

Purpose : Stores information about the order placed by the end user.

Primary Key :

Foreign Key : ORDER_DETAILS.ORDERID ⟶ ORDER_DETAILS.ORDERID

- ***Columns Definition:***

S.NO.	NAME	DATATYPE	SIZE	NULLS?	DEFAULT VALUE
1	**ORDERID**	NUMBER	5	No	
2	**BOOKID**	NUMBER	5	No	
3	**QUANTITY**	NUMBER	3	No	

- ***Table Description:***

S.NO.	NAME	DESCRIPTION
1	**ORDERID**	A system generated number auto posted to this online shopping system project table column.
2	**BOOKID**	Reference to Book_Details table.
3	**QUANTITY**	Store the order quantity.

Table 4:

Table Definition:

Table Name : TEMP_DETAILS

Purpose : This online shopping system project table will holds the details about the book that online shopping system project report have been searched by the end user.

Primary Key :

Foreign Key :

Columns Definitions:

S.NO.	NAME	DATA TYPE	SIZE	NULLS ?	DEFAULT VALUE
1	**BookId**	Number	5	No	
2	**CategoryId**	Number	5	No	
3	**Title**	VARCHAR2	50	No	
4	**Author**	VARCHAR2	30	No	
5	**Publisher**	VARCHAR2	30	No	
6	**Edition**	VARCHAR2	5	No	
7	**Price**	Number	7,2	No	
8	**Quantity**	Number	3	No	
9	**Description**	Varchar2	100	Yes	

Table description:

S.NO.	NAME	DESCRIPTION
1	**BookId**	Store the book id of book details table.
2	**CategoryID**	Store the category id of book details table.
3	**Title**	Title of the Book.
4	**Author**	Author Name of the book.

5	**Publisher**	Store the Book Publisher.
6	**Edition**	Store the edition of the Book.
7	**Price**	Contain the Price Information.
8	**Quantity**	Contain the quantity available in store.
9	**Description**	Details of the book information

Table 5:

Table Definition:

Table Name : **USER_Profile**

Purpose : Store the information about all the registered user.

Primary Key : UserName

Foreign Key :

Columns Definition:

S. NO.	NAME	DATATYPE	SIZE	NULLS ?	DEFAULT VALUE
1	**USERNAME**	VARCHAR2	30	No	
2	**PASSWORD**	VARCHAR2	15	No	
3	**FIRSTNAME**	VARCHAR2	10	No	
4	**MIDDLENAME**	VARCHAR2	10	Yes	
5	**LASTNAME**	VARCHAR2	10	No	
6	**ADDRESS1**	VARCHAR2	40	No	
7	**ADDRESS2**	VARCHAR2	40	Yes	
8	**CITY**	VARCHAR2	40	No	
9	**STATE**	VARCHAR2	20	No	
10	**PINCODE**	VARCHAR2	10	No	
11	**EMAIL**	VARCHAR2	25	No	
12	**PHONE**	VARCHAR2	12	No	

 Table Description:

S. NO.	NAME	DESCRIPTION
1	**USERNAME**	Store the user name
2	**PASSWORD**	Store information of user password
3	**FIRSTNAME**	Store the first name of the user
4	**MIDDLENAME**	Store the middle name of the user.
5	**LASTNAME**	Store the information of the user last name.
6	**ADDRESS1**	Store the information of the user address.
7	**ADDRESS2**	Optional
8	**CITY**	User city
9	**STATE**	User state
10	**PINCODE**	State pin code
11	**EMAIL**	User email address
12	**PHONE**	User phone number

Table 6:

 Table Definition:

Table Name : **USER_AUTH**

Purpose : Stores the username and password of various end users.

Primary Key :

Foreign Key : USER_PROFILE.USERNAME→USER_AUTH.USERNAME

Columns definition:

S. NO.	NAME	DATATYPE	SIZE	NULLS ?	DEFAULT VALUE
1	**USERNAME**	VARCHAR2	30	No	
2	**PASSWORD**	VARCHAR2	15	No	

 Table description:

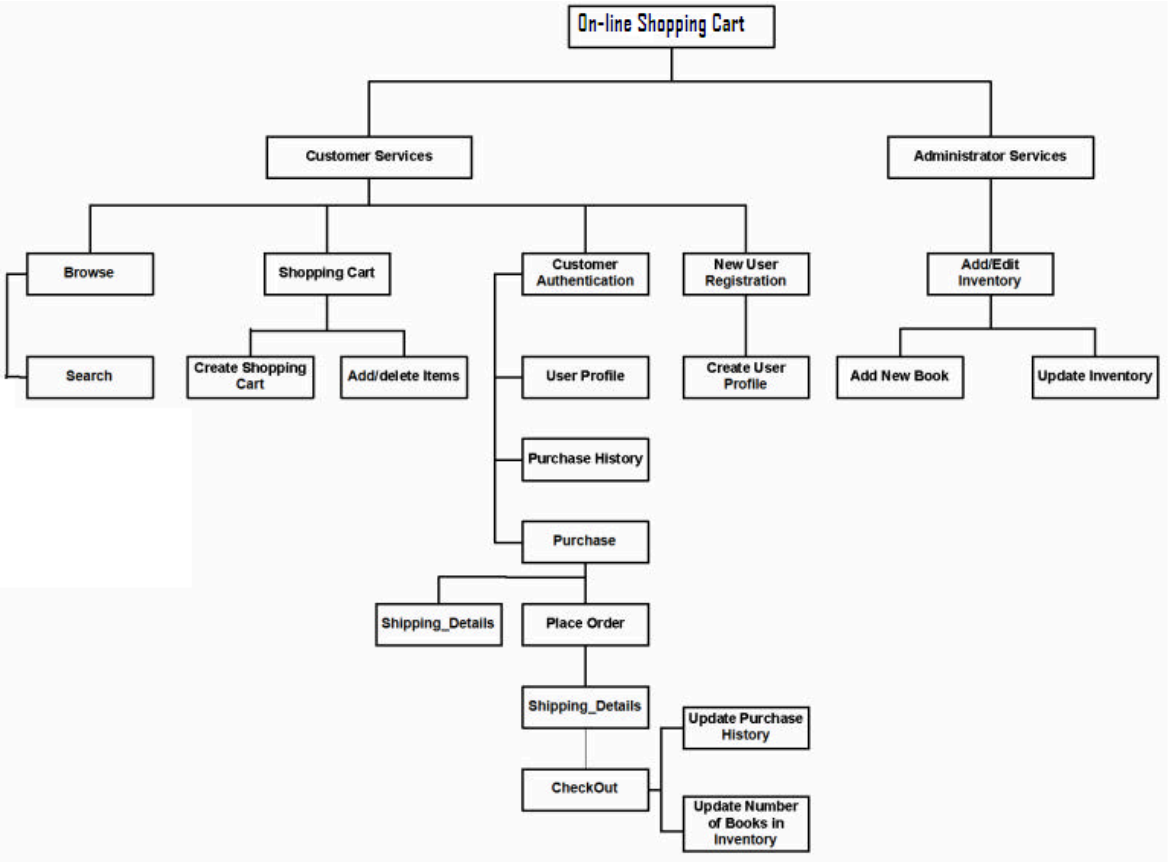
S. NO.	NAME	DESCRIPTION
1	**USERNAME**	Store the user name of the end users.
2	**PASSWORD**	Stores the password of the user.

4.2. Process Model

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This online shopping system project model consists of the
Functional Decomposition Diagram and Data Flow Diagram.

4.2.1. Functional Decomposition Diagram

A decomposition diagram shows a top-down functional decomposition of a system and exposes the system's structure. The objective of the Functional Decomposition is to break down a system step by step, beginning with the main function of a system and continuing with the interim levels down to the level of elementary functions. The diagram is the starting point for more detailed process diagrams, such as data flow diagrams (DFD). Figure 2 shows the Functional Decomposition Diagram for this online shopping system project project.



Functional Decomposition Diagram

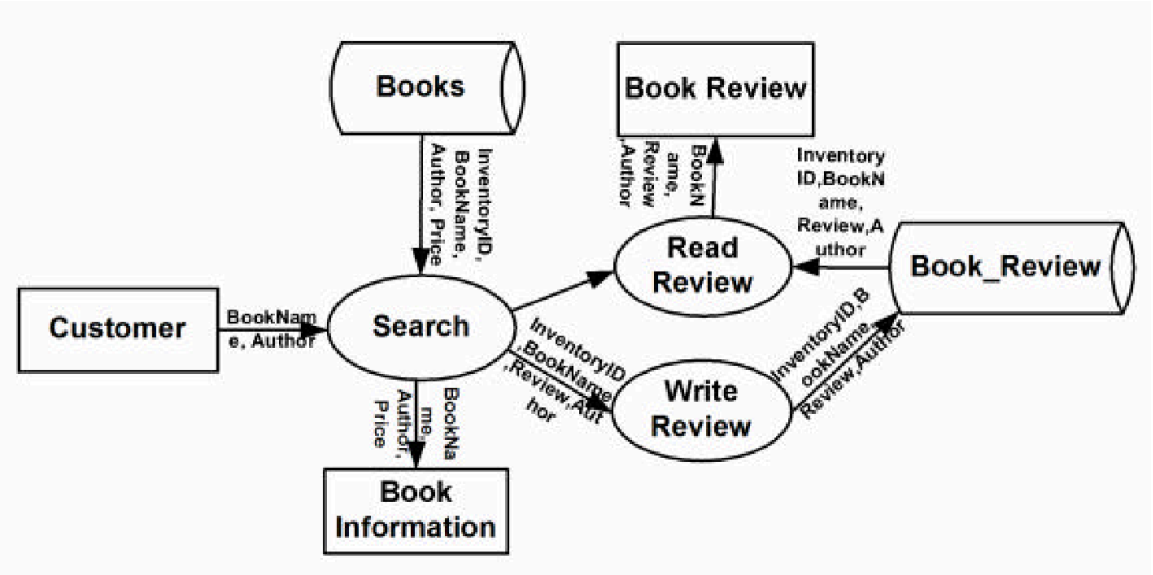
4.2.2 Data Flow Diagram (DFD)

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing a DFD:

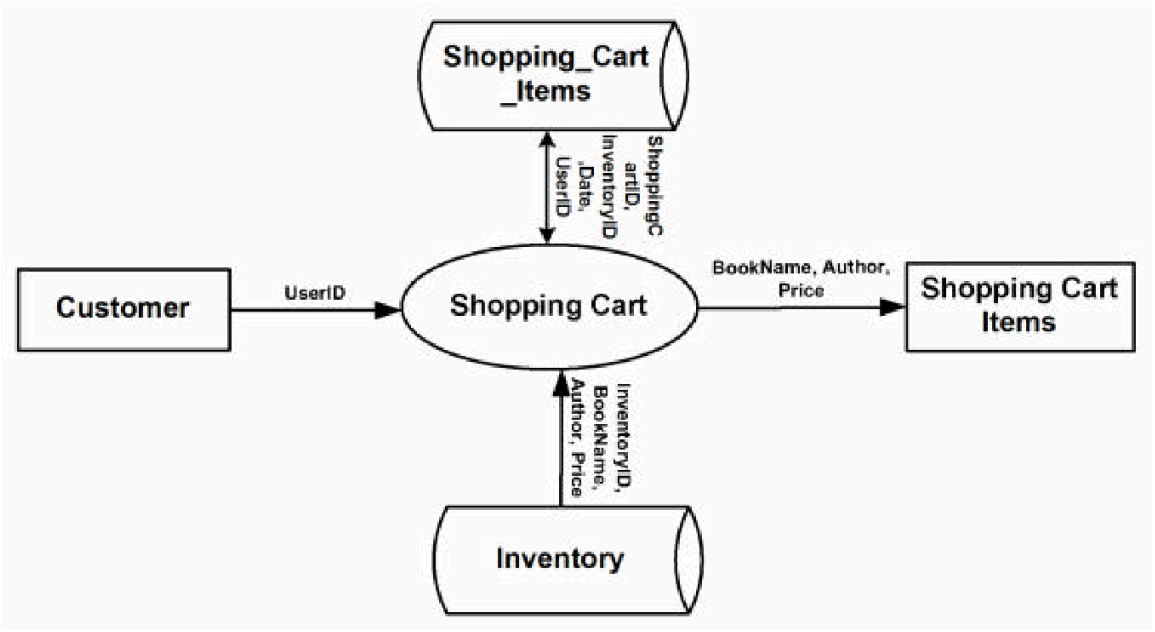
1. Rectangles representing *external entities*, which are sources or destinations of data.
2. Ellipses representing *processes*, which take data as input, validate and process it and output it.
3. . Arrows representing the *data flows*, which can either, be electronic data or physical items.
4. Open-ended rectangles or a Disk symbol representing *data stores*, including electronic stores such as databases or XML files and physical stores such as filing cabinets or stacks of paper.

Data Flow Diagrams for the current system. Each process within the system is first shown as a Context Level DFD and later as a Detailed DFD. The Context Level DFD provides a conceptual view of the process and its surrounding input,output and data stores. The Detailed DFD provides a more detailed and comprehensive view of the interaction among the sub-processes within the system.

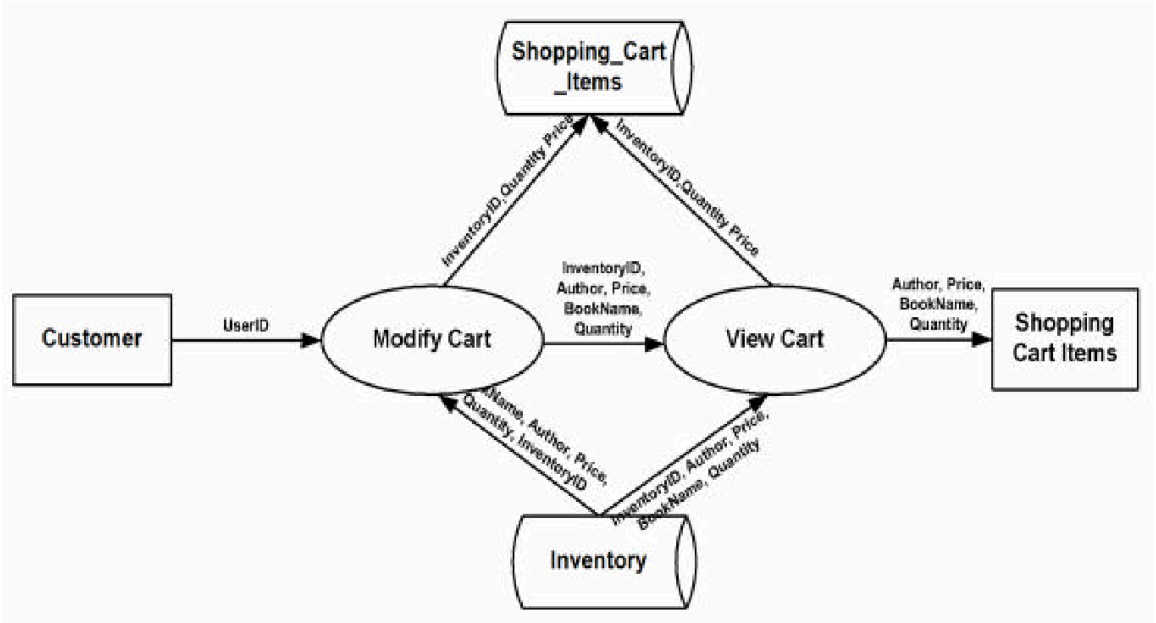
Customer-Browse Context DFD



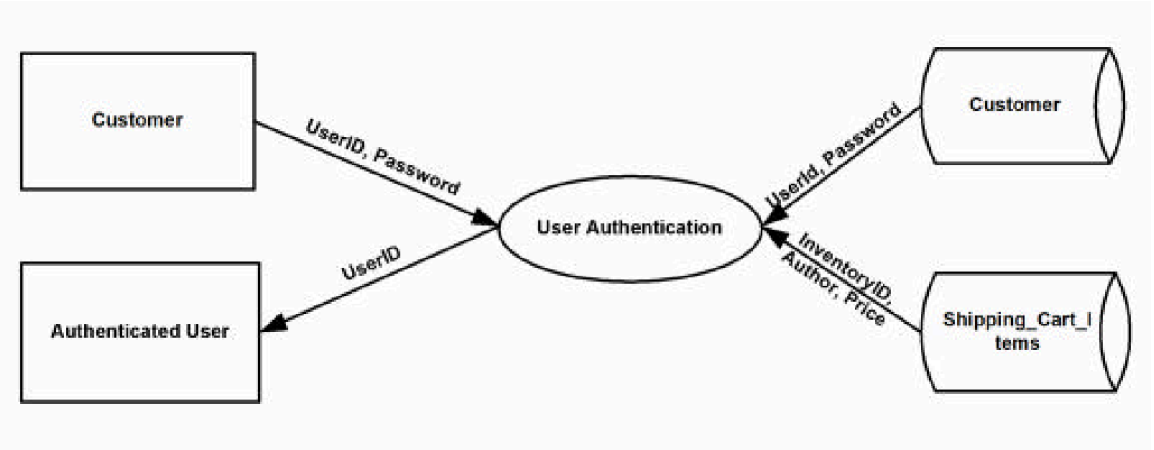
Customer - ShoppingCart Context DFD



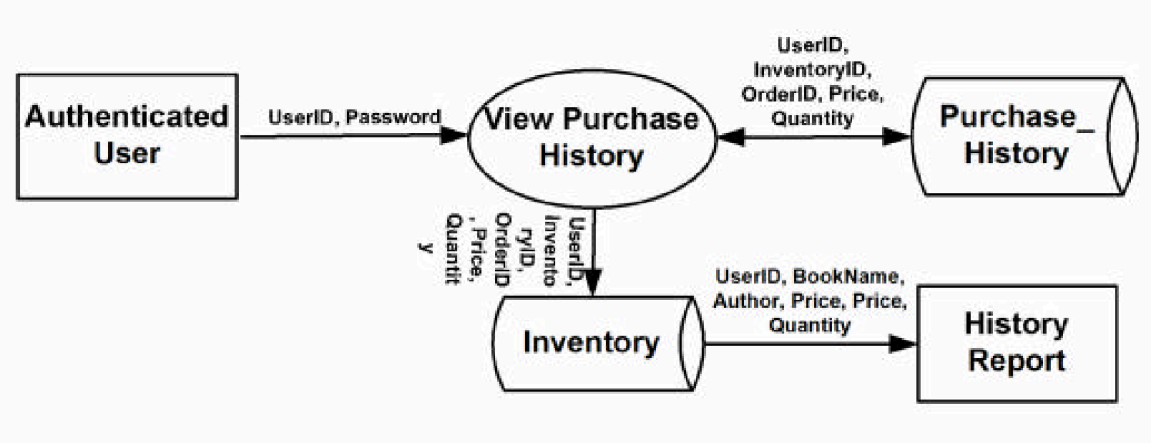
Customer - Shopping Cart Detailed DFD



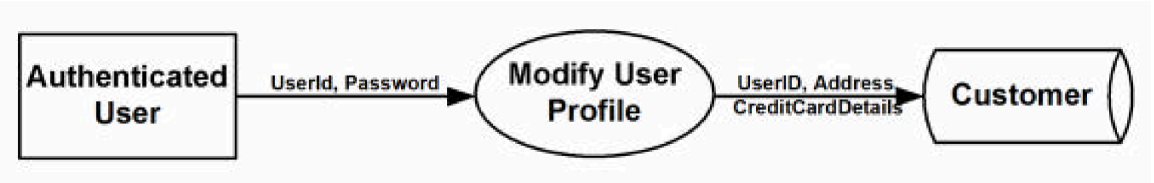
Customer-Authentication Context DFD



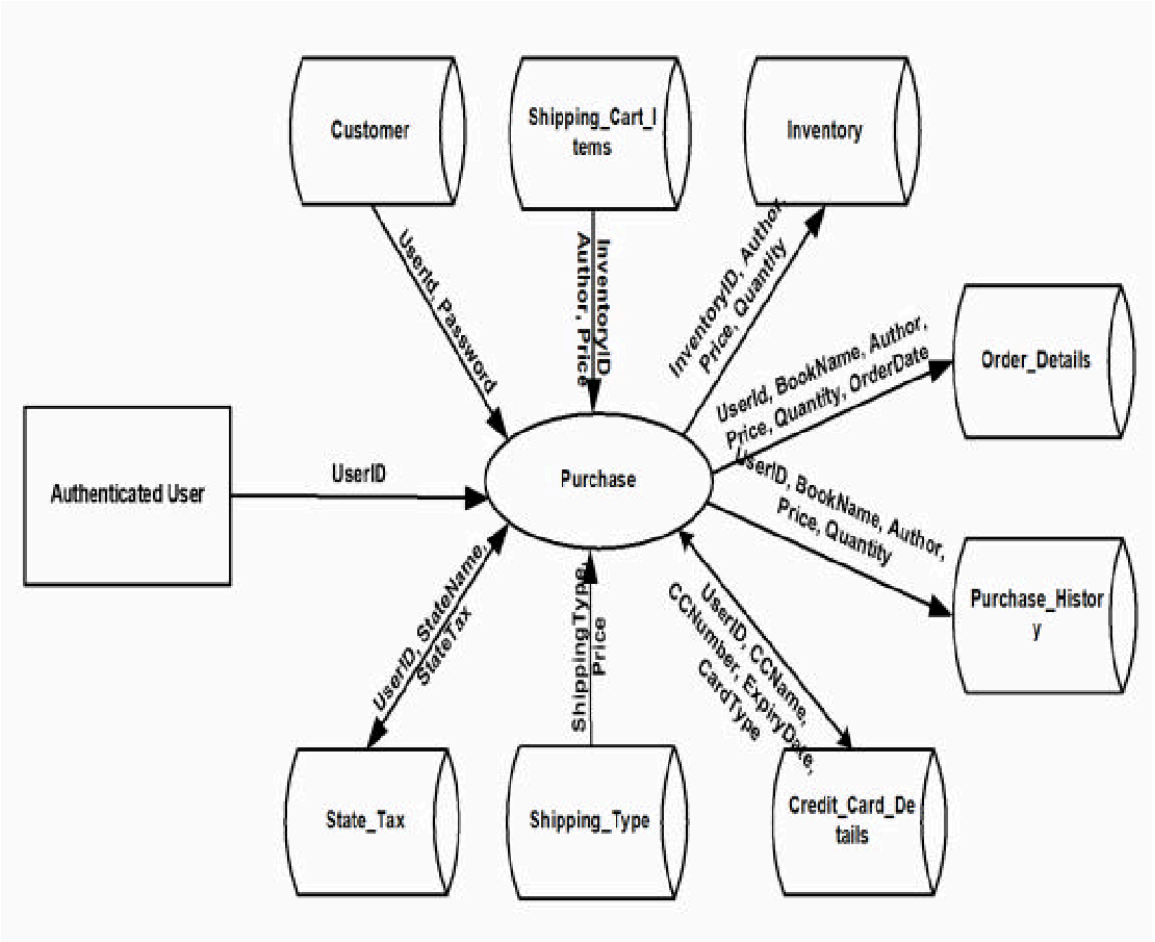
Customer-Authentication-PurchaseHistory DFD



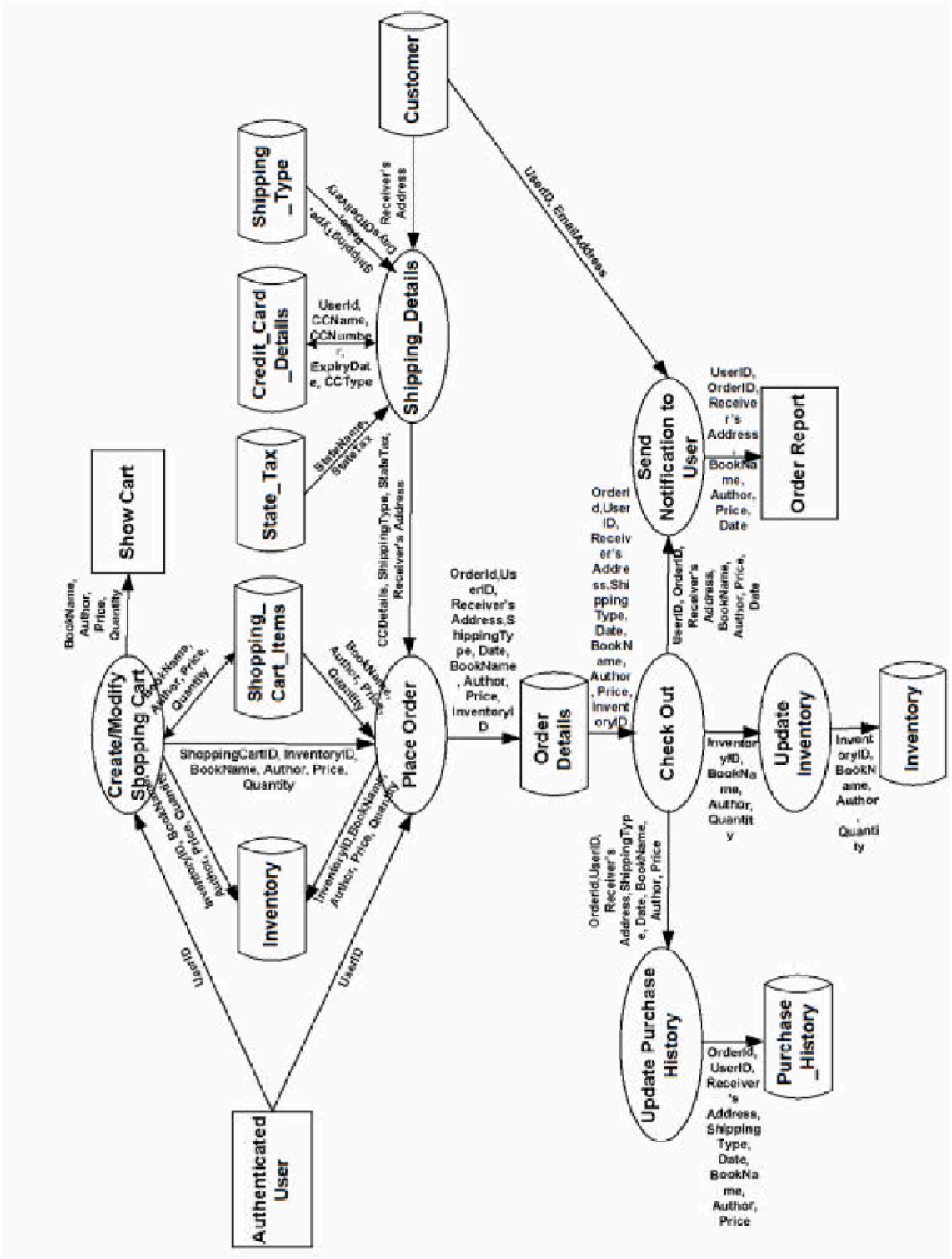
Customer-Authentication-UserProfile DFD



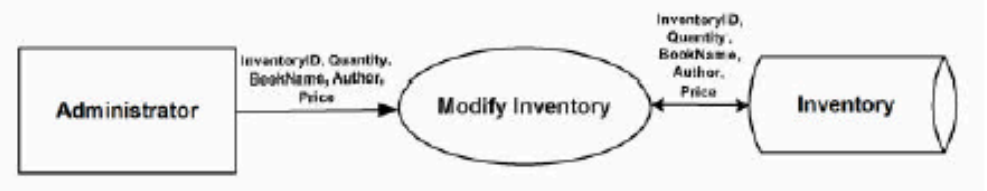
Authenticated User-Purchase Context DFD

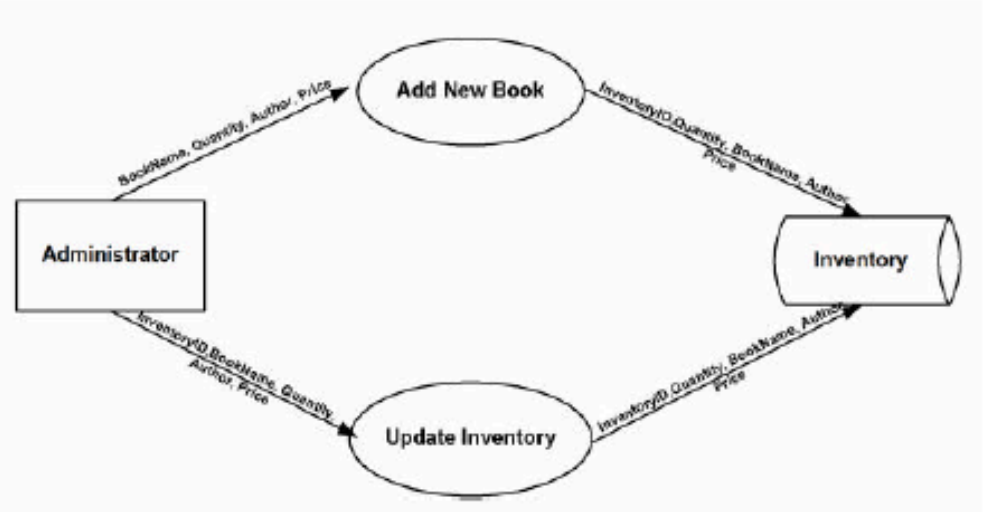


Authenticated User-Purchase DFD



Customer-NewUserRegistration DFD





Coding

Login.html

```
<html>
<head>
<title>Login</title>
</head>
<body bgcolor="#CCFFFF">
	<CENTER>
	<H1> <u>Online Shopping Cart</u> </H1>
	</CENTER>
	<SCRIPT language="JavaScript">
	function validate()
	{

if(document.frm.UserName.value==""||document.frm.Password.value=="")
		{
			alert("UserName or Password cannot be blank!!");
			return;
		}
		document.frm.action="Validate.jsp";
		document.frm.submit();
	}
	</SCRIPT>
	<form name = frm method="POST">
	<p align="center"><font size="5"><b>Login Form</b></font></p>
	<TABLE ALIGN="center" height="57">
	<TR>
		<TD >
	<b>UserName:</b>
		</TD>
		<TD >
		<input type="text" name="UserName" size="20" tabindex="1">
		</TD>
	</TR>
	<TR>
		<TD >
	<b>Password:</b>
		</TD>
		<TD >
		<input type="password" name="Password" size="20" tabindex="2">
		</TD>
	<TR align="center">
		<TD colspan=2>
		<input type="button" value="  Login  " name="B1" onclick="validate();"
tabindex="3">
		</TD>
	</TR>
	<TR >
```

```
                <TD ><font size="4">New User?</font></TD>
                <TD><a href="NewUser.htm"><font ize="4">Register</font></a></TD>
        </TR>
        </TABLE>
        </form>
</body>
</HTML>
```

NewUser.html

```
<html>
<head>
<title>New User Registration</title>
</head>
<body bgcolor="#CCFFFF">
<CENTER>
<H1> <u>Online Shopping Cart</u> </H1>
</CENTER>
<form name="form1" method="POST" >
 <SCRIPT language="JavaScript">
  function validate()
  {
  if( document.form1.UserName.value=="")
  {
  alert("UserName cannot be blank");
  return;
  }
  if( document.form1.Password.value=="")
  {
  alert("Password cannot be blank");
  return;
  }
  if( document.form1.Password.value!= document.form1.Password2.value)
  {
  alert("The two passwords do not match");
  return;
  }
  if( document.form1.FirstName.value=="")
  {
  alert("First Name cannot be blank");
 return;
  }
  if(document.form1.LastName.value=="")
  {
  alert("Last Name cannot be blank");
  return;
  }

  if( document.form1.Address1.value=="")
```

```
  {
  alert("Address cannot be blank");
return;
  }

  if( document.form1.City.value=="")
  {
  alert("City cannot be blank");
 return;
  }
  if( document.form1.State.value=="Select State")
  {
  alert("You must select a state");
return;
  }
  if( document.form1.PinCode.value=="")
  {
  alert("PinCode cannot be blank");
 return;
  }
  if( document.form1.Email.value=="")
  {
  alert("Email ID cannot be blank");
 return;
  }
if( document.form1.Phone.value=="")
  {
  alert("Phone Number cannot be blank");
 return;
  }
  document.forms[0].action="Sign.jsp";
  document.forms[0].submit();

  }
 </SCRIPT>

 <p align="center">
  <b><font size="5"> New User Registration</font></b> 
       
  </p>
 <p align="left">
   <b><font color="#FF0000">( Field marked * are
necessary)</font>         &nbs
p;            
;            
    </b>
  </p>
  <div align="left">
```

```
<table border="0" width="100%">
  <tr>
    <td width="23%"><b>User *:</b></td>
    <td width="77%">
      <b>
                    <input type="text" name="UserName" size="20" tabindex="1">
      </b>
    </td>
  </tr>
  <tr>
    <td width="23%">
      <b>
      Password*:</b></td>
    <td width="77%"><b><input type="password" name="Password" size="20"
tabindex="2">
      </b>
    </td>
  </tr>
  <tr>
    <td width="23%">
      <b>
      Confirm Password*:</b>
    </td>
    <td width="77%"><b><input type="password" name="Password2" size="20"
tabindex="3">
      </b>
    </td>
  </tr>
  <tr>
    <td width="23%">
      <b>
      First Name*:</b></td>
    <td width="77%">
      <b><input type="text" name="FirstName" size="20" tabindex="4">
      </b>
    </td>
  </tr>
  <tr>
    <td width="23%">
      <b>
      Middle Name:</b></td>
    <td width="77%">
      <b>
      <input type="text" name="MiddleName" size="20" tabindex="5">
      </b>
    </td>
  </tr>
  <tr>
```

```
      <td width="23%">
        <b>
        Last Name*:</b></td>
      <td width="77%">
        <b><input type="text" name="LastName" size="20" tabindex="6">
        </b>
      </td>
    </tr>
  </table>
 </div>
 <div align="center">
  <center>
  <table border="0" width="100%" height="201">
    <tr>
      <td width="23%" height="25"><b>Address1*:&amp;nbsp;&amp;nbsp;</b></td>
      <td width="77%" height="25"><b><input type="text" name="Address1" size="43"
tabindex="7"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>Address2:&amp;nbsp;</b></td>
      <td width="77%" height="25"><b><input type="text" name="Address2" size="43"
tabindex="8"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>City*:&amp;nbsp;</b></td>
      <td width="77%" height="25"><b><input type="text" name="City" size="20"
tabindex="9"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>State*:</b></td>
      <td width="77%" height="25"><b><select size="1" name="State" tabindex="10">
        <option selected value="Select State">Select State</option>
        <option>Assam</option>
        <option>Arunachal Pradesh</option>
        <option>Haryana</option>
        <option>Maharashtra</option>
        <option>Rajasthan</option>
        <option>Uttar pradesh</option>
       </select> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>Pin Code*:&amp;nbsp;</b></td>
      <td width="77%" height="25"><b><input type="text" name="PinCode" size="20"
tabindex="11"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>Email ID*:</b></td>
```

```
      <td width="77%" height="25"><b><input type="text" name="Email" size="20"
tabindex="12"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="25"><b>Phone Number*:</b></td>
      <td width="77%" height="25"><b><input type="text" name="Phone" size="20"
tabindex="13"> </b> </td>
    </tr>
    <tr>
      <td width="23%" height="1"></td>
      <td width="77%" height="1"> </td>
    </tr>
    <tr>
      <td width="23%" height="21"></td>
      <td width="77%" height="21">
  <b>
  <input type="submit" value="Submit" name="B1" onclick="validate();"
tabindex="13">
  </b>
 </td>
    </tr>
   </table>
   </center>
  </div>

</form>


</body>

</HTML>
```

Sign.jsp

```
<html>
<body bgcolor="#CCFFFF">
<%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>

<%
        Connection conn;
        conn=null;
        ResultSet rs;
        rs=null;

        String UserName=request.getParameter("UserName");
        String Password=request.getParameter("Password");
```

```
String Password2=request.getParameter("Password2");
String FirstName=request.getParameter("FirstName");
String MiddleName=request.getParameter("MiddleName");
String LastName=request.getParameter("LastName");
String Address1=request.getParameter("Address1");
String Address2=request.getParameter("Address2");
String City=request.getParameter("City");
String State=request.getParameter("State");
String PinCode=request.getParameter("PinCode");
String Email=request.getParameter("Email");
String Phone=request.getParameter("Phone");

int flag=0;

try
{
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
        PreparedStatement stat= conn.prepareStatement("SELECT * FROM
user_profile where username=?" );

stat.setString(1,UserName);
        rs = stat.executeQuery();
out.println("query executed");
        if(rs.next())
        {
                flag=1;
                %>
                <SCRIPT language="JavaScript">
                alert("User name already exists");
                location.href="NewUser.htm";
                </SCRIPT>
                <%
        }
        else
        {
                flag=0; // conn.close();
        }
}
catch(Exception E)
{
        out.println("Error  "+E);
}
if(flag == 0)
{
        try
        {
```

```
                PreparedStatement stat1=conn.prepareStatement("INSERT INTO
user_auth values(?,?)");
                        stat1.setString(1,UserName);
                        stat1.setString(2,Password);

                        PreparedStatement stat=conn.prepareStatement("INSERT INTO
user_profile values(?,?,?,?,?,?,?,?,?,?,?,?)");
                        stat.setString(1,UserName);
                        stat.setString(2,Password);
                        stat.setString(3,FirstName);
                        stat.setString(4,MiddleName);
                        stat.setString(5,LastName);
                        stat.setString(6,Address1);
                        stat.setString(7,Address2);
                        stat.setString(8,City);
                        stat.setString(9,State);
                        stat.setString(10,PinCode);
                        stat.setString(11,Email);
                        stat.setString(12,Phone);

                        stat.executeUpdate();
                stat1.executeUpdate();
                        flag=0;
                        response.sendRedirect("Login.htm");
                }
                catch(Exception E)
                {
                        out.println("Error inserting value"+E);
                }
                finally
                {
                        rs.close();
                        conn.close();
                }

        }

%>
</body>
</html>
```

Validate.jsp

```
<html>
<body>
<%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
<%
        Connection conn;
        conn=null;
        ResultSet rs;
        rs=null;
        try
        {
                String strName=request.getParameter("UserName");
                String strPass=request.getParameter("Password");
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                conn = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
                PreparedStatement stat= conn.prepareStatement("SELECT * FROM
user_auth where username=? and password =?");
                stat.setString(1,strName);
                stat.setString(2,strPass);
                rs = stat.executeQuery();
                if(rs.next())
                {

                        String str1,str2;
                        str1=rs.getString(1);
                        str2=rs.getString(2);
                        out.println(str1+str2);
                        session.putValue("user",strName);
                        if(str1.equals("Administrator"))
                        {
                                response.sendRedirect("Admin.htm");
                        }
                        else
                        {
                                out.println("Logged IN");
                                response.sendRedirect("SearchCriteria.jsp");
                        }
                }
                else
                {
                        response.sendRedirect("InvalidUser.htm");
                }
        }
        catch(Exception E)
        {
                out.println("Error "+E);
        }
        finally
```

```
        {
                rs.close();
                conn.close();
        }
        %>
</body>
</html>
```

SearchCriteria.jsp

```
<html>
<head>
<title>Search Items</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        </CENTER>
        <SCRIPT language="JavaScript">
        function startSearch()
        {
                if(document.frm.R1[1].checked)
                {
                        if(document.frm.BookTitle.value=="")
                        {
                                alert("You must enter a book title");
                                return;
                        }
                }

                if(document.frm.R1[2].checked)
                {
                        if(document.frm.BookAuthor.value=="")
                        {
                                alert("You must enter an author name");
                        return;
                        }
                }

                if(document.frm.R1[3].checked)
                {
                        if(document.frm.BookPublisher.value=="")
                        {
                        alert("You must enter a publisher name");
                        return;
                        }
                }

                document.frm.action="search.jsp";
                document.frm.submit();
```

```
	}

	</SCRIPT>

	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
	Connection con = null;

	PreparedStatement stat = null;
	ResultSet rs = null;
	int ctr=0, flag=0;
	try
	{
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");

		stat = con.prepareStatement("select * from category_details");

	  rs=stat.executeQuery();
	%>


	<form name ="frm" method="POST">
	<center><font size="5">Search Books</font></p></center>
	<BR><BR>

	<table border="0" width="100%" height="196">
	<tr>
		<td width="17%" align="right" height="25"><input type="radio"
value="Category" checked name="R1"></td>
		<td width="32%" height="25">Search by Category</td>
		<td width="51%" height="25">
			<select size="1" name="BookCategory">
			<option selected value="Select">Select Category</option>
				<%
				while(rs.next())
				{
				String category=rs.getString(2);%>
				<option
value="<%=category%>"><%=category%></option>
				<%}%>
			</select>
		</td>
	</tr>
	<tr>
```

```
		<td width="17%" align="right" height="25"><input type="radio"
name="R1" value="Title"></td>
		<td width="32%" height="25">Search by Book Title</td>
		<td width="51%" height="25"><input type="text" name="BookTitle"
size="25"></td>
	</tr>
	<tr>
		<td width="17%" align="right" height="25"><input type="radio"
name="R1" value="Author"></td>
		<td width="32%" height="25">Search by Book Author</td>
		<td width="51%" height="25"><input type="text" name="BookAuthor"
size="25"></td>
	</tr>
	<tr>
		<td width="17%" align="right" height="25"><input type="radio"
name="R1" value="Publisher"></td>
		<td width="32%" height="25">Search by Book Publisher</td>
		<td width="51%" height="25"><input type="text" name="BookPublisher"
size="25"></td>
	</tr>
	<tr>
	</tr>
	<tr>
		<td width="17%" align="right" height="21"></td>
		<td width="32%" height="21">  
    <input type="button" value="    Search    " name="Search"
onClick="startSearch();"></td>
		<td width="51%" height="21"><a href="Logout.jsp">Click here to
logout</a></td>
	</tr>
	</table>

	</form>
	<%
	}
	catch(Exception e)
	{
		out.print("Error = " + e + "<HR>");
	}
	finally
	{
		rs.close();
		con.close();
	}
	%>
</body>
</html>
```

```
<%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
<html>
<head>
<title>Search Results</title>
</head>
<body bgcolor="#CCFFFF">
<CENTER>
<H1> <u>Online Shopping Cart</u> </H1>
<font size="5" >Search Results</font>
</CENTER>
<BR>
<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
<%
        String str_searchvar=null;
        int chk_ctr=0;
        String str_colvar=null;
        String tab_var="";
        String user_src=(String)session.getValue("user");
        System.out.println("0");
        if (user_src!=null)
        {
                String query=null;
                Connection conn;
                conn=null;
                ResultSet rs;
                rs=null;
                if(request.getParameter("R1").equals("Title"))
                {
                        str_searchvar=request.getParameter("BookTitle");
                        str_colvar="TITLE";
                }
                if(request.getParameter("R1").equals("Category"))
                {
                        chk_ctr=1;
                        str_searchvar=request.getParameter("BookCategory");
                        str_colvar="Categoryname";
                }
                if(request.getParameter("R1").equals("Publisher"))
                {
                        str_searchvar=request.getParameter("BookPublisher");
                        str_colvar="Publisher";
                }
                if(request.getParameter("R1").equals("Author"))
                {
                        str_searchvar=request.getParameter("BookAuthor");
                        str_colvar="Author";
                }
                if (request.getParameter("R2")==null)
```

```
		{
			tab_var="BOOK_DETAILS";
		}
		else
		{
		// to get the name of the table in which the search to be done-------
			if (request.getParameter("R2").equals("A"))
			{
				tab_var="TEMP_DETAIL";
			}
			else
			{
				tab_var="BOOK_DETAILS";
			}
		}


	System.out.println(str_searchvar);
	query = "select a.* from "+tab_var+" a, category_details b where a."+str_colvar+"
= ? and a.CATEGORYID = b.CATEGORYID";
	System.out.println("1");
	try
	{
	Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
	conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
	int ctr=0;
	System.out.println("2");
	PreparedStatement stm = conn.prepareStatement(query);
	stm.setString(1,str_searchvar);
	rs = stm.executeQuery();
%>

			<form name="cart" action="Inter_Cart.jsp" method="POST">
			<table border=1>
			<tr>
			<td>S.No</td>
			<td>Select</td>
			<td>Book Name</td>
			<td>Author Name</td>
			<td>Publisher</td>
			<td>Edition</td>
			<td>Price (in Rs.)</td>
			<td>Quantity Available(Nos.)</td>
			<td>Description</td>
			</tr>

<%
	while(rs.next())
```

```
        {
        System.out.println("inside while");
        ctr=ctr+1;
        out.println("<tr><td>"+ctr+"</td>");
        int int_BOOKID=0;
        int_BOOKID=rs.getInt(1);
        System.out.println(int_BOOKID);
        out.println("<td><input type=checkbox name=chk"+ctr+"
value="+int_BOOKID+">");

        int int_CATEGORYID=rs.getInt(2);

        String str_TITLE=rs.getString(3);
        out.println("<td>");
        out.println(str_TITLE);
        out.println("</td>");

        String str_AUTHOR=rs.getString(4);
        out.println("<td>");
        out.println(str_AUTHOR);
        out.println("</td>");


        String str_PUBLISHER=rs.getString(5);
        out.println("<td>");
        out.println(str_PUBLISHER);
        out.println("</td>");


        String str_EDITION=rs.getString(6);
        out.println("<td>");
        out.println(str_EDITION);
        out.println("</td>");

        int int_PRICE=rs.getInt(7);
        out.println("<td>");
        out.println(int_PRICE);
        out.println("</td>");

        int int_QUANTITY=rs.getInt(8);
        out.println("<td>");
        out.println(int_QUANTITY);
        out.println("</td>");

        String str_DESCRIPTION=rs.getString(9);
        out.println("<td>");
        out.println(str_DESCRIPTION);
        out.println("</td></tr>");
```

```
        String ins_query ="INSERT INTO temp_detail VALUES (?,?,?,?,?,?,?,?,?)";
        PreparedStatement stat_ins= conn.prepareStatement(ins_query);
        stat_ins.setInt(1,int_BOOKID);
        stat_ins.setInt(2,int_CATEGORYID);
        stat_ins.setString(3,str_TITLE);
        stat_ins.setString(4,str_AUTHOR);
        stat_ins.setString(5,str_PUBLISHER);
        stat_ins.setString(6,str_EDITION);
        stat_ins.setInt(7,int_PRICE);
        stat_ins.setInt(8,int_QUANTITY);
        stat_ins.setString(9,str_DESCRIPTION);
        int x = stat_ins.executeUpdate();
        System.out.println(x);
        }
                        String str_ctr=String.valueOf(ctr);
                        session.putValue("ctr_val",str_ctr);

%>
        </table>
        <br>
        <input type="submit" value="    Add to Cart    " name="Add">
        </form>

<%

        rs.close();
        stm.close();
        }
        catch(Exception e){}
        }
        else
                {
                        response.sendRedirect("Unauthorised.htm");
                }
%>
</body>
</html>
```

Inter_Cart.jsp

```
<%@ page errorPage="errorpage.jsp" language="java" %>

<%
        int i;
        String bookid=null;
        String user_src=null;
        user_src=(String)session.getValue("user");
        if (user_src!=null)
        {
                int counter=0;
```

```
                counter=Integer.parseInt((String)session.getValue("ctr_val"));
                int ctr=0;
                for(i=1;i<=counter;i++)
                {
                        if(request.getParameter("chk"+i)!=null)
                        {
                                bookid=request.getParameter("chk"+i);
                                ctr=ctr+1;
                                session.putValue("chk_var"+ctr,bookid);
                        }
                }
                session.putValue("TotalSel",String.valueOf(ctr));
                response.sendRedirect("Cart.jsp");
        }
        else
        {
                response.sendRedirect("Unauthorised.htm");
        }
%>
```

Cart.jsp

```
<html>
<head>
<title>Shopping Cart</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        </CENTER>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>

        <%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
        <%
        int i;
        String user_src=null;
        user_src=(String)session.getValue("user");
        if (user_src!=null)
        {
                int counter=0;
                counter=Integer.parseInt((String)session.getValue("TotalSel"));
                int[] bookid=new int[counter];
                Connection conn;
                conn=null;
                ResultSet rs;
                rs=null;
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
```

```
PreparedStatement stat=null;
int ctr=0,bookprice=0;
for(i=1;i<=counter;i++)
{
	if((String)session.getValue("chk_var"+i)!=null)
	{

bookid[i-1]=Integer.parseInt((String)session.getValue("chk_var"+i));
	}
}
session.putValue("BookArr",bookid);
%>

<h3>Your Cart Details are as follows:</h3>
<form name="shop" action="Cart_Quantity.jsp">
<table border=1>
<tr>
<td>S.No</td>
<td>Book Name</td>
<td>Price (in Rs.)</td>
</tr>
<%
int price=0;
for (int j=1;j<=counter;j++)
{
	out.println("<tr><td>"+j);


	String query="select title, price from book_details where
bookid=?";
	stat= conn.prepareStatement(query);
	stat.setInt(1,bookid[j-1]);

	rs = stat.executeQuery();
	while (rs.next())
	{
		out.println("<td>");
		out.println(rs.getString(1));
		out.println("<td>");
		price=rs.getInt(2);
		out.println(price);
	}
	rs.close();
	bookprice=bookprice+price;
	session.putValue("bPrice",String.valueOf(bookprice));
	out.println("</tr>");

}
```

```
ResultSet rscombo=null;
String query1="select * from category_details";
PreparedStatement stat_combo = conn.prepareStatement(query1);

rscombo=stat_combo.executeQuery();%>
<tr>
	<td colspan=3>Total Cart Amount(in Rs.)=
	<%=bookprice%><td>
</tr>
</table>
<br>
<input type=submit value="Place Order">
</form>
<br><hr><h3>Search More</h3>

<form name="search" action="NewSearch.jsp" method="POST">
<table border=1>
<tr>
	<td><input type="radio" value="Category" checked
name="R1"></td>
	<td>Search based on Category</td>
	<td>
	<select size="1" name="BookCategory">
	<option selected value="Select">Select Category</option>
		<%
		while(rscombo.next())
		{
			String category=rscombo.getString(2);
		%>
		<option
value="<%=category%>"><%=category%></option>
		<%}%>
	</select>
	</td>
</tr>
<tr>
	<td><input type="radio" name="R1" value="Title"></td>
	<td>Search by Book Title</td>
	<td><input type="text" name="BookTitle" size="25"></td>
</tr>
<tr>
<td><input type="radio" name="R1" value="Author"></td>
	<td>Search by Book Author</td>
	<td><input type="text" name="BookAuthor" size="25"></td>
</tr>
<tr>
	<td><input type="radio" name="R1" value="Publisher"></td>
	<td>Search by Book Publisher</td>
```

```
                    <td><input type="text" name="BookPublisher" size="25"></td>
                </tr>
                <tr>
                    <td><input type="radio" name="R2" value="A">Advanced
Search</td>
                    <td><input type="radio" name="R2" value="N" checked>New
Search</td>
                    <td><input type=submit value="Search"></td>
                </tr>
                </table>
                </form>
                <%}
        else
        {
                response.sendRedirect("Unauthorised.htm");
        }
        %>
</body>
</html>
```

Cart_quantity.jsp

```
<html>
<head>
<title>Shopping Cart</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        </CENTER>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
        <%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
        <%
        int i;
        String user_src=null;
        user_src=(String)session.getValue("user");
        if (user_src!=null)
        {
          int counter=0;
          counter=Integer.parseInt((String)session.getValue("TotalSel"));%>

        <script language="Javascript">
        function check_sum()
        {
                var cost=0;
                x_avlqty=new Array(<%=counter%>);
                y_orderqty=new Array(<%=counter%>);
                z_price=new Array(<%=counter%>);
                for (var i=0;i<<%=counter%>;i++)
```

```
		{
			z_price[i]=(document.shop.elements[i*4].value);
			x_avlqty[i]=document.shop.elements[i*4+1].value;
			y_orderqty[i]=document.shop.elements[i*4+2].value;
			if(parseInt(y_orderqty[i])<0)
			{
				alert("Book quantity cannot be Negative");
				return false;
			}
			if (parseInt(x_avlqty[i])<parseInt(y_orderqty[i]))
			{
				alert("Order Quantity Cannot be greater than available
quantity");
				return false;
			}
			else
			{

document.shop.elements[i*4+3].value=((parseInt(y_orderqty[i]))*(parseInt(z_price[i])));
				cost=cost+parseInt(document.shop.elements[i*4+3].value);
			}
		}

		document.shop.tot_val.value=cost;
	}
	</script>
	<%
		int[] bookid=new int[counter];
		Connection conn;
		conn=null;
		ResultSet rs;
		rs=null;
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");

		PreparedStatement stat=null;
		int ctr=0,bookprice=0;
		for(i=1;i<=counter;i++)
		{
			if((String)session.getValue("chk_var"+i)!=null)
			{

bookid[i-1]=Integer.parseInt((String)session.getValue("chk_var"+i));
			}
		}
		session.putValue("BookArr",bookid);
	%>
```

```
<h3>Your Cart Details are as follows:</h3>
<form name="shop" action="Shop.jsp" method="POST">
<table border=1>
<tr>
<td>S.No</td>
<td>Book Name</td>
<td>Price per unit (in Rs.)</td>
<td>Quantity Available</td>
<td>Quantity Purchased</td>
<td>Amount</td>
</tr>
<%
int price=0;
int qty_avbl=0;
for (int j=1;j<=counter;j++)
{
	out.println("<tr><td>"+j);
	out.println("</td>");

	String query="select title, price,quantity from book_details where
bookid=?";
	stat= conn.prepareStatement(query);
	stat.setInt(1,bookid[j-1]);
	rs = stat.executeQuery();

	while (rs.next())
	{
		out.println("<td>");
		out.println(rs.getString(1));out.println("</td>");
		price=rs.getInt(2);%>
		<td><input type="hidden" name="bk_price<%=j%>"
value=<%=price%></td>
		<%out.println(price);
		qty_avbl=rs.getInt(3);%>
		<td><input type=hidden name="qty_avl<%=j%>"
value=<%=qty_avbl%>><%=qty_avbl%></td>
		<td><input type=text name="qty<%=j%>" value="0"
onChange="return check_sum();"></td>
		<td><input type=text name="cal_amt<%=j%>"></td>
	<%
	}%>
	</tr>

	<%
	rs.close();
	bookprice=bookprice+price;
	session.putValue("bPrice",String.valueOf(bookprice));
```

```
                        out.println("</tr>");
                }%>
                <tr><td><td><td><td><td><td>Total Amount(in Rs.)<td><input type="text"
name="tot_val" value="0"></td></tr>
                </table>
                <br>
                <input type=submit value="Place Order">
                </form>
                <br><hr>
<%}
else
        {
                response.sendRedirect("Unauthorised.htm");
        }
%>

</body>
</html>
```

NewSearch.jsp

```
<html>
<head>
<title>Search Results</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        </CENTER>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
        <%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
        <%
        String user_src=(String)session.getValue("user");
        if (user_src!=null)
        {
        String str_searchvar=null;
        String str_colvar=null;
        String query="";
        String tab_var="";
        int total_sel=Integer.parseInt((String)session.getValue("TotalSel"));
        int[] old_bookid=new int[total_sel];
        Connection conn;
        conn=null;
        ResultSet rs;
        rs=null;
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
        PreparedStatement stat=null,stat_ins=null,stat_del=null;
        for (int i=1;i<=total_sel;i++)
```

```
{
        old_bookid[i-1]=Integer.parseInt((String)session.getValue("chk_var"+i));

}%>
        <h3>Your Cart Details are as follows:</h3>
        <form name="shop" action="Cart_quantity.jsp">
        <table border=1>
        <tr>
        <td>S.No</td>
        <td>Book Name</td>
        <td>Price (in Rs.)</td>
        </tr>
        <%
        int price=0; int bookprice=0;
        for (int j=0;j<total_sel;j++)
        {
                out.println("<tr><td>"+j+1);
                query="select title, price from book_details where bookid=?";
                stat= conn.prepareStatement(query);
                stat.setInt(1,old_bookid[j]);

                rs = stat.executeQuery();
                while (rs.next())
                {
                        out.println("<td>");
                        out.println(rs.getString(1));
                        out.println("<td>");
                        price=rs.getInt(2);
                        out.println(price);
                }
                rs.close();
                bookprice=bookprice+price;
                out.println("</tr>");
        }%>
        <tr>
        <td colspan=3>Total Cart Amount(in Rs.)=
        <%=bookprice%>
        <td></tr>
        </table>
        <br>
        <input type=submit value="Place Order">
        </form>
<%
String test_src=null;
int chk_ctr=0;


if(request.getParameter("R1").equals("Title"))
```

```
{
	str_searchvar=request.getParameter("BookTitle");
	str_colvar="Title";

}
if(request.getParameter("R1").equals("Category"))
{
	chk_ctr=1;
	str_searchvar=request.getParameter("BookCategory");
	str_colvar="Categoryname";
}

if(request.getParameter("R1").equals("Publisher"))
{
	str_searchvar=request.getParameter("BookPublisher");
	str_colvar="Publisher";
}
if(request.getParameter("R1").equals("Author"))
{
	str_searchvar=request.getParameter("BookAuthor");
	str_colvar="Author";
}

String ss = request.getParameter("R2");

if (request.getParameter("R2")==null)
{
	tab_var="book_details";
}
else
{
// to get the name of the table in which the search to be done-------
	if (request.getParameter("R2").equals("A"))
	{

		tab_var="temp_detail";
	}
	else
	{
		tab_var="book_details";

	}
}


//-------- whether search is by category or other criteria-----------
if (chk_ctr==0)
{
```

```
		query="select a.* from "+tab_var+" a, category_details b where
a."+str_colvar+" = ? and a.CATEGORYID = b.CATEGORYID";
	}
	else
	{

		query="SELECT a.*,b.categoryname FROM "+tab_var+"
a,CATEGORY_DETAILS b where b."+str_colvar+"=? and a.categoryid=b.categoryid";
	}

	try
	{
		stat=conn.prepareStatement(query);
		stat.setString(1,str_searchvar);


		rs = stat.executeQuery();

		int ctr=0;%>
		<h3>Search Results are as follows:</h3>
		<form name="cart" action="NewInter_cart.jsp" method="POST">
		<table border=1>
		<tr>
		<td>S.No</td>
		<td>Select</td>
		<td>Book Name</td>
		<td>Author Name</td>
		<td>Publisher</td>
		<td>Edition</td>
		<td>Price (in Rs.)</td>
		<td>Quantity Available(Nos.)</td>
		<td>Description</td>
		</tr>
		<%
		String delete_query="Delete from TEMP_DETAIL";
		stat_del= conn.prepareStatement(delete_query);
		String ins_query=null;
		int int_BOOKID,int_CATEGORYID,int_PRICE,int_QUANTITY;
		String
str_TITLE,str_AUTHOR,str_PUBLISHER,str_EDITION,str_DESCRIPTION;

		int rs_del=stat_del.executeUpdate();
		while(rs.next())
		{
			ctr=ctr+1;
			out.println("<tr><td>"+ctr);
			int_BOOKID=rs.getInt(1);
```

```
                    out.println("<td><input type=checkbox name=chk"+ctr+"
value="+int_BOOKID+">");

                    int_CATEGORYID=rs.getInt(2);

                    str_TITLE=rs.getString(3);
                    out.println("<td>");
                    out.println(str_TITLE);

                    str_AUTHOR=rs.getString(4);
                    out.println("<td>");
                    out.println(str_AUTHOR);



                    str_PUBLISHER=rs.getString(5);
                    out.println("<td>");
                    out.println(str_PUBLISHER);

                    str_EDITION=rs.getString(6);
                    out.println("<td>");
                    out.println(str_EDITION);

                    int_PRICE=rs.getInt(7);
                    out.println("<td>");
                    out.println(int_PRICE);

                    int_QUANTITY=rs.getInt(8);
                    out.println("<td>");
                    out.println(int_QUANTITY);

                    str_DESCRIPTION=rs.getString(9);
                    out.println("<td>");
                    out.println(str_DESCRIPTION);


                    out.println("</tr>");

                    ins_query="INSERT INTO TEMP_DETAIL
VALUES(?,?,?,?,?,?,?,?,?)";
                    stat_ins=conn.prepareStatement(ins_query);
                    stat_ins.setInt(1,int_BOOKID);
                    stat_ins.setInt(2,int_CATEGORYID);
               stat_ins.setString(3,str_TITLE);
                    stat_ins.setString(4,str_AUTHOR);
                    stat_ins.setString(5,str_PUBLISHER);
                    stat_ins.setString(6,str_EDITION);
                    stat_ins.setInt(7,int_PRICE);
```

```
	stat_ins.setInt(8,int_QUANTITY);
		stat_ins.setString(9,str_DESCRIPTION);


	int rs_ins=stat_ins.executeUpdate();

	}
	String str_ctr=String.valueOf(ctr);
	session.putValue("ctr_val",str_ctr);

	%>


</table>

<input type="submit" value="    Add to Cart    " name="Add">
</form>
<%

ResultSet rscombo=null;
	String query1="select * from category_details";
	PreparedStatement stat_combo = conn.prepareStatement(query1);

	rscombo=stat_combo.executeQuery();%>

<br><h3>Search Results are as follows:</h3>
<form name="newsrc" action="NewSearch.jsp" method="POST">
<table border=1>
	<tr>
<td><input type="radio" value="Category" checked name="R1"></td>
<td>Search based on Category</td>
<td>
 <select size="1" name="BookCategory">
	<option selected value="Select">Select Category</option>
		<%
		while(rscombo.next())
		{
			String category=rscombo.getString(2);
			%>
		<option value="<%=category%>"><%=category%></option>
		<%}%>
	</select>
</td>
</tr>
	<tr>
	<td><input type="radio" name="R1" value="Title"></td>
	<td>Search by Book Title</td>
	<td><input type="text" name="BookTitle" size="25"></td>
```

```
		</tr>
		<tr>
		<td><input type="radio" name="R1" value="Author"></td>
			<td>Search by Book Author</td>
			<td><input type="text" name="BookAuthor" size="25"></td>
		</tr>
		<tr>
			<td><input type="radio" name="R1" value="Publisher"></td>
			<td>Search by Book Publisher</td>
			<td><input type="text" name="BookPublisher" size="25"></td>
		</tr>

		<tr>
			<td><input type="radio" name="R2" value="A">Advanced Search</td>
			<td><input type="radio" name="R2" value="N" checked>New
Search</td>
			<td><input type=submit value="Search"></td>
		</tr>
		</table>
		</form>
		<%}
		catch(Exception E)
		{
			out.println("Error  "+E);
		}
		finally{conn.close();}


}
else
		{
			response.sendRedirect("Unauthorised.htm");
		}
%>
</body>
</html>
```

Shop.jsp

```
<html>
<head>
<title>Order Details</title>
</head>
<body bgcolor="#CCFFFF">
		<CENTER>
		<H1> <u>Online Shopping Cart</u> </H1>
		</CENTER>
		<BR>
		<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
		<%@  page errorPage="errorpage.jsp" language="java"  import="java.sql.*"  %>
```

```
<%
if (session.getValue("user")!=null)
{%>
<script language="Javascript">
function validate()
{
var r = document.form1;
var creditcard=r.Card.value;
var year = r.Year.value;
var month = r.Month.value;
var day = r.Day.value;

if(creditcard.length!=16)
{
        alert("Invalid Credit Card Number");
        r.Card.focus();
        return;
}
//---------------------------------------------------------------------

if(day=="Select Day")
{
        alert("Please select the day");
        r.Day.focus();
        return;
}
//---------------------------------------------------------------------
if(month=="Select Month")
{
        alert("Please select the month");
        r.Month.focus();
        return;
}
//---------------------------------------------------------------------
if(year.length==0)
{
        alert("Please enter the year");
        r.Year.focus();
}
else if (year.length!=4)
{
        alert("Please enter the year in YYYY format");
        r.Year.focus();
}
//---------------------------------------------------------------------
if (( month == "3" || month == "5" || month == "8" || month == "10") && (day ==
"31"))
    {
```

```
        alert("Please enter a valid date");
            r.Day.focus();
        }
            if ((month == "1") && (day == "30" || day == "31"))
        {
            alert("Please enter a valid date");
            r.Day.focus();
        }

        //------- To check that online shopping system project report card date  is not less
than current date----------
        var vr_day;
            var vr_month;
            var vr_year;
            var d=new Date();
            vr_year=d.getFullYear();
            vr_month=d.getMonth();
            vr_day=d.getDate();

            if(parseInt(vr_year)>parseInt(document.form1.Year.value))
            {
                    alert("Year cannot be less than Current Year");
                    return false;
            }
            else if(parseInt(vr_year)==parseInt(document.form1.Year.value))
            {
                    if(parseInt(vr_month)>parseInt(document.form1.Month.value))
                    {
                            alert("Month cannot be less than Current Month");
                            return false;
                    }
                    else if(parseInt(vr_month)==parseInt(document.form1.Month.value))
                    {
                            if(parseInt(vr_day)>parseInt(document.form1.Day.value))
                            {
                                    alert("Day cannot be less than current Day");
                                    return false;
                            }
                    }
            }

            //---------------------------------------------------------------------
            document.form1.submit();
            document.forms[0].action="Final.jsp";
            }
            </script>
            <%
                    int counter=0;
```

```
            counter=Integer.parseInt((String)session.getValue("TotalSel"));
            String query="";
            int[] bookid=new int[counter];
            int[] bk_price=new int[counter];
            int[] bk_quantity=new int[counter];

        Connection conn;
            conn=null;
            ResultSet rs;
            rs=null;
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");

            PreparedStatement stat=null;
            int ctr=0,bookprice=0;
            int i=1;
            while(i<=counter)
            {
                if((String)session.getValue("chk_var"+i)!=null)
                {

bookid[i-1]=Integer.parseInt((String)session.getValue("chk_var"+i));

                    query="SELECT PRICE FROM BOOK_DETAILS
WHERE BOOKID=?";
                    stat=conn.prepareStatement(query);
                    stat.setInt(1,bookid[i-1]);

                    rs=stat.executeQuery();
                    if(rs.next())
                    {
                        bk_price[i-1]=rs.getInt(1);
                    }
                    else
                    {
                        bk_price[i-1]=0;
                    }
                }
                if(request.getParameter("qty"+i)!=null)
                {

bk_quantity[i-1]=Integer.parseInt(request.getParameter("qty"+i));
                    String sess_var=String.valueOf(bk_quantity[i-1]);

session.putValue(sess_var,String.valueOf(bk_quantity[i-1]));
                }
                bookprice=bookprice+(bk_price[i-1]*bk_quantity[i-1]);
                i++;
```

```
                }
                session.putValue("bPrice",String.valueOf(bookprice));
        %>

        <form method="POST" action="Final.jsp" name="form1">
        <table width="70%">
        <tr>
                <td colspan=4 ><b>Total Amount (in Rs.) is :
<%=bookprice%></b></td></tr>
        <tr>
        </tr>
        <tr>
                <td><font size="3"><b>Credit Card Number: </b></font></td>
                <td colspan=3>
        <input type="text" name="Card" size="39" maxlength="16" tab="1"></td>
        </tr>
        <tr>
                <td><font size="3"><b>Expiry Date :     </b></font></td>
                <td>

                <select size="1" name="Day">
        <option value="Select Day">Select Day</option>
        <option value="1">1</option>
        <option value="2">2</option>
        <option value="3">3</option>
        <option value="4">4</option>
        <option value="5">5</option>
        <option value="6">6</option>
        <option value="7">7</option>
        <option value="8">8</option>
        <option value="9">9</option>
        <option value="10">10</option>
        <option value="11">11</option>
        <option value="12">12</option>
        <option value="13">13</option>
        <option value="14">14</option>
        <option value="15">15</option>
        <option value="16">16</option>
        <option value="17">17</option>
        <option value="18">18</option>
        <option value="19">19</option>
        <option value="20">20</option>
        <option value="21">21</option>
        <option value="22">22</option>
        <option value="23">23</option>
        <option value="24">24</option>
        <option value="25">25</option>
        <option value="26">26</option>
```

```
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
</td>
<td>
<select size="1" name="Month">
<option value="Select Month">Select Month</option>
<option value="0">January</option>
<option value="1">February</option>
<option value="2">March</option>
<option value="3">April</option>
<option value="4">May</option>
<option value="5">June</option>
<option value="6">July</option>
<option value="7">August</option>
<option value="8">September</option>
<option value="9">October</option>
<option value="10">November</option>
<option value="11">December</option>
</select>
</td>
<td>
<font size="3"><input type="text" name="Year" size="11"></font>
        </td>
</tr>
</table>

</font><input type="button" value="Submit" name="B1" onClick="validate();" >
</form>
<%
}
else
{
        response.sendRedirect("Unauthorised.htm");
}%>
</body>
</html>
```

NewInter_Cart.jsp

```
<%@ page errorPage="errorpage.jsp" language="java" %>
<%
        int i;
        String bookid=null;
        String user_src=null;
        user_src=(String)session.getValue("user");
        if (user_src!=null)
```

```
	{
		int counter=0;
		int total_sel=Integer.parseInt((String)session.getValue("TotalSel"));
		counter=Integer.parseInt((String)session.getValue("ctr_val"));
		int ctr=total_sel;
		for(i=1;i<=counter;i++)
		{
			if(request.getParameter("chk"+i)!=null)
			{
				bookid=request.getParameter("chk"+i);
				ctr=ctr+1;
				session.putValue("chk_var"+ctr,bookid);
			}
		}
		session.putValue("TotalSel",String.valueOf(ctr));
		response.sendRedirect("Cart.jsp");
	}
	else
	{
		out.println("Unauthorised Shopping is not allowed");
	}
%>
```

Logout.jsp

```
<html>
<head>
<title>Logged out</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        </CENTER>

        <%@  page errorPage="errorpage.jsp" language="java" %>
        <%
        session.invalidate();
        %>
      <center>
        <BR><BR><BR><BR><b>You have logged out successfully.
      <BR><BR>Return to Shopping Cart? <a href="Login.htm">Click here</a></b>
      </center>
</body>
</html>
```

Final.jsp

```
<html>
<head>
<title>Order placed</title>
</head>
<body bgcolor="#CCFFFF">
        <CENTER>
        <H1> <u>Online Shopping Cart</u> </H1>
        <H2>Order Details</H2></CENTER>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
        <%@  page errorPage="errorpage.jsp" language="java"
                import="java.sql.*,java.util.*,java.text.*"  %>
<%!
int OrderID,price;
%>
<%
        String user_src=(String)session.getValue("user");
        if (user_src!= null)
        {
                try{
                Connection conn;
                conn=null;
                ResultSet rs=null;
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                conn =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");

                PreparedStatement stat1=null,stat_sel=null,stat_ins=null;
```

```
OrderID=(int)(10000*Math.random()+1);
price=Integer.parseInt((String)session.getValue("bPrice"));
int counter=0;
counter=Integer.parseInt((String)session.getValue("TotalSel"));
int[] bookid=new int[counter];
int[] bk_quantity=new int[counter];
int org_bkqty=0;
int i=0;
String ins_query="";
String sel_qtyQuery="";
String ins_qty="";
for(i=1;i<=counter;i++)
{
	ResultSet rs_sql=null;
	if((String)session.getValue("chk_var"+i)!=null)
	{

bookid[i-1]=Integer.parseInt((String)session.getValue("chk_var"+i));
	}

	if((String)session.getValue("bookqty"+i)!=null)
	{

bk_quantity[i-1]=Integer.parseInt((String)session.getValue("bookqty"+i));
	}
	ins_query="INSERT INTO ORDER_DETAILS VALUES(?,?,?)";
	stat1=conn.prepareStatement(ins_query);
	stat1.setInt(1,OrderID);
	stat1.setInt(2,bookid[i-1]);
	stat1.setInt(3,bk_quantity[i-1]);
	int rs_int=0;
	rs_int=stat1.executeUpdate();
	sel_qtyQuery="SELECT QUANTITY FROM BOOK_DETAILS
WHERE BOOKID = ?";
	stat_sel = conn.prepareStatement(sel_qtyQuery);
	stat_sel.setInt(1,bookid[i-1]);
	rs_sql=stat_sel.executeQuery();
	if(rs_sql.next())
	{
		org_bkqty=rs_sql.getInt(1);
	}
	rs_sql.close();
	String ns_qty="UPDATE BOOK_DETAILS SET QUANTITY=?
WHERE BOOKID=?";
	stat_ins=conn.prepareStatement(ns_qty);
	stat_ins.setInt(1,org_bkqty-bk_quantity[i-1]);
	stat_ins.setInt(2,bookid[i-1]);
	rs_int=stat_ins.executeUpdate();
```

```
                }
                }catch(Exception e){}
        %>
        <form name="newsrc"  method="POST">
        <BR><font size="3"><b>
        Your order has been successfully placed.
        <BR><BR>Order Number is : <%=OrderID%>
        <BR><BR>Total Amount is : <%=price%>
        <BR><BR>Date of Order is:
        <%

        out.println((new java.util.Date()).toLocaleString());
        %>
        <BR><BR>
      Your Order will be delivered within next 48 working hours
<%
    String card_no=request.getParameter("Card");
    Connection con= null;
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con =DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
    java.util.Date now = new java.util.Date();
    DateFormat df1 = DateFormat.getDateInstance(DateFormat.SHORT);
  String s1 = df1.format(now);
    try
   {
  String str = "insert into order_table values(?,?,?,?)";
    PreparedStatement stat= con.prepareStatement(str);
    stat.setInt(1,OrderID);
    stat.setString(2,user_src);
    stat.setInt(3,price);
    stat.setString(4,s1);
    int x = stat.executeUpdate();
  }catch(Exception e){out.println(e); }
    }
    else
    {
        response.sendRedirect("Unauthorised.htm");
    }
    %>
</body>
</html>
```

Administrator Modules

Admin.htm

```
<HTML>
<HEAD>
<TITLE>Administering Shopping Cart</TITLE>
</HEAD>
```

```
<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1>Administering Online Shopping Cart </H1>
	</CENTER>
	<br><hr><br><br>
	<h2>Welcome Administrator</h2><br>
	<Form name="admin" method="POST">
	<TABLE width=100% border="0" cellpadding="2" cellspacing="0">
	<TR>
		<th align = "left" width="4%"></th>
		<th align = "left" width="20%"></th>
		<th align = "left" width="38%"></th>
		<th align = "left" width="38%"></th>
	</TR>
	<TR>
		<TH>1</TH>
		<TH align = "left">Book</TH>
		<TH><a href="Insert.jsp">Insert New Book Details</a></TH>
		<TH><a href="Modify.jsp">Modify Book Details</a></TH>
	</TR>
	<TR>
		<TH>2</TH>
		<TH align = "left">Category</TH>
		<TH><a href="Insert_Category.jsp">Insert Book Category</a></TH>
		<TH><a href="Update_Category.jsp">Modify Book Category</a></TH>
	</TR>
</BODY>
</HTML>
```

Insert.jsp

```
<HTML>
<HEAD>
<TITLE>Insert New Book</TITLE>
</HEAD>
<BODY>
	<CENTER>
	<H1> <u>Administering Online Shopping Cart</u> </H1>
	<H2> Add Book</H2>
	</CENTER>
	<BR>
	<HR>
	<BR>
	<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
	<BODY bgcolor="#CCFFFF">
	<SCRIPT language="Javascript">
	function check()
	{
	var ctr=0;
	if(document.addbook.title.value=="")
	{
		alert("Please Enter Title");
		return;
	}
	if(document.addbook.author.value=="")
	{
		alert("Please Enter Author");
		return;
	}
	if(document.addbook.pub.value=="")
	{
		alert("Please Enter Publisher Name");
		return;
	}
	if(document.addbook.edition.value=="")
	{
		alert("Please Enter Edition");
		return;
	}
	if(document.addbook.price.value=="")
	{
		alert("Please Enter Price (in Rs.)");
		return;
	}

	if(document.addbook.qty.value=="")
	{
```

```
        alert("Please Enter Quantity");
        return;
}
if(document.addbook.desc.value=="")
{
        alert("Please Enter Description");
        return;
}
if(isNaN(document.addbook.price.value))
{
        alert("Please Enter a numeric value for Price (in Rs.)");
        ctr=1;
}
if(isNaN(document.addbook.qty.value))
{
        alert("Please Enter a numeric value for Quantity");
        ctr=1;
}
if(ctr==0)
{
        document.forms[0].action="AddBook.jsp";
        document.forms[0].submit();
}
else
{
        return true;
};

}

</SCRIPT>
<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>
<%
Connection con = null;
//Statement stat = null;
PreparedStatement stat = null;
//Statement stat1 = null;
PreparedStatement stat1 = null;
ResultSet rs = null;
ResultSet rs1 = null;
String query=null;
String query1=null;
int ctr=0, flag=0;
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
	con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
	query="Select a.*,b.categoryname from book_details a, category_details b where
a.categoryid=b.categoryid";
	query1="select * from category_details";
	stat = con.prepareStatement(query);
	stat1 = con.prepareStatement(query1);

	rs=stat.executeQuery();

	rs1=stat1.executeQuery();

	%>
	<br><h3>List of Available Books:</h3>
	<table border=1>
	<tr>
		<td>S.No</td>
		<td>Book Id</td>
		<td>Book Name</td>
		<td>Author Name</td>
		<td>Publisher</td>
		<td>Edition</td>
		<td>Price (in Rs.)</td>
		<td>Quantity Available(Nos.)</td>
		<td>Description</td>
		<td>Category</td>
	</tr>
	<%
	int int_BOOKID,int_CATEGORYID,int_PRICE,int_QUANTITY;
	String
str_TITLE,str_AUTHOR,str_PUBLISHER,str_EDITION,str_DESCRIPTION;
	while(rs.next())
	{
		ctr=ctr+1;
		out.println("<tr><td>"+ctr);
		int_BOOKID=rs.getInt(1);
		out.println("<td>");
		out.println(int_BOOKID);

		int_CATEGORYID=rs.getInt(2);

		str_TITLE=rs.getString(3);
		out.println("<td>");
		out.println(str_TITLE);

		str_AUTHOR=rs.getString(4);
		out.println("<td>");
		out.println(str_AUTHOR);
```

```
                str_PUBLISHER=rs.getString(5);
                out.println("<td>");
                out.println(str_PUBLISHER);

                str_EDITION=rs.getString(6);
                out.println("<td>");
                out.println(str_EDITION);

                int_PRICE=rs.getInt(7);
                out.println("<td>");
                out.println(int_PRICE);

                int_QUANTITY=rs.getInt(8);
                out.println("<td>");
                out.println(int_QUANTITY);

                str_DESCRIPTION=rs.getString(9);
                out.println("<td>");
                out.println(str_DESCRIPTION);

                out.println("<td>");
                out.println(rs.getString(10));

                out.println("</tr>");
        }
        out.println("</table>");%>
        <br>
        <hr>
        <h3> Add New Book Details</h3>
        <form name="addbook">
        <table border=1>
        <tr><td>Book Title</td><td><input type="text" name="title"></td>
        <tr><td>Author Name</td><td><input type="text" name="author"></td>
        <tr><td>Publisher</td><td><input type="text" name="pub"></td>
        <tr><td>Edition</td><td><input type="text" name="edition"></td>
        <tr><td>Price (in Rs.)</td><td><input type="text" name="price"></td>
        <tr><td>Quantity Available(Nos.)</td><td><input type="text"
name="qty"></td>
        <tr><td>Description</td><td><input type="text" name="desc"></td>
        <tr><td>Category</td><td><select size="1" name="cat">
        <%
                while(rs1.next())
                {%>
                <option
value="<%=rs1.getString(1)%>"><%=rs1.getString(2)%></option>
                <%}%>
                </select></td>
                </tr>
```

```
                <tr><td><input type="submit" value="   Add    "
onClick="check();"></td>
                <td><input type="Reset" value="    Clear    "></td>
                </table>
                </form>
        <%
        }
        catch(Exception e)
        {
        out.print("Error = " + e + "<HR>");
        }
        %>
</BODY>
</HTML>
```

Modify.jsp

```
<HTML>
<HEAD>
<TITLE>Modify Book Details</TITLE>
</HEAD>

<BODY bgcolor="#CCFFFF">

        <CENTER>
        <H1> Administering Online Shopping Cart </H1>
        <H2>Modify Book Details</H2>
        </CENTER>
        <BR>
        <HR>
        <BR>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>

        <%@ page errorPage="errorpage.jsp" import="java.net.*" %>
        <%@ page import="java.io.*" %>
        <%@ page import="java.sql.*" %>
        <%
                Connection con = null;

                PreparedStatement stat = null;

                ResultSet rs = null;
                String query=null;
                int ctr=0;
                try
                {
```

```
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


		query="Select a.*,b.categoryname from book_details a, category_details b
where a.categoryid=b.categoryid";
		stat = con.prepareStatement(query);

		rs=stat.executeQuery();
	%>
		<br><h3>List of Available Books:</h3>
		<form name="Mod" action="Modify_Book.jsp">
		<table border=1>
		<tr>
		<td>S.No</td>
		<td>Book Id</td>
		<td>Book Name</td>
		<td>Author Name</td>
		<td>Publisher</td>
		<td>Edition</td>
		<td>Price (in Rs.)</td>
		<td>Quantity Available(Nos.)</td>
		<td>Description</td>
		<td>Category</td>
		</tr>
		<%
		int int_BOOKID,int_CATEGORYID,int_PRICE,int_QUANTITY;
		String
str_TITLE,str_AUTHOR,str_PUBLISHER,str_EDITION,str_DESCRIPTION;
		while(rs.next())
		{
			ctr=ctr+1;
			out.println("<tr><td>"+ctr);
			int_BOOKID=rs.getInt(1);
			out.println("<td>");
			out.println("<input type=radio name=R1
value="+int_BOOKID+">");

			int_CATEGORYID=rs.getInt(2);

			str_TITLE=rs.getString(3);
			out.println("<td>");
			out.println(str_TITLE);

			str_AUTHOR=rs.getString(4);
			out.println("<td>");
			out.println(str_AUTHOR);
```

```
                    str_PUBLISHER=rs.getString(5);
                    out.println("<td>");
                    out.println(str_PUBLISHER);

                    str_EDITION=rs.getString(6);
                    out.println("<td>");
                    out.println(str_EDITION);

                    int_PRICE=rs.getInt(7);
                    out.println("<td>");
                    out.println(int_PRICE);

                    int_QUANTITY=rs.getInt(8);
                    out.println("<td>");
                    out.println(int_QUANTITY);

                    str_DESCRIPTION=rs.getString(9);
                    out.println("<td>");
                    out.println(str_DESCRIPTION);

                    out.println("<td>");
                    out.println(rs.getString(10));

                    out.println("</tr>");
                }
        %>
                <tr>
                    <td><input type="submit" value="   Modify    "></td>
                    <td><input type="Reset" value="   Clear    "></td>
                </tr>
                </table>
                </form>
        <%
                }
                catch(Exception e)
                {
                out.print("Error = " + e + "<HR>");
        }
        %>
</BODY>
</HTML>
```

AddBook.jsp

```
<HTML>
<HEAD>
<TITLE>Insert New Book</TITLE>
</HEAD>
```

```
<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1> Administering Online Shopping Cart </H1>
	<H2>Add Book</H2>
	</CENTER>
	<BR>
	<HR>

	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
	Connection con = null;

	PreparedStatement stat = null;

	Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
	con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


	ResultSet rs1 = null;

	int rs;
	String query=null;
	String query1=null;
	String TITLE;
	TITLE=request.getParameter("title");
	query1="Select a.* from book_details a where a.title=?";
	PreparedStatement stat1= con.prepareStatement(query1);
	stat1.setString(1,TITLE);
	rs1=stat1.executeQuery();

	if(rs1.next())
	{
	%>
		<SCRIPT language="JavaScript">
		alert("This online shopping system project book already exists.");
		location.href="Insert.jsp";
		</SCRIPT>
	<%
	}
	else
	{
		int int_BOOKID,int_CATEGORYID,int_PRICE,int_QUANTITY;
		String
str_TITLE,str_AUTHOR,str_PUBLISHER,str_EDITION,str_DESCRIPTION;

		int_BOOKID=(int)(10000*Math.random()+1);
```

```
            int_CATEGORYID=Integer.parseInt(request.getParameter("cat"));
            str_TITLE=request.getParameter("title");
            str_AUTHOR=request.getParameter("author");
            str_PUBLISHER=request.getParameter("pub");
            str_EDITION=request.getParameter("edition");
            int_PRICE=Integer.parseInt(request.getParameter("price"));
            int_QUANTITY=Integer.parseInt(request.getParameter("qty"));
            str_DESCRIPTION=request.getParameter("desc");


            query="insert into book_details values (?,?,?,?,?,?,?,?,?)";
            stat = con.prepareStatement(query);
            stat.setInt(1,int_BOOKID);
            stat.setInt(2,int_CATEGORYID);
            stat.setString(3,str_TITLE);
            stat.setString(4,str_AUTHOR);
            stat.setString(5,str_PUBLISHER);
            stat.setString(6,str_EDITION);
            stat.setInt(7,int_PRICE);
            stat.setInt(8,int_QUANTITY);
            stat.setString(9,str_DESCRIPTION);

            rs=stat.executeUpdate();
            if(rs==1)
            {%>
            <BR>
            <H3>Successfully inserted the data.</H3>
            <% }
            else
            {%>
                    <BR>
                    <H3>Could not add the book</H3>
            <%}
        }%>

        <BR>
        <a href="Logout.jsp">Click here to logout</a>
        <BR><BR>
        <a href="Admin.htm">Click here to go to administrator page</a>
        <BR><BR>
        <a href="Insert.jsp"> Click here to add more books</a>

</BODY>
</HTML>
```

Modify_Book.jsp

```
<HTML>
<HEAD>
<TITLE>Modify Book</TITLE>
</HEAD>
<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1> Administering Online Shopping Cart </H1>
	<H2>Modify Book Details</H2>
	</CENTER>
	<HR>
	<BR>
	<BR>
	<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>

	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
		Connection con = null;

		PreparedStatement stat = null;
		ResultSet rs = null;
		String query=null;
		int book_sel = Integer.parseInt(request.getParameter("R1"));
		try
		{
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");
		query="Select a.*,b.categoryname from book_details a, category_details b
where a.bookid=? and a.categoryid=b.categoryid";

		stat = con.prepareStatement(query);
		stat.setInt(1,book_sel);
		rs=stat.executeQuery();
		if(rs.next())
		{
	%>
		<br><h3>Details of Selected Book:</h3>
		<form name="Mod" action="Final_Mod_Book.jsp">
		<input type="hidden" name="categoryid"
value="<%=rs.getString(2)%>">
		<table border=1>
		<tr>
		<td>Book Id</td><td><input type="hidden" name="bookid"
value="<%=book_sel%>"><%=book_sel%> </td></tr>
		<tr>
```

```
                <td>Book Name</td><td><input type="text" name="btitle"
value=<%=rs.getString(3)%>></td>
                </tr><tr>
                <td>Author Name</td><td><input type="text" name="bauthor"
value=<%=rs.getString(4)%>></td>
                </tr><tr>
                <td>Publisher</td><td><input type="text" name="bpub"
value=<%=rs.getString(5)%>></td>
                </tr><tr>
                <td>Edition</td><td><input type="text" name="bedition"
value=<%=rs.getString(6)%>></td>
                </tr><tr>
                <td>Price (in Rs.)</td><td><input type="text" name="bprice"
value=<%=rs.getString(7)%>></td>
                </tr><tr>
                <td>Quantity Available(Nos.)</td><td><input type="text" name="bqty"
value=<%=rs.getString(8)%>></td>
                </tr><tr>
                <td>Description</td><td><input type="text" name="bdesc"
value=<%=rs.getString(9)%>></td>
                </tr><tr>
                <td>Category</td><td><input type="text" name="bcat"
value=<%=rs.getString(10)%>></td>
                </tr><tr>
                <tr><td><input type="submit" value=" Save Changes "></td><td><input
type="Reset" value="    Clear    "></td></tr>
                </table>
                </form>
                <%
                }
        }
                catch(Exception e)
                {
                        out.print("Error = " + e + "<HR>");
                }
        %>
</BODY>
</HTML>
```

Final_mod_Book.jsp

```
<HTML>
<HEAD>
<TITLE>Book Details Modified</TITLE>
</HEAD>

<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1> Administering Online Shopping Cart </H1>
	<H2>Book Details Modified</H2>
	</CENTER>
	<BR>
	<HR>
	<BR>
	<BR>
	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
		Connection con = null;
		//Statement stat = null;
		PreparedStatement stat = null;
		//Statement stat1 = null;
		PreparedStatement stat1 = null;
		ResultSet rs = null;

		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


		int rs_int;
		int int_BOOKID,int_CATEGORYID,int_PRICE,int_QUANTITY;

		String query=null;
		String
str_TITLE,str_AUTHOR,str_PUBLISHER,str_EDITION,str_DESCRIPTION;

		int_BOOKID=Integer.parseInt(request.getParameter("bookid"));
		int_CATEGORYID=Integer.parseInt(request.getParameter("categoryid"));
		str_TITLE=request.getParameter("btitle");
		str_AUTHOR=request.getParameter("bauthor");
		str_PUBLISHER=request.getParameter("bpub");
		str_EDITION=request.getParameter("bedition");
		int_PRICE=Integer.parseInt(request.getParameter("bprice"));
		int_QUANTITY=Integer.parseInt(request.getParameter("bqty"));
		str_DESCRIPTION=request.getParameter("bdesc");
```

```
                query="update book_details set
categoryid=?,title=?,author=?,publisher=?,edition=?,price=?,quantity=?,description=?
where bookid=?";
                stat = con.prepareStatement(query);
                stat.setInt(1,int_CATEGORYID);
                stat.setString(2,str_TITLE);
                stat.setString(3,str_AUTHOR);
                stat.setString(4,str_PUBLISHER);
                stat.setString(5,str_EDITION);
                stat.setInt(6,int_PRICE);
                stat.setInt(7,int_QUANTITY);
                stat.setString(8,str_DESCRIPTION);
                stat.setInt(9,int_BOOKID);

            rs_int=stat.executeUpdate();

                if (rs_int==1)
                {
                        out.println("Book details for Book Id "+int_BOOKID+"
successfully modified.");
                }
                else
                {
                        out.println("Sorry Transaction not Completed");
                }
        %>
        <BR><BR>
        <a href="Logout.jsp">Click here to logout</a>
        <BR><BR>
        <a href="Admin.htm">Click here to go to administrator page</a>
        <BR><BR>
        <a href="Modify.jsp"> Click here to modify more books</a>
</BODY>
</HTML>
```

Insert_Category.jsp

```
<HTML>
<HEAD>
<TITLE>Insert New Category</TITLE>
</HEAD>
<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1> Administering Online Shopping Cart </H1>
	<H2> Add Category</H2>
	</CENTER>
	<BR>
	<HR>
	<BR>
	<BR>
	<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>

	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
		Connection con = null;

		PreparedStatement stat = null;
		ResultSet rs = null;
		String query=null;
		int ctr=0, flag=0;
		try
		{
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


		query="select * from category_details";
		stat = con.prepareStatement(query);

		rs=stat.executeQuery();
		%>
		<br><h3>Add New Category:</h3>
		<form name="addbook" action="AddCategory.jsp">
		<table border=1>
		<tr>
		<td>Categories Available</td>
		<td><select size="1" name="cat">
			<%
			while(rs.next())
			{%>
```

```
                        <option
value="<%=rs.getString(1)%>"><%=rs.getString(2)%></option>
                        <%}%>
                        </select></td>
                </tr>
                <tr>
                <td>Category to be added</td>
                <td><input type="text" name="cat_name"></td></tr>
                <tr><td><input type="submit" value="   Add    "></td><td><input
type="Reset" value="   Clear    "></td>
                </table></form>
        <%
                }
                catch(Exception e)
                {
                out.print("Error = " + e + "<HR>");
                }
                finally
                {
                rs.close();
                con.close();
                }
        %>
</BODY>
</HTML>
```

Update_Category.jsp

```
<HTML>
<HEAD>
<TITLE>Category Modified</TITLE>
</HEAD>
<BODY bgcolor="#CCFFFF">
        <CENTER>
        <H1> Administering Online Shopping Cart </H1>
        <H2>Category Modified</H2>
        </CENTER>
        <BR>
        <HR>
        <BR>
        <a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
        <BR><BR>

        <script language="Javascript">
        function sel_click()
        {
                var f = document.modcat;
                var i=f.cat.selectedIndex;
                var str=f.cat.options[i].text;
```

```
		document.modcat.vr_cat.value=str;
		document.modcat.cat_val.value=document.modcat.elements[0].value;
	}
	</script>

	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
		Connection con = null;

		PreparedStatement stat = null;
		ResultSet rs = null;
		String query=null;
		try
		{
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");

		query="select * from category_details";
		stat = con.prepareStatement(query);

		rs=stat.executeQuery();
	%>
		<br><h2>Modify Category:--</h2>
		<form name="modcat" action="ModifyCategory.jsp">
		<table border=1>
		<tr>
		<td>Category</td>
		<td><select size="1" name="cat" onChange="sel_click();">
			<%
			while(rs.next())
			{%>
			<option value="<%=rs.getString(1)%>">
			<%=rs.getString(2)%></option>
			<%}%>
			</select></td>
		</tr>
		<tr><td colspan=2><input type="text" name="vr_cat"><input type="text"
name="cat_val"></td>
		<tr><td><input type="submit" value="   Modify  "></td><td><input
type="Reset" value="   Clear    "></td>
		</table></form>
	<%
		}
		catch(Exception e)
		{
		//	out.print("Error = " + e + "<HR>");
```

```
		}
		finally
		{
			rs.close();
			con.close();
		}
	%>
</BODY>
</HTML>
```

ModifyCategory.jsp

```
<HTML>
<HEAD>
<TITLE>Modify Category</TITLE>
</HEAD>
<BODY bgcolor="#CCFFFF">
	<CENTER>
	<H1> Administering Online Shopping Cart </H1>
	</CENTER>
	<BR>
	<HR>
	<BR>
	<BR>
	<%@ page errorPage="errorpage.jsp" import="java.net.*" %>
	<%@ page import="java.io.*" %>
	<%@ page import="java.sql.*" %>
	<%
		Connection con = null;
		PreparedStatement stat = null;
		int rs = 0;
		String query=null;
		try
		{
		String new_cat="";
		int cat_id=0;
				new_cat=request.getParameter("vr_cat");
		cat_id=Integer.parseInt(request.getParameter("cat_val"));

		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


		query="UPDATE category_details SET CATEGORYNAME=? WHERE
CATEGORYID=?";

		stat = con.prepareStatement(query);
		stat.setString(1,new_cat);
		stat.setInt(2,cat_id);
		rs=stat.executeUpdate();
```

```
                if (rs==1)
                {
                        out.println("Category Successfully Modified");
                }
                else
                {
                        out.println("Transaction Not Allowed");
                }
                }
                catch(Exception e)
                {
                        out.print("Error = " + e + "<HR>");
                }
                finally
                {
                        con.close();
                }
        %>
                <BR>
                <BR>
                <a href="Logout.jsp">Click here to logout</a>
                <BR><BR>
                <a href="Admin.htm">Click here to go to administrator page</a>
                <BR><BR>
                <a href="Update_Category.jsp"> Click here to modify more
categories</a>
</BODY>
</HTML>
```

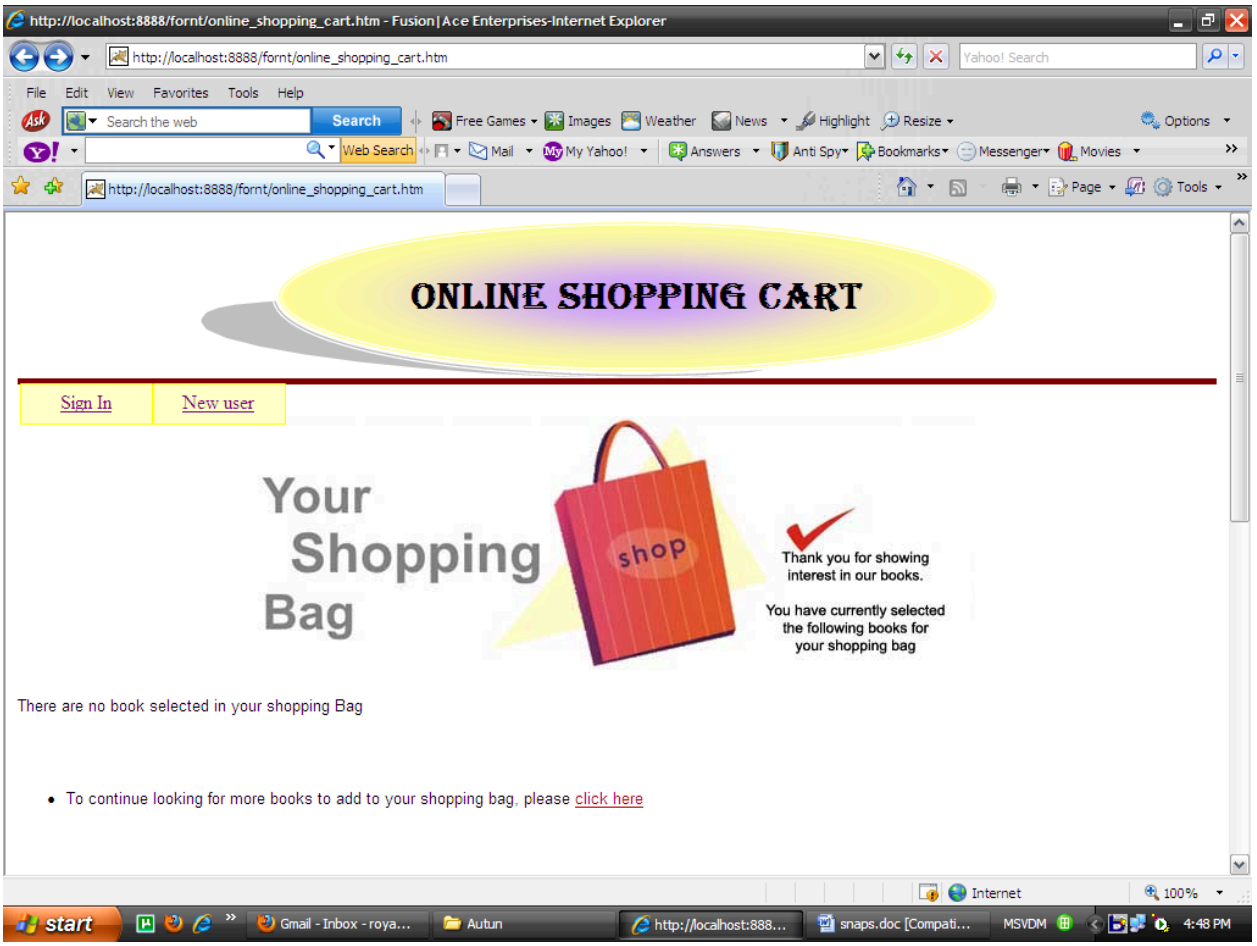
AddCategory.jsp

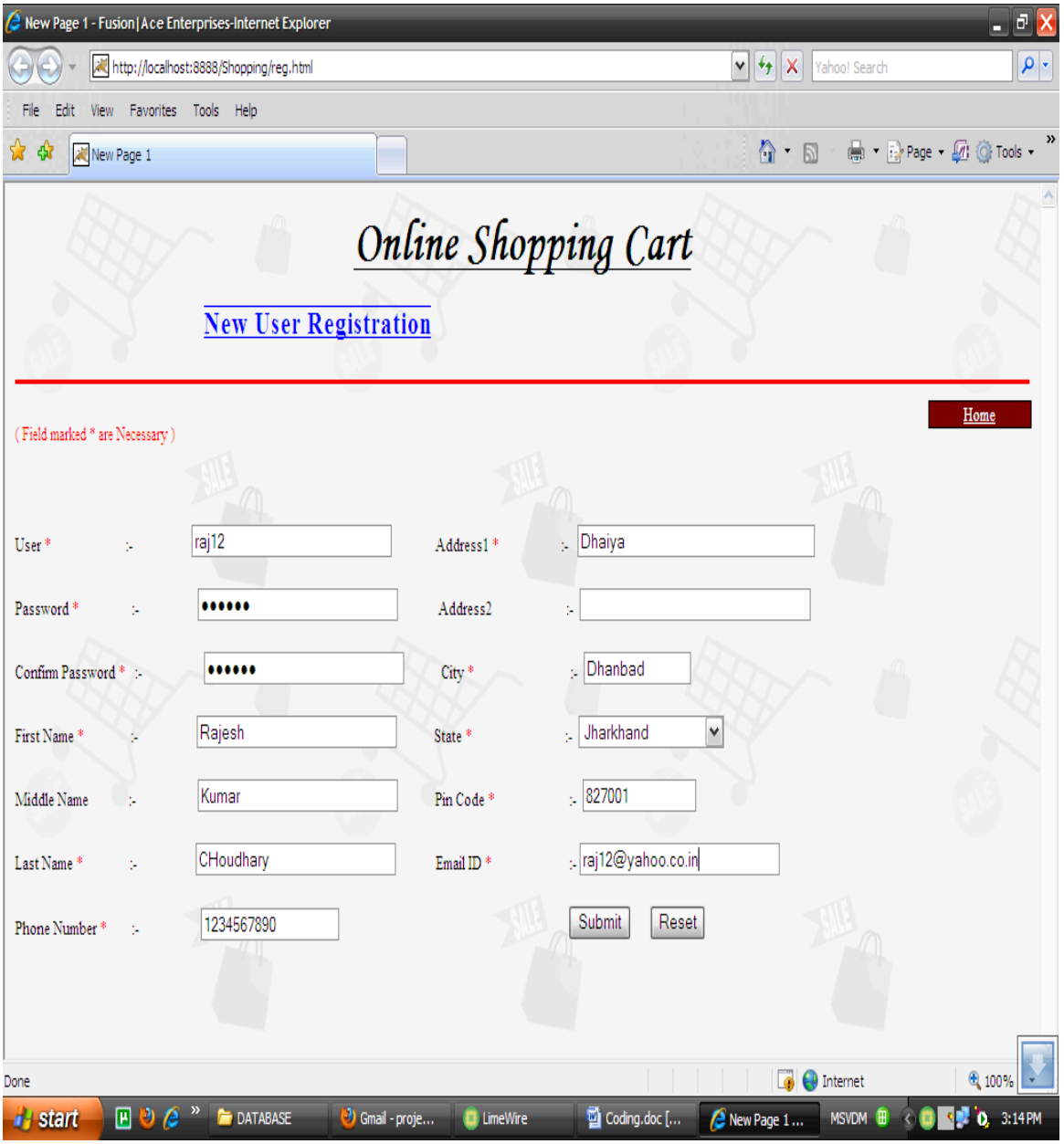
```
<HTML>

<HEAD>
<TITLE>Insert New Category</TITLE>
</HEAD>

<BODY bgcolor="#CCFFFF">
        <CENTER>
        <H1> Administering Online Shopping Cart </H1>
        <H2> Add Category</H2>
        </CENTER>
        <BR>
        <HR>
        <BR>

        <%@ page errorPage="errorpage.jsp" import="java.net.*" %>
        <%@ page import="java.io.*" %>
        <%@ page import="java.sql.*" %>
        <%
```

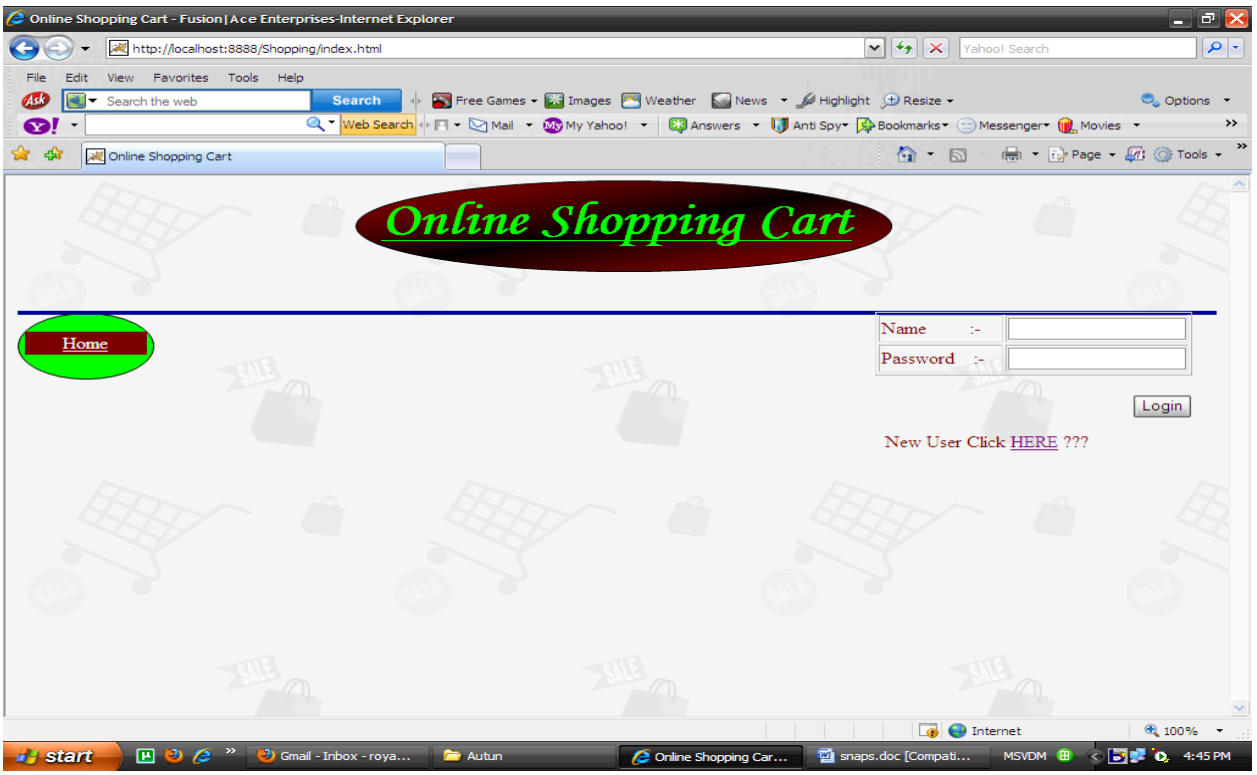
```
		Connection con = null;

		PreparedStatement stat = null;
		Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
		con = DriverManager.getConnection("jdbc:odbc:Data","scott","tiger");


		int rs;
		int int_CATEGORYID;

		String query=null;
		String str_CATEGORY;

		int_CATEGORYID=(int)(10000*Math.random()+1);
		str_CATEGORY=request.getParameter("cat_name");


		query="insert into CATEGORY_details values (?,?)";
		stat = con.prepareStatement(query);
		stat.setInt(1,int_CATEGORYID);
		stat.setString(2,str_CATEGORY);

		rs=stat.executeUpdate();
		if (rs==1)
		{
			%><H3>New Category Successfully Added</H3><%
		}
		else
		{
			%><H3>Transaction Not Allowed</H3><%
		}
	%>
	<BR>
	<a href="Logout.jsp" ><font size="4"  >Click here to Logout</font></a>
	<BR><BR>
	<a href="Admin.htm"><font size="4"  >Click here to go to administrator
page</font></a>
	<BR><BR>
	<a href="Insert_Category.jsp"><font size="4"  > Click here to add more
categories</font></a>
	<BR><BR>
</BODY>
</HTML>
```

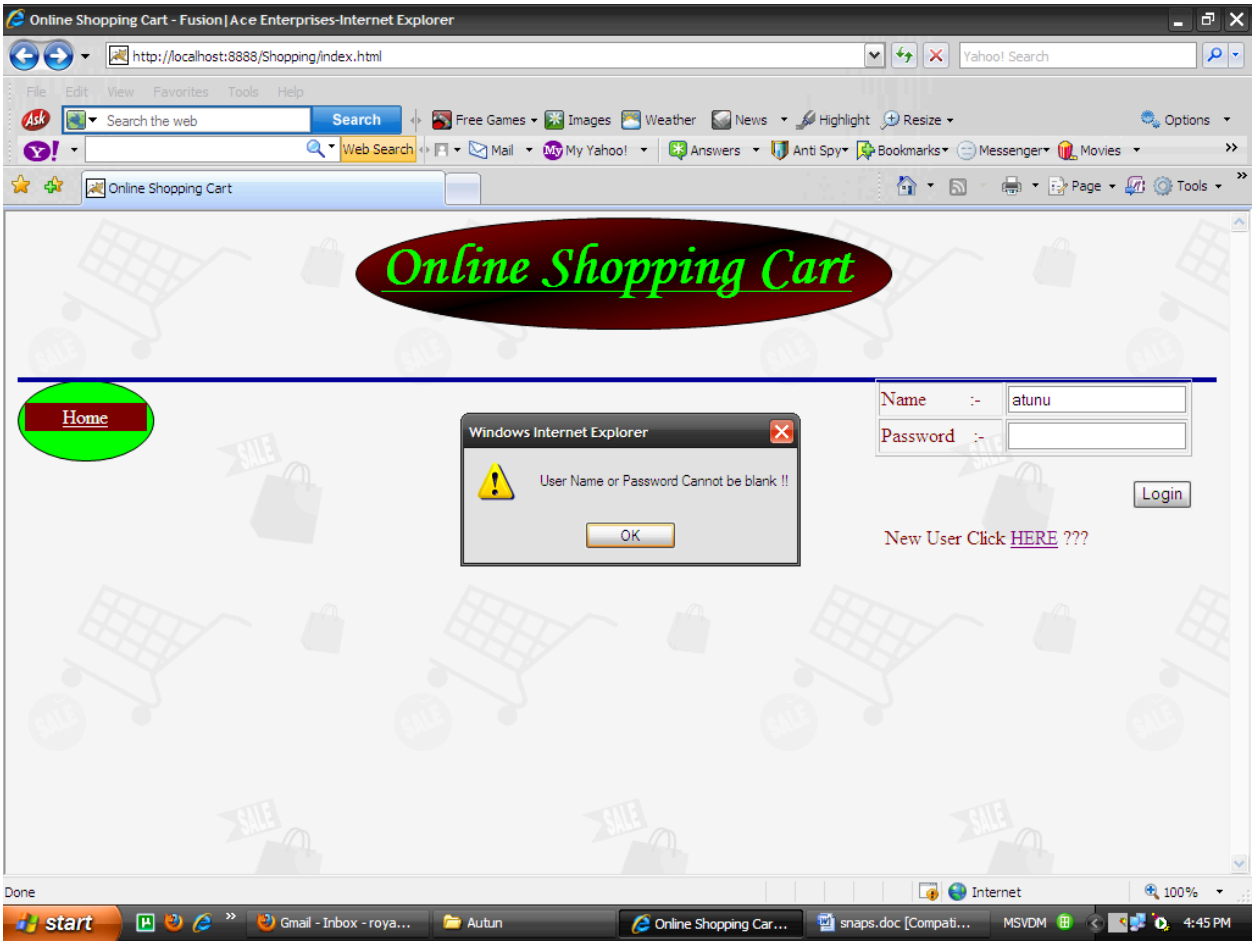
End of Coding

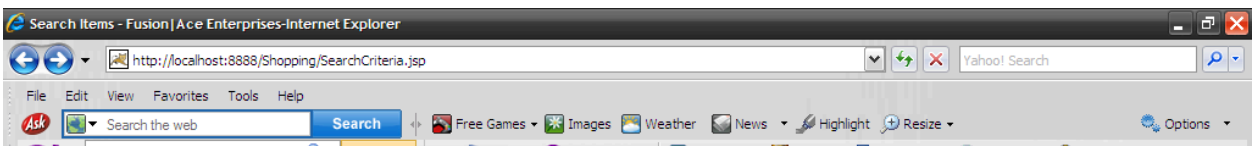
User Interface

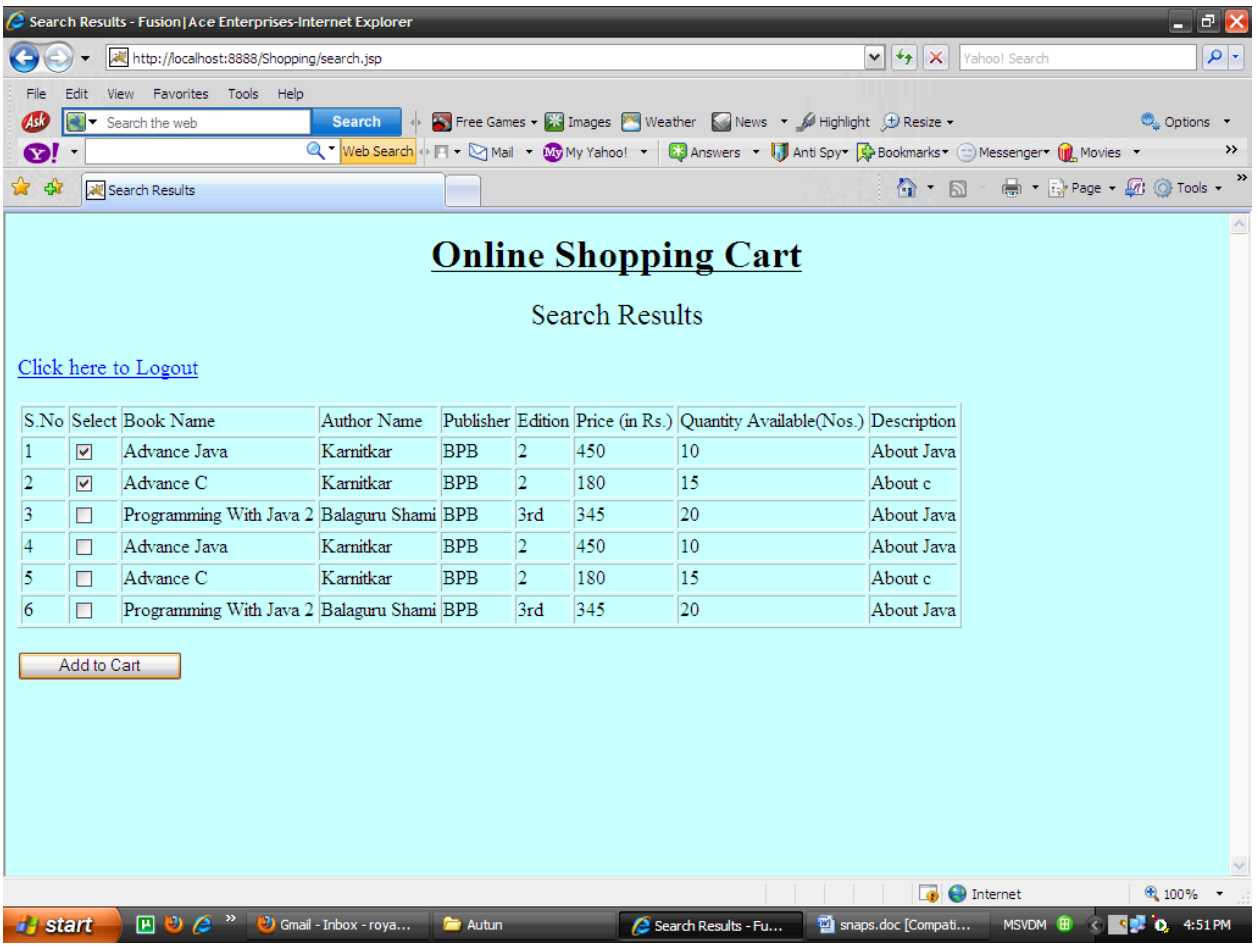



New Page 1 - Fusion | Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/reg.html
File Edit View Favorites Tools Help
New Page 1
Online Shopping Cart
New User Registration
Home
(Field marked * are Necessary)
User * :- raj12
Password * :- ••••••
Confirm Password * :- ••••••
First Name * :- Rajesh
Middle Name :- Kumar
Last Name * :- CHoudhary
Phone Number * :- 1234567890
Address1 * :- Dhaiya
Address2 :-
City * :- Dhanbad
State * :- Jharkhand
Pin Code * :- 827001
Email ID * :- raj12@yahoo.co.in
Submit
Reset
Done
Internet
100%
start
DATABASE
Gmail - proje...
LimeWire
Coding.doc [...
New Page 1 ...
MSVDM
3:14 PM

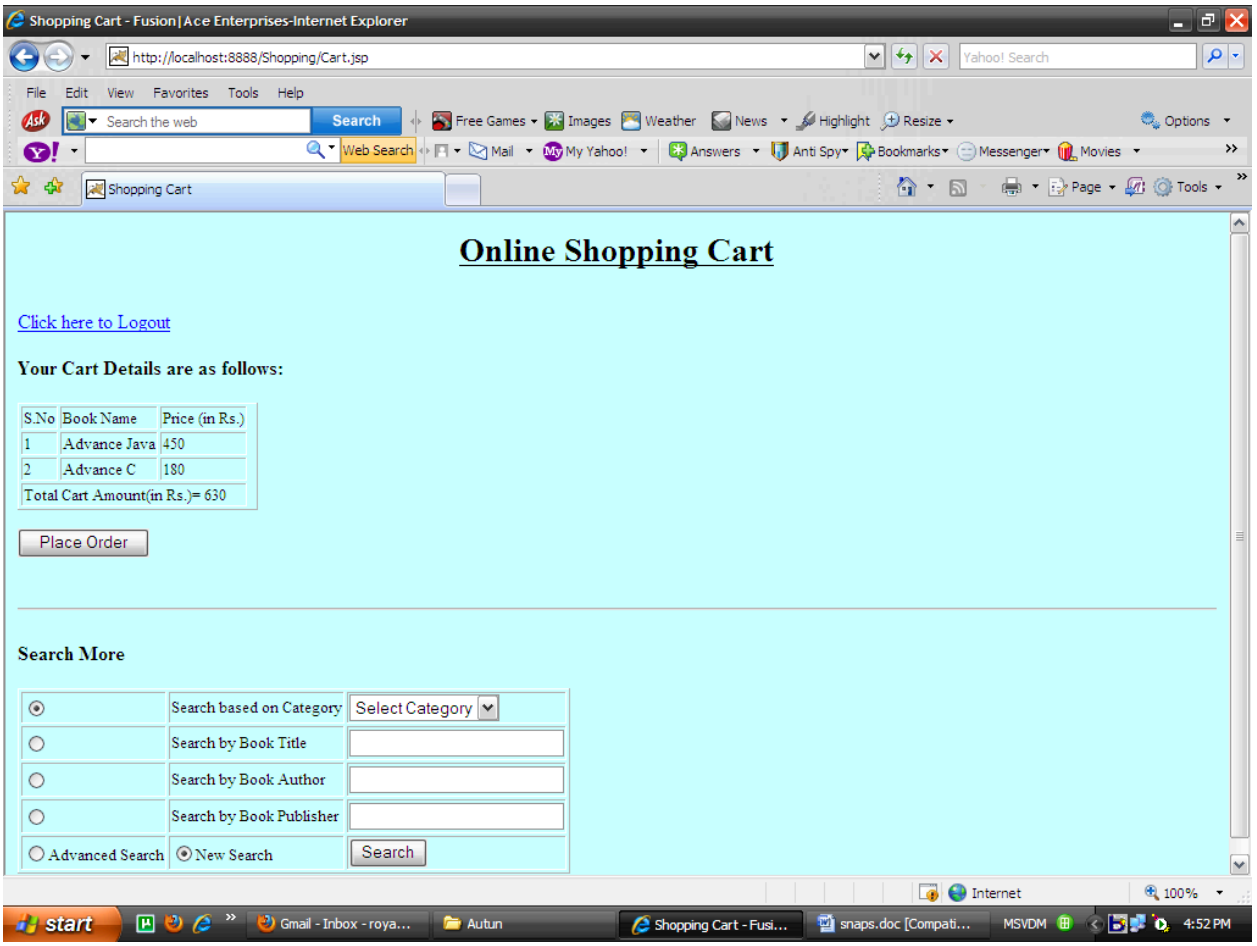

Online Shopping Cart - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/index.html
Online Shopping Cart
Home
Name :-
Password :-
Login
New User Click HERE ???

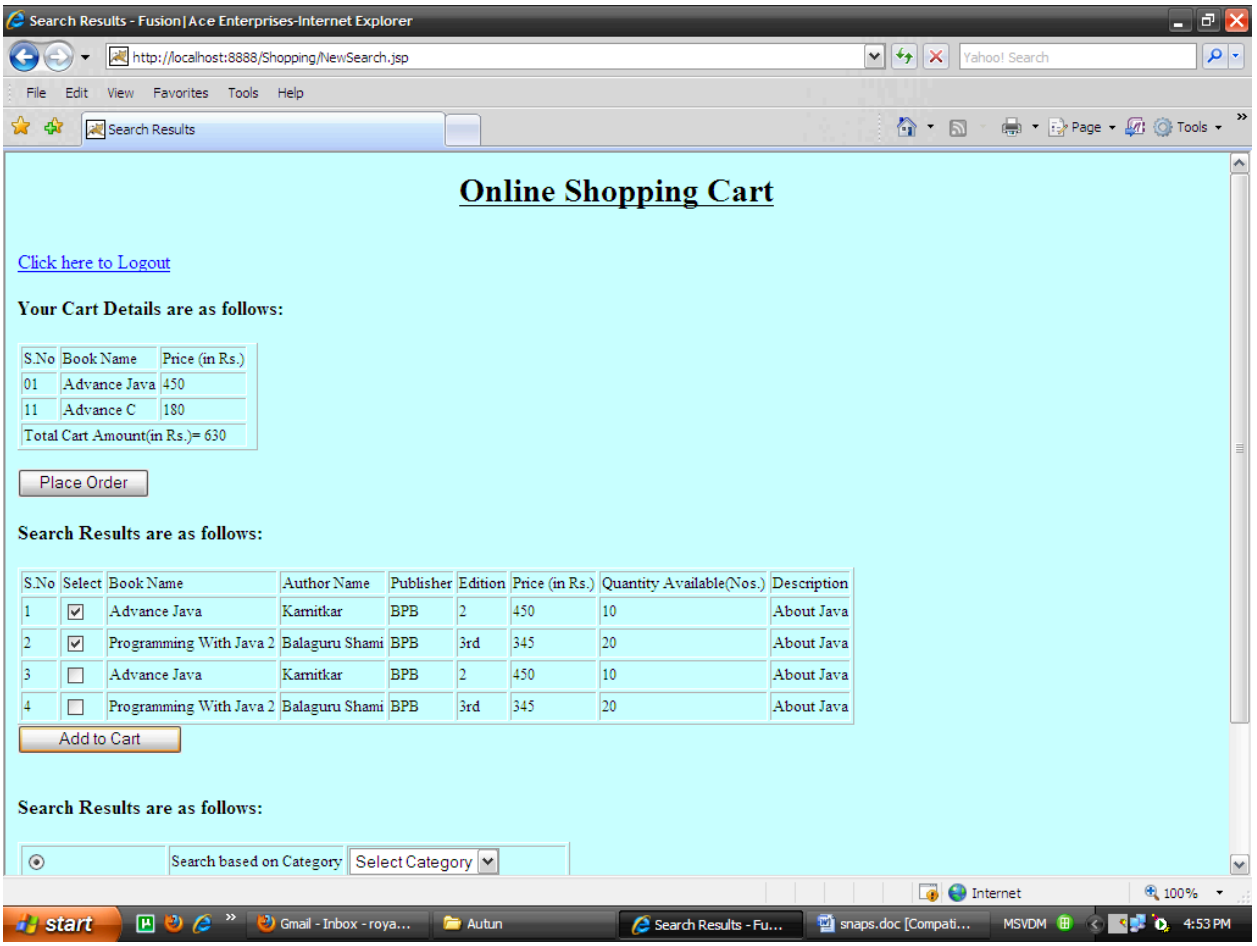

Online Shopping Cart - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/index.html
Online Shopping Cart
Home
Windows Internet Explorer
User Name or Password Cannot be blank !!
OK
Name :- atunu
Password :-
Login
New User Click HERE ???

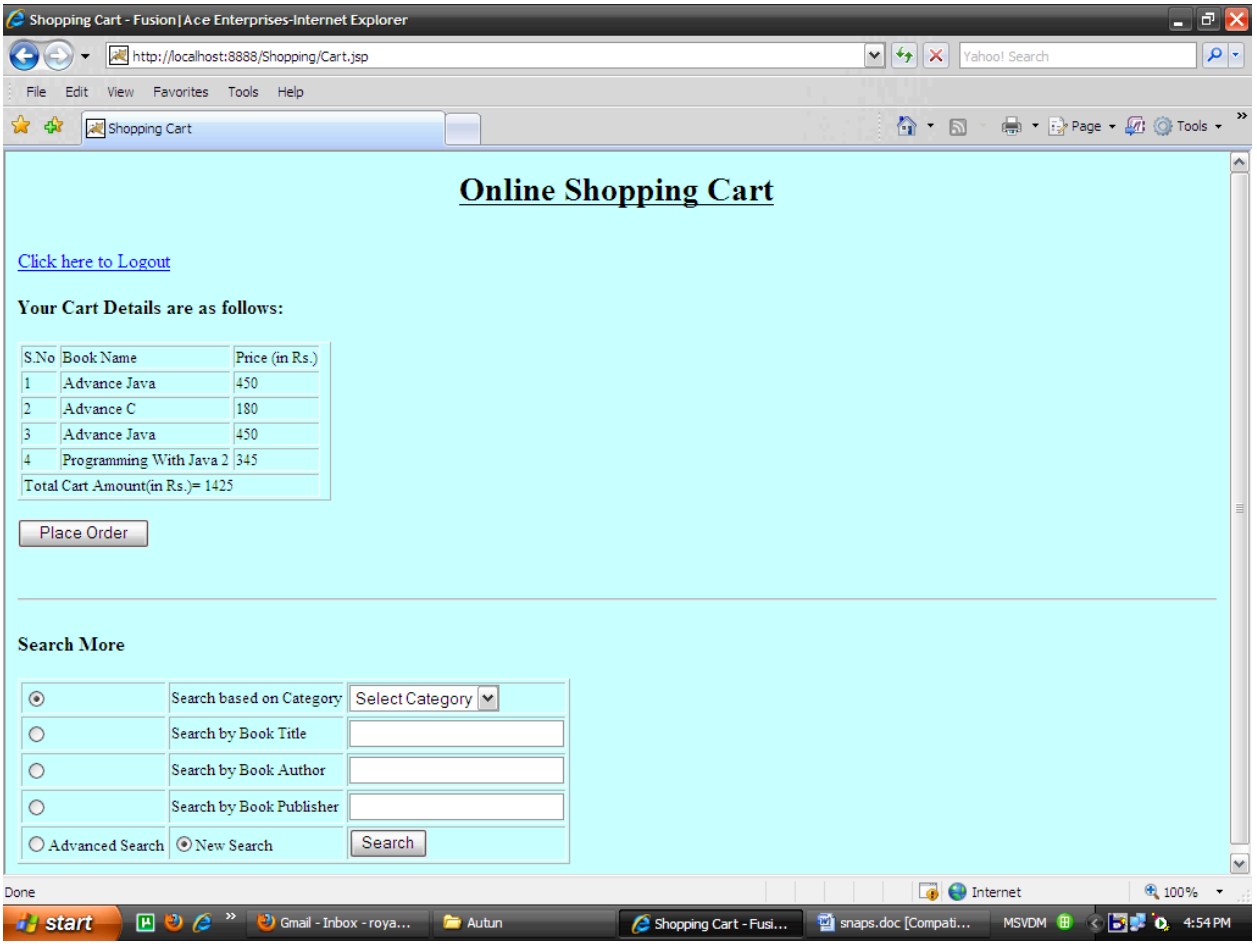

Search Items - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/SearchCriteria.jsp

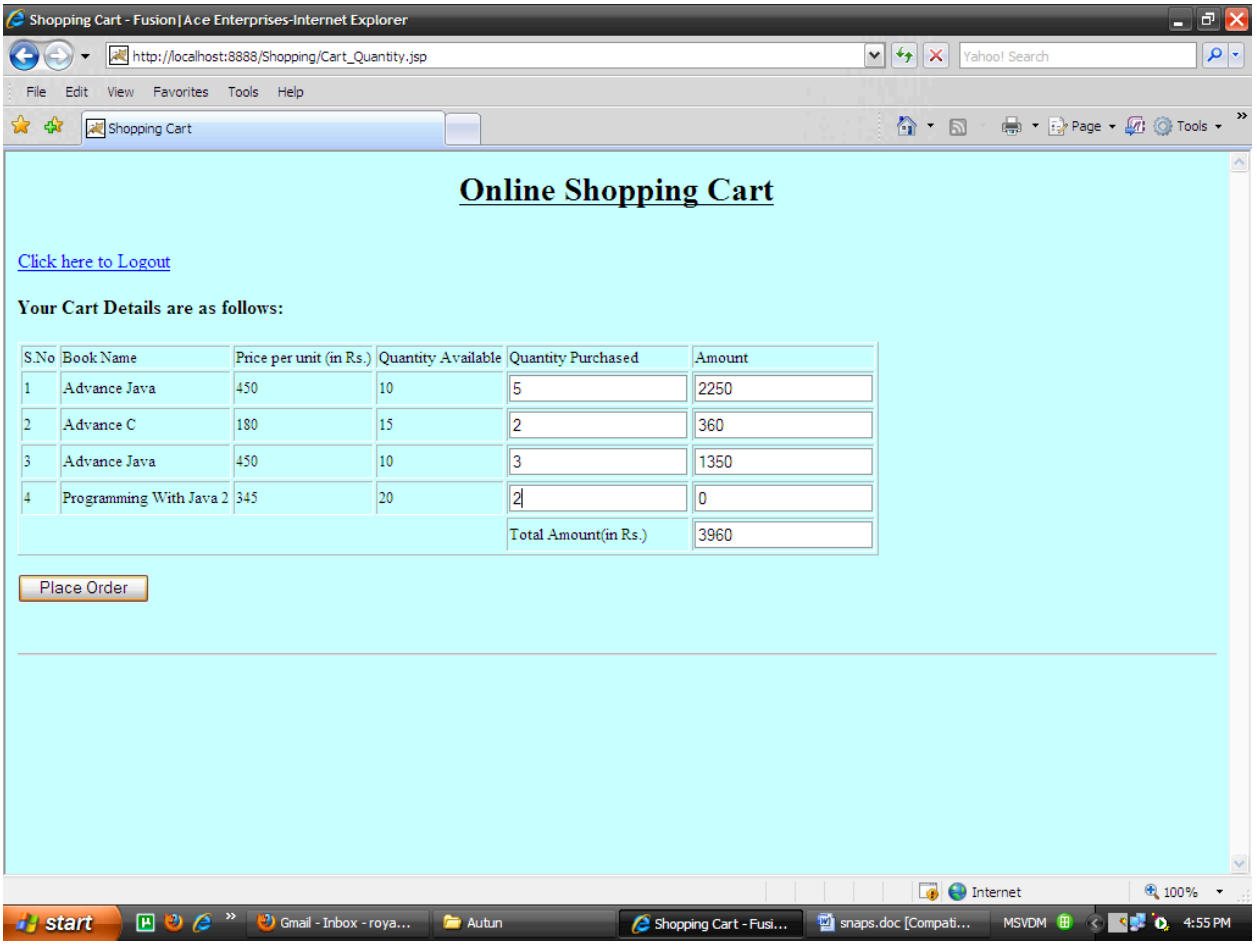


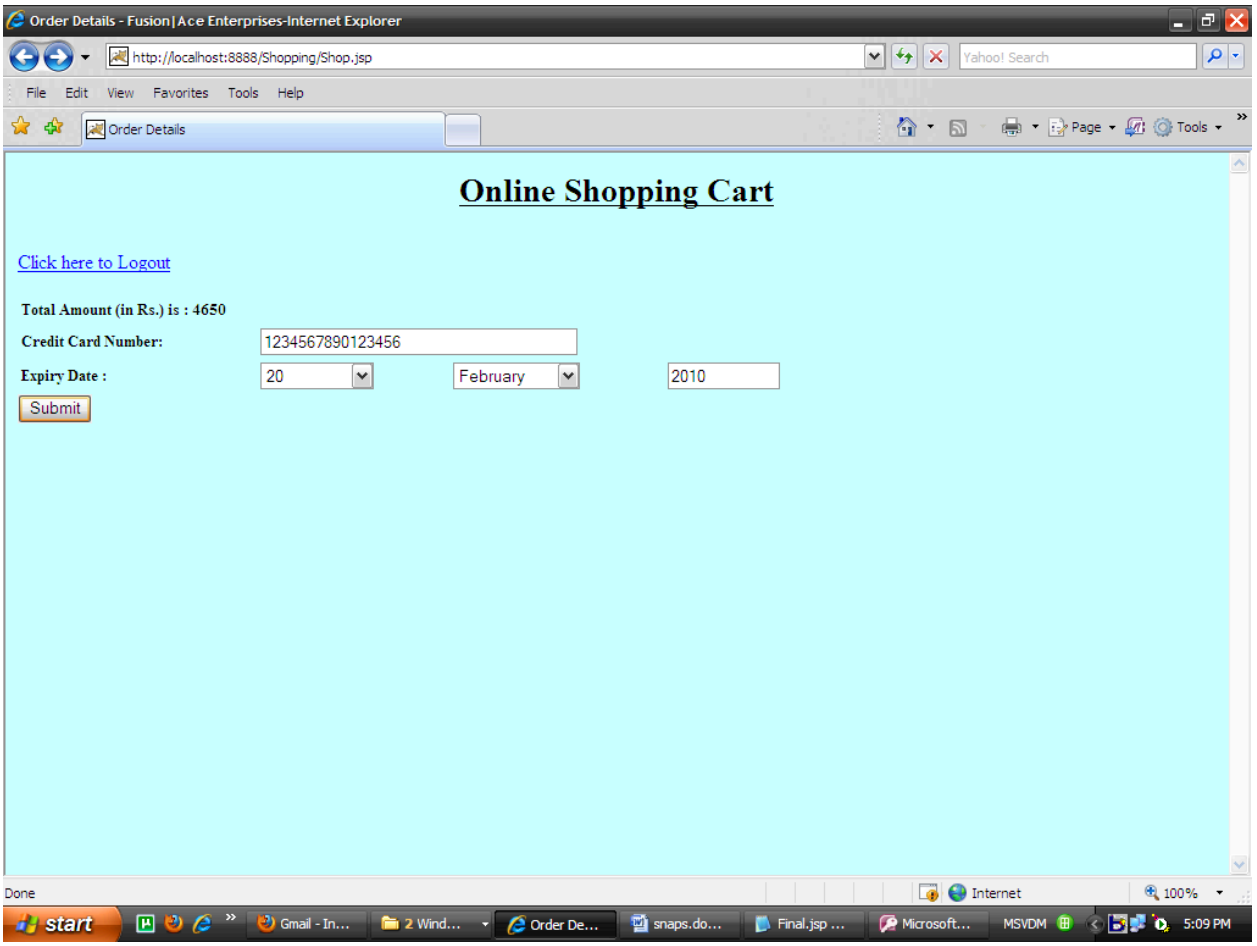
S.No	Select	Book Name	Author Name	Publisher	Edition	Price (in Rs.)	Quantity Available(Nos.)	Description
1	☑	Advance Java	Karnitkar	BPB	2	450	10	About Java
2	☑	Advance C	Karnitkar	BPB	2	180	15	About c
3	☐	Programming With Java 2	Balaguru Shami	BPB	3rd	345	20	About Java
4	☐	Advance Java	Karnitkar	BPB	2	450	10	About Java
5	☐	Advance C	Karnitkar	BPB	2	180	15	About c
6	☐	Programming With Java 2	Balaguru Shami	BPB	3rd	345	20	About Java

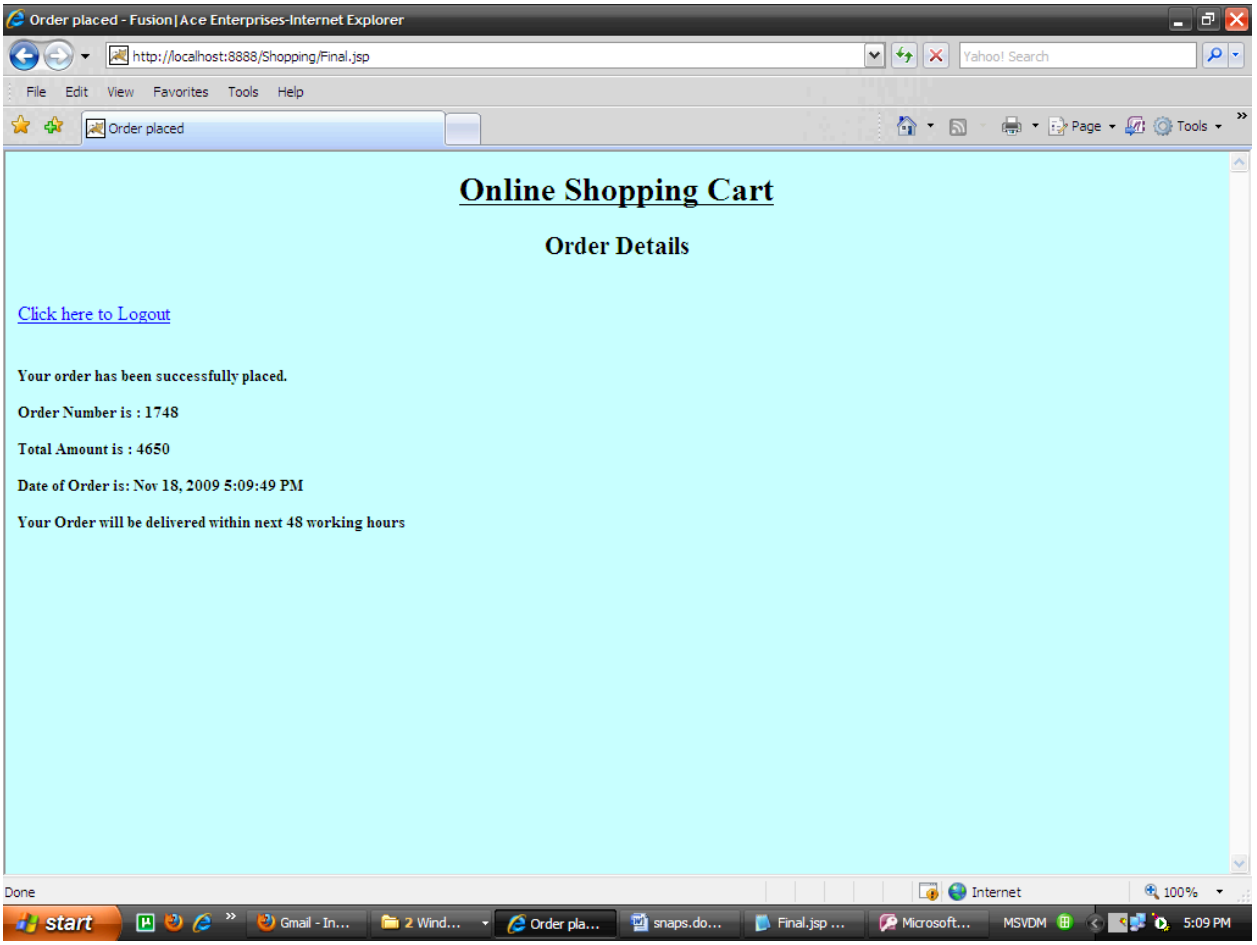

Shopping Cart - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Cart.jsp
Shopping Cart
Online Shopping Cart
Click here to Logout
Your Cart Details are as follows:
S.No Book Name Price (in Rs.)
1 Advance Java 450
2 Advance C 180
Total Cart Amount(in Rs.)= 630
Place Order
Search More
Search based on Category
Select Category
Search by Book Title
Search by Book Author
Search by Book Publisher
Advanced Search
New Search
Search

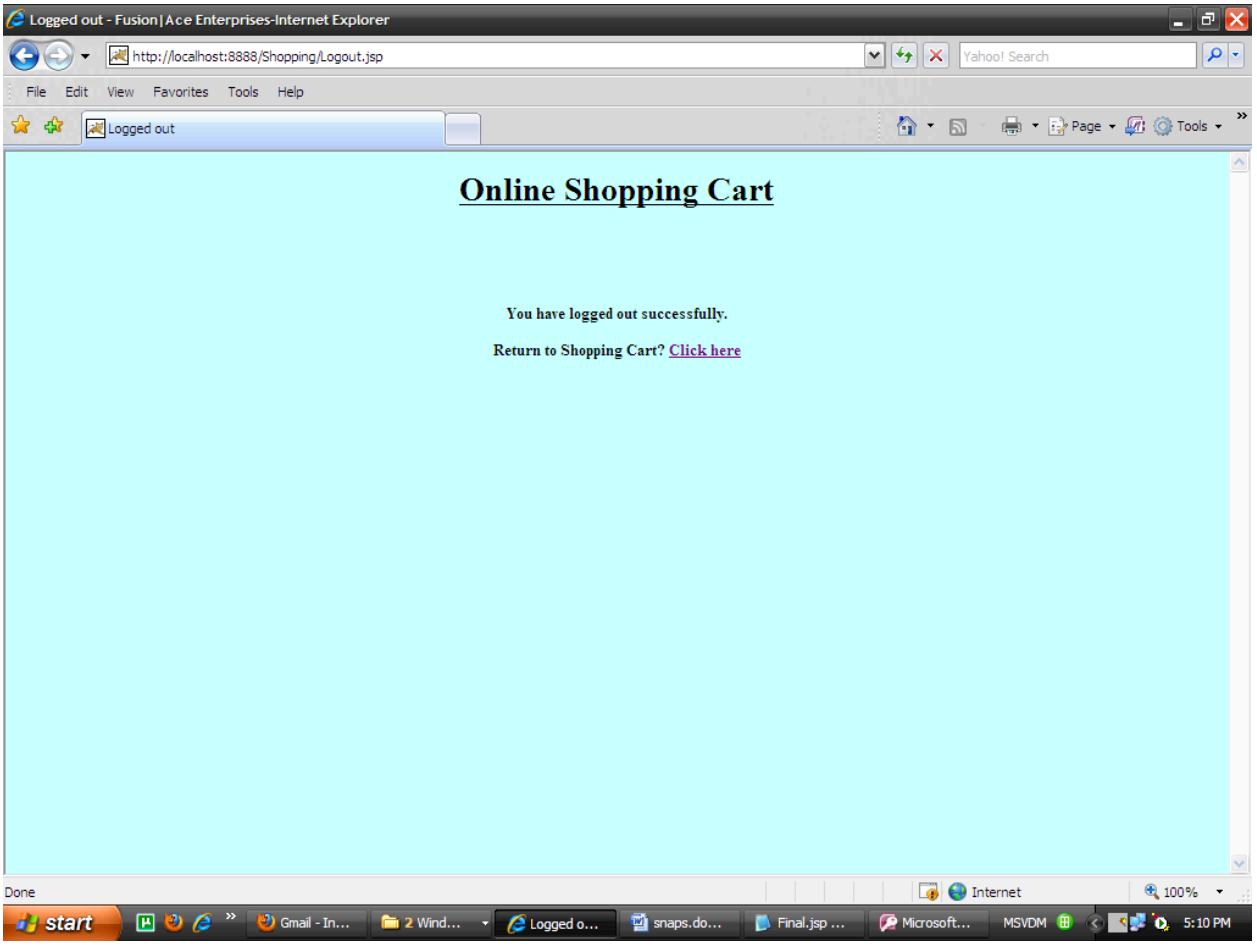

Search Results - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/NewSearch.jsp
File Edit View Favorites Tools Help
Search Results
Online Shopping Cart
Click here to Logout
Your Cart Details are as follows:
S.No Book Name Price (in Rs.)
01 Advance Java 450
11 Advance C 180
Total Cart Amount(in Rs.)= 630
Place Order
Search Results are as follows:
S.No Select Book Name Author Name Publisher Edition Price (in Rs.) Quantity Available(Nos.) Description
1 Advance Java Kamitkar BPB 2 450 10 About Java
2 Programming With Java 2 Balaguru Shami BPB 3rd 345 20 About Java
3 Advance Java Kamitkar BPB 2 450 10 About Java
4 Programming With Java 2 Balaguru Shami BPB 3rd 345 20 About Java
Add to Cart
Search Results are as follows:
Search based on Category
Select Category
Internet
100%
start
Gmail - Inbox - roya...
Autun
Search Results - Fu...
snaps.doc [Compati...
MSVDM
4:53 PM


Shopping Cart - Fusion | Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Cart.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Shopping Cart
Page Tools
Online Shopping Cart
Click here to Logout
Your Cart Details are as follows:
S.No Book Name Price (in Rs.)
1 Advance Java 450
2 Advance C 180
3 Advance Java 450
4 Programming With Java 2 345
Total Cart Amount(in Rs.)= 1425
Place Order
Search More
Search based on Category
Select Category
Search by Book Title
Search by Book Author
Search by Book Publisher
Advanced Search
New Search
Search
Done
Internet
100%
start
Gmail - Inbox - roya...
Autun
Shopping Cart - Fusi...
snaps.doc [Compati...
MSVDM
4:54 PM

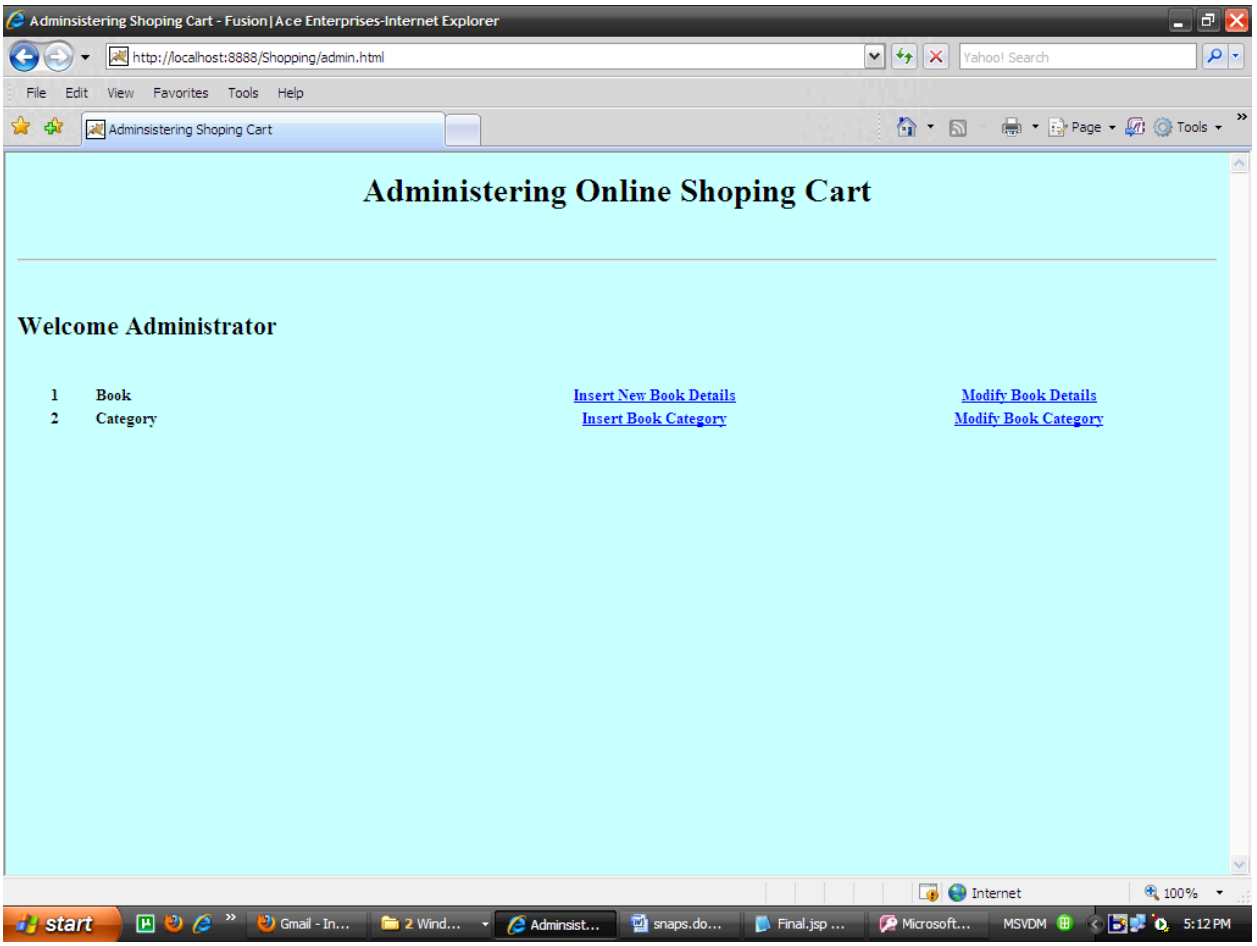

Shopping Cart - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Cart_Quantity.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Shopping Cart
Page Tools
Online Shopping Cart
Click here to Logout
Your Cart Details are as follows:
S.No Book Name Price per unit (in Rs.) Quantity Available Quantity Purchased Amount
1 Advance Java 450 10 5 2250
2 Advance C 180 15 2 360
3 Advance Java 450 10 3 1350
4 Programming With Java 2 345 20 2 0
Total Amount(in Rs.) 3960
Place Order
Internet
100%
start
Gmail - Inbox - roya...
Autun
Shopping Cart - Fusi...
snaps.doc [Compati...
MSVDM
4:55 PM

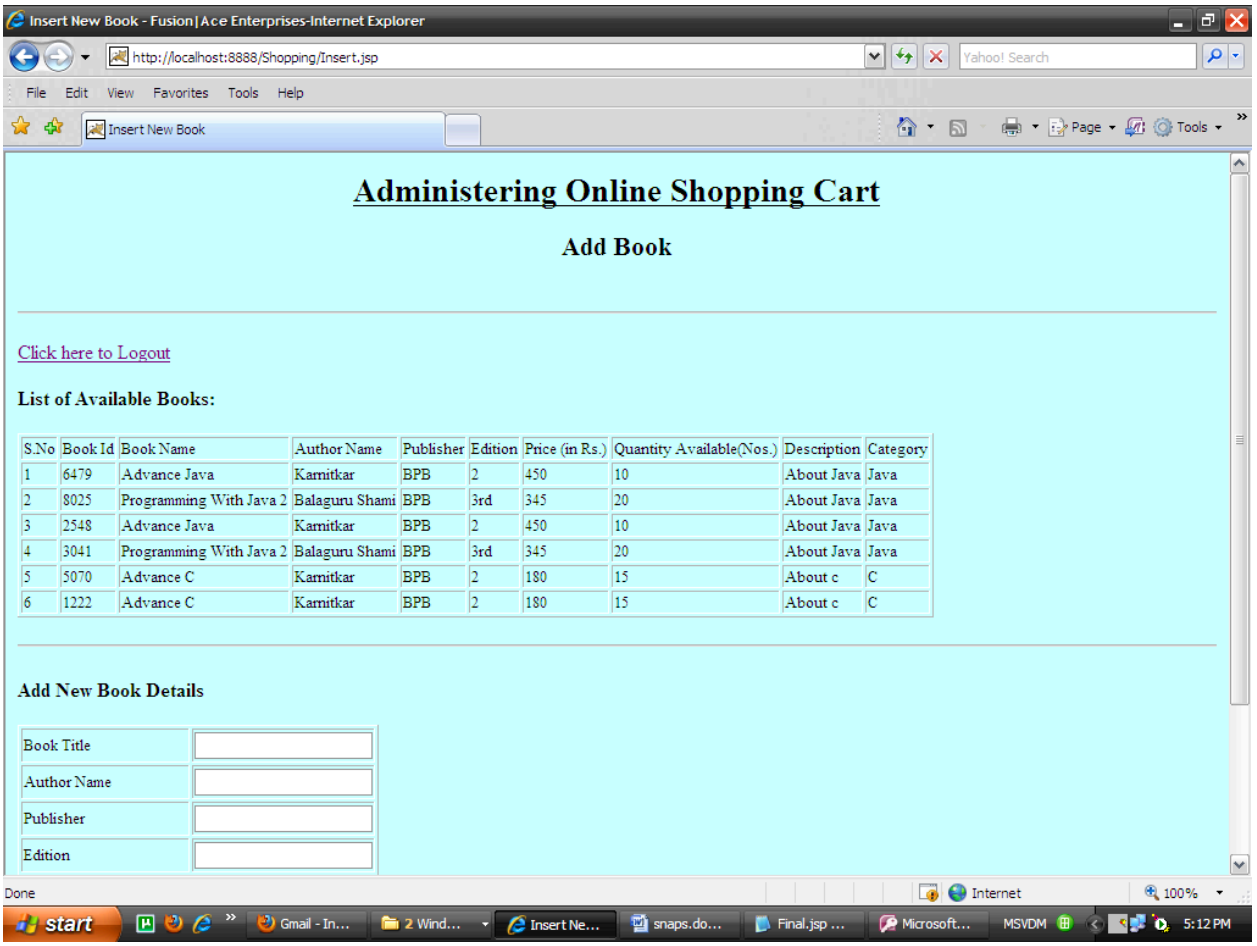

Order Details - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Shop.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Order Details
Page Tools
Online Shopping Cart
Click here to Logout
Total Amount (in Rs.) is : 4650
Credit Card Number:
1234567890123456
Expiry Date :
20
February
2010
Submit
Done
Internet
100%
start
Gmail - In...
2 Wind...
Order De...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:09 PM

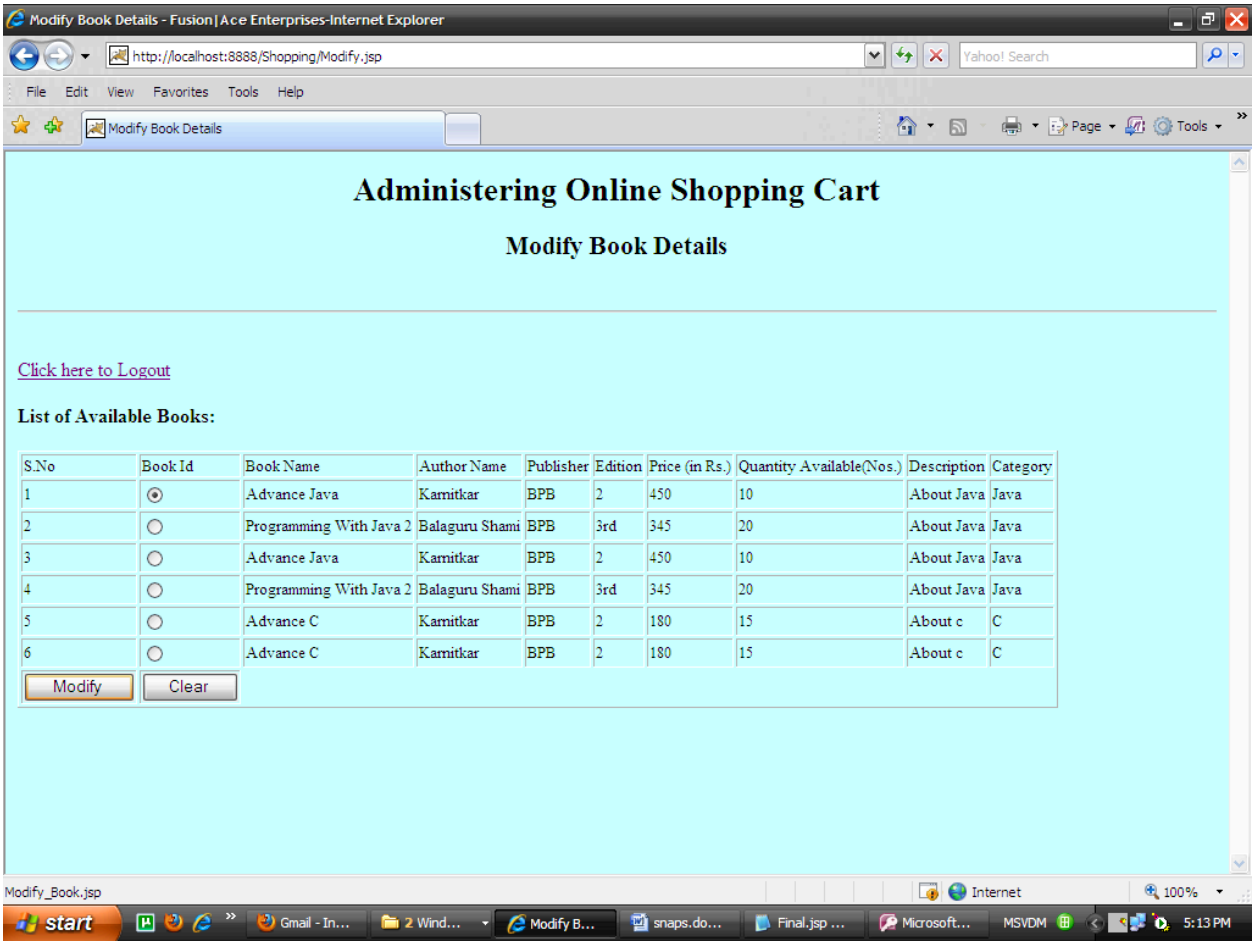

Order placed - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Final.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Order placed
Page Tools
Online Shopping Cart
Order Details
Click here to Logout
Your order has been successfully placed.
Order Number is : 1748
Total Amount is : 4650
Date of Order is: Nov 18, 2009 5:09:49 PM
Your Order will be delivered within next 48 working hours
Done
Internet
100%
start
Gmail - In...
2 Wind...
Order pla...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:09 PM

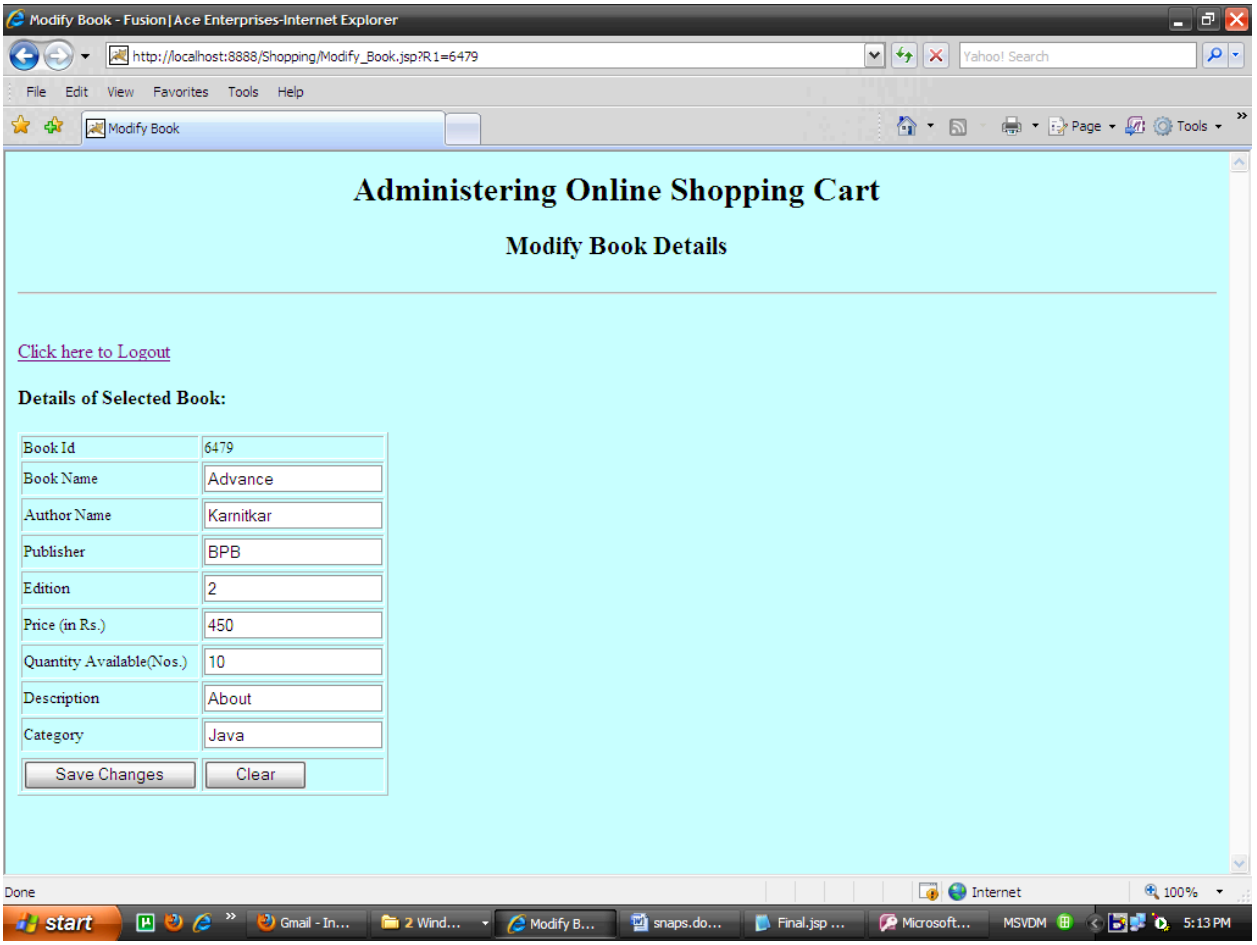

Logged out - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Logout.jsp
File Edit View Favorites Tools Help
Logged out
Online Shopping Cart
You have logged out successfully.
Return to Shopping Cart? Click here
Done
Internet
100%
start
5:10 PM

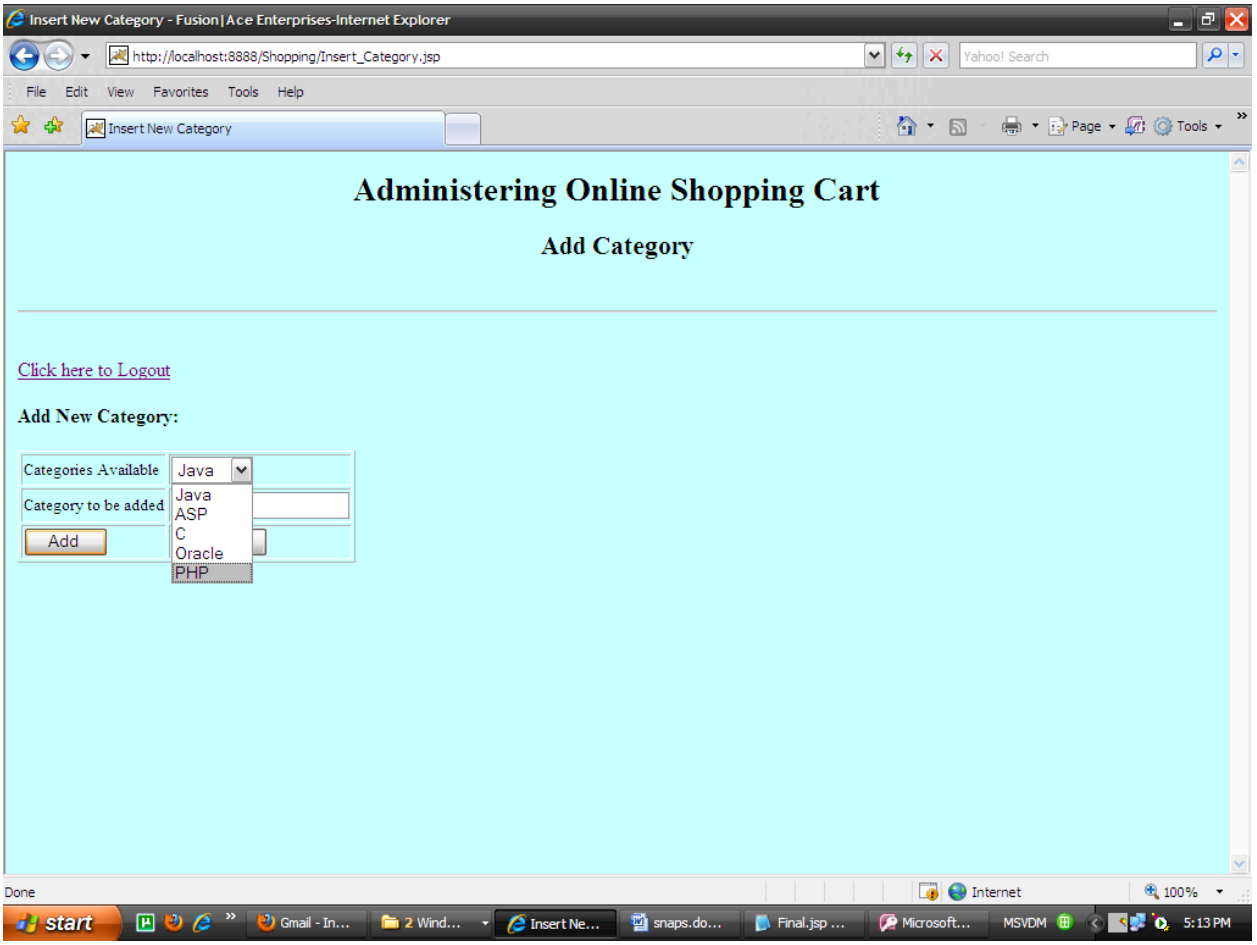
Admin Interface

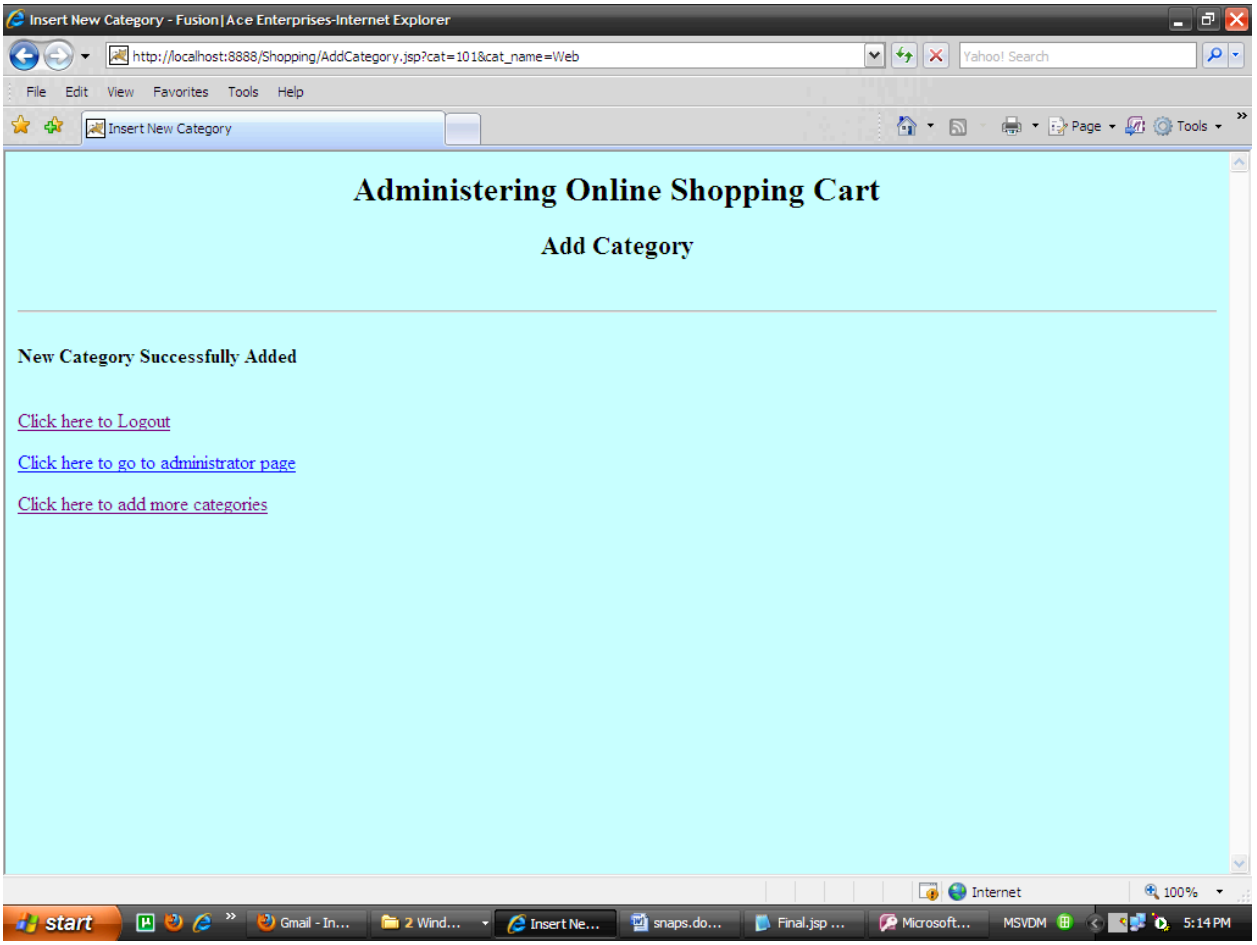


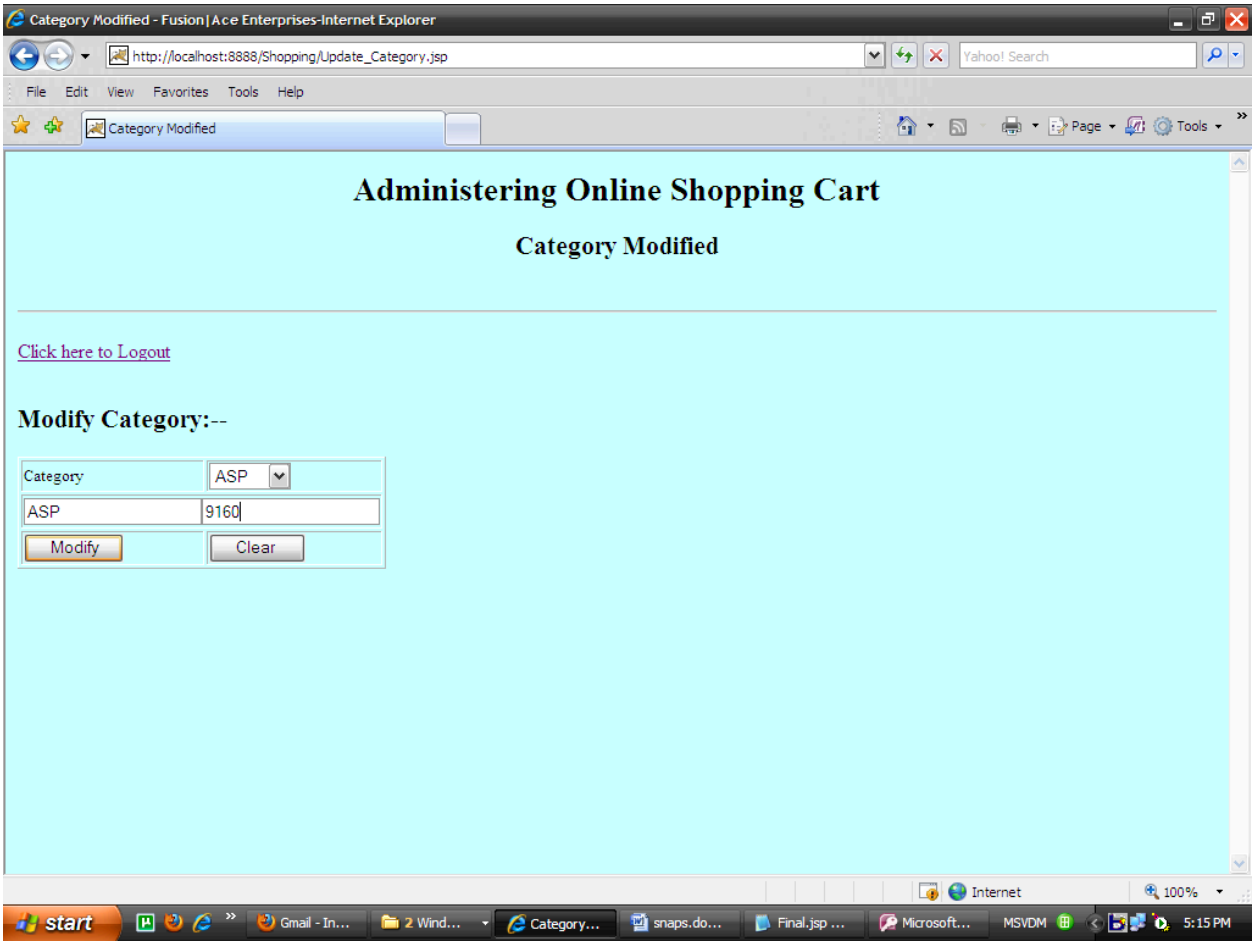

Insert New Book - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Insert.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Insert New Book
Page Tools
Administering Online Shopping Cart
Add Book
Click here to Logout
List of Available Books:
S.No Book Id Book Name Author Name Publisher Edition Price (in Rs.) Quantity Available(Nos.) Description Category
1 6479 Advance Java Karnitkar BPB 2 450 10 About Java Java
2 8025 Programming With Java 2 Balaguru Shami BPB 3rd 345 20 About Java Java
3 2548 Advance Java Karnitkar BPB 2 450 10 About Java Java
4 3041 Programming With Java 2 Balaguru Shami BPB 3rd 345 20 About Java Java
5 5070 Advance C Karnitkar BPB 2 180 15 About c C
6 1222 Advance C Karnitkar BPB 2 180 15 About c C
Add New Book Details
Book Title
Author Name
Publisher
Edition
Done
Internet
100%
start
Gmail - In...
2 Wind...
Insert Ne...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:12 PM

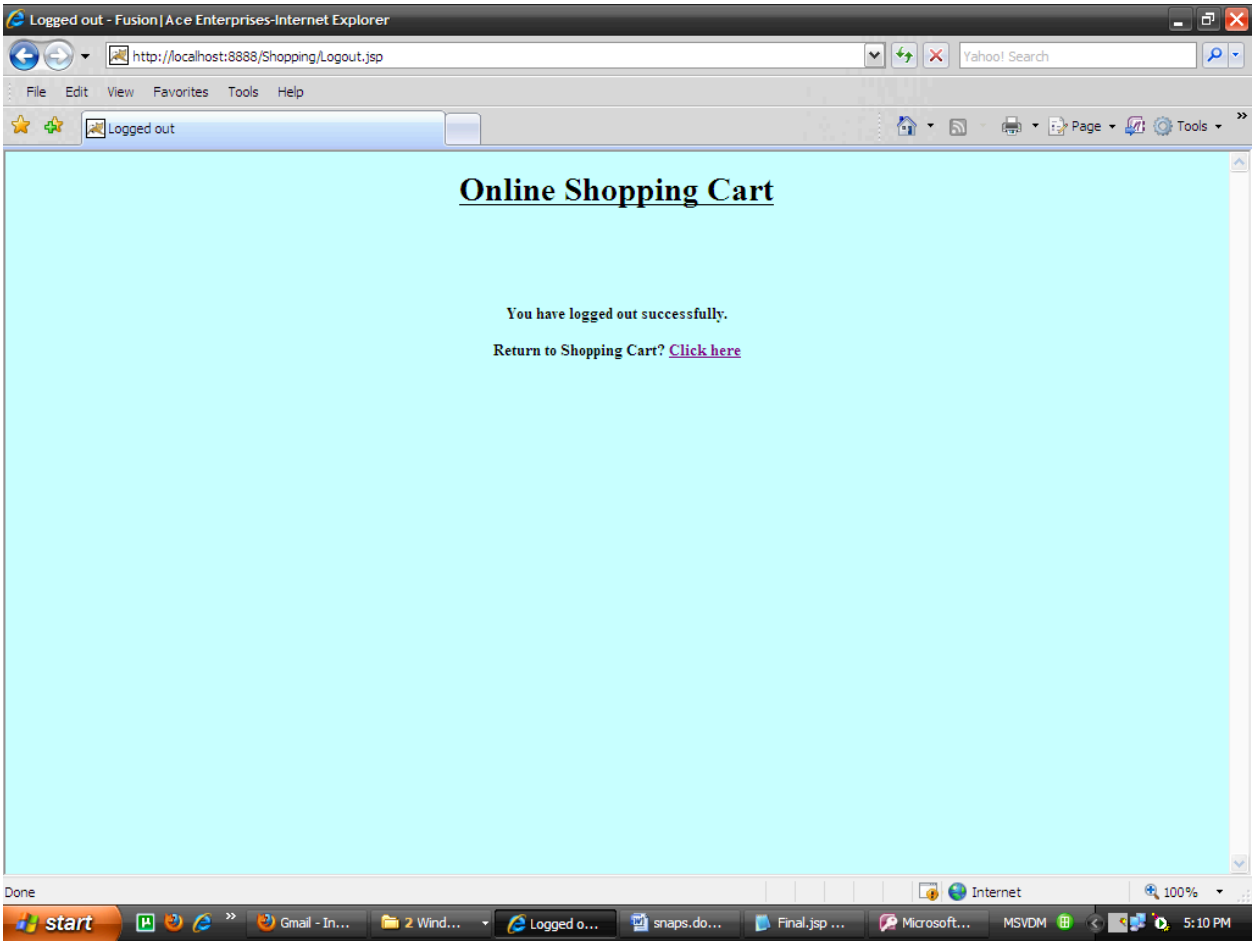

Modify Book Details - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Modify.jsp
Yahoo! Search
File Edit View Favorites Tools Help
Modify Book Details
Page Tools
Administering Online Shopping Cart
Modify Book Details
Click here to Logout
List of Available Books:
S.No | Book Id | Book Name | Author Name | Publisher | Edition | Price (in Rs.) | Quantity Available(Nos.) | Description | Category
1 | Advance Java | Karnitkar | BPB | 2 | 450 | 10 | About Java | Java
2 | Programming With Java 2 | Balaguru Shami | BPB | 3rd | 345 | 20 | About Java | Java
3 | Advance Java | Karnitkar | BPB | 2 | 450 | 10 | About Java | Java
4 | Programming With Java 2 | Balaguru Shami | BPB | 3rd | 345 | 20 | About Java | Java
5 | Advance C | Karnitkar | BPB | 2 | 180 | 15 | About c | C
6 | Advance C | Karnitkar | BPB | 2 | 180 | 15 | About c | C
Modify
Clear
Modify_Book.jsp
Internet
100%
start
Gmail - In...
2 Wind...
Modify B...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:13 PM


Modify Book - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Modify_Book.jsp?R1=6479
File Edit View Favorites Tools Help
Modify Book
Page Tools
Administering Online Shopping Cart
Modify Book Details
Click here to Logout
Details of Selected Book:
Book Id 6479
Book Name Advance
Author Name Karnitkar
Publisher BPB
Edition 2
Price (in Rs.) 450
Quantity Available(Nos.) 10
Description About
Category Java
Save Changes
Clear
Done
Internet
100%
start
Gmail - In...
2 Wind...
Modify B...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:13 PM


Insert New Category - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Insert_Category.jsp
File Edit View Favorites Tools Help
Insert New Category
Page Tools
Administering Online Shopping Cart
Add Category
Click here to Logout
Add New Category:
Categories Available
Java
Category to be added
Add
Java
ASP
C
Oracle
PHP
Done
Internet
100%
start
Gmail - In...
2 Wind...
Insert Ne...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:13 PM


Insert New Category - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/AddCategory.jsp?cat=101&cat_name=Web
Yahoo! Search
File Edit View Favorites Tools Help
Insert New Category
Page Tools
Administering Online Shopping Cart
Add Category
New Category Successfully Added
Click here to Logout
Click here to go to administrator page
Click here to add more categories
Internet
100%
start
Gmail - In...
2 Wind...
Insert Ne...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:14 PM


Category Modified - Fusion|Ace Enterprises-Internet Explorer
http://localhost:8888/Shopping/Update_Category.jsp
File Edit View Favorites Tools Help
Category Modified
Page Tools
Administering Online Shopping Cart
Category Modified
Click here to Logout
Modify Category:--
Category
ASP
ASP
9160
Modify
Clear
Internet
100%
start
Gmail - In...
2 Wind...
Category...
snaps.do...
Final.jsp ...
Microsoft...
MSVDM
5:15 PM



End of Output

7.1Testing

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the attendant “costs” associated with a software failure are motivating forces for well planned through testing.

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery to customer. The goal is to design a series of test cases that online shopping system project report have a high likelihood of finding errors but how? That online shopping system project report’ where software testing techniques enter the pictures.

7.1.1 Testing Objectives

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that online shopping system project report has a high probability of finding an as-yet-undiscovered error.
- A successful test is that online shopping system project report uncovers an as-yet-undiscovered error.

7.1.2 Testing Principle

- All tests should be traceable to customer requirement.
- Tests should be planned long before testing begins.
- The Pareto principle applies to software testing.
- Exhaustive testing is not possible.
- To be most effective, an independent third party should conduct testing.

"Software testing involves executing an implementation of the software with test data and examining the outputs of the software and its operational behavior to check that online shopping system project report it is performing as required. Testing is a dynamic technique of verification and validation because it works with an executable representation of the system "

7.1.3 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design-the software component or module. Using the component – level design description as a guide, important control paths are tested to uncover errors within the boundary of the

module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented and the step can be conducted in parallel for multiple components.

- **Login Module:**

Sr. No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Can ID field be Null?	Null ID	ID cannot be NULL	Warning msg "ID can't be NULL"	Success
2	Can password be Null?	Null password	Password Can't be NULL	Warning msg "password can't be Null"	Success
3	Login button is working or not?	Button pressed	Perform login processing	Call proxy Inbox frame	Success
4.	Is Login Frame displaying properly?	Invoke Login Frame	All text fields are displayed and are properly aligned	Little alignment problem	Success

Server side Login Module

Sr. No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Is Database Connection establishing?	Connection object is created	Connection establishes	No error during connection was found	Success
2	Is able to retrieve Login ID and password from database	Login ID+ Pass-Word	Able to fetch data from Database	No error found during data fetching	Success
3	Is able to match Login ID & Password	Login Id +Pass-wor d	Proper matching	Matching done	Success

Client- Server Login Module combined Testing

Sr. No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Client Server Connection	Connecti on cmd	Connection established	Connection is established	Success
2	Server response handled properly or not?	Login cmd	Proper Message Displayed to User	All type of messages are displayed in proper format to user	Success
3	Communication between client and server	Login cmd	Login cmd is received by server and response is send to client	Communication is taking place.	Success
4	What if User Id doesn't exist	Login cmd+ UserID	Server should report non existence of UserID	Error msg: "Login ID doesn't exist"	Success
5	What if wrong password is entered?	Login cmd+ UserID+ Password	User should be prompt for reentry of password	Error msg: "Invalid Password"	Success

Add Record Module:

Sr. No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1.	Is Database Connection establishing?	Connection object is created	Connection establishes	No error during connection was found	Success
2	Addition of Books.	Book Name/Author/Stock/Edition	Addition Completed.	Addition completed	Success
3	Addition of Books if already exits	Book Name/Author/Stock/Edition	Addition should not complete	Addition is not completed, because record already exists.	Success

- **Search Record Module:**

Sr. No	**Test Case Description**	**Input**	**Expected Behavior**	**Observed behavior**	**Test Result**
1.	Is Database Connection establishing?	Connection object is created	Connection establishes	No error during connection was found	Success
2	On search Criteria of Title/Publisher/Edition should come in list box	Book Name/Author/Stock/Edition	Result found.	Result found.	Success

7.1.4 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that online shopping system project report has been dictated by design.

Incremental integration is the antithesis of the big bang approach. The program is constructed and tested in small increments, where errors are easier to isolate and correct, interfaces are more likely to be tested completely, and a systematic test approach may be applied.

Register Module, Login Module and Logout Module is Integrated

Sr No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Is new User created?	Login ID +personal information	User should be created and personal information should be stored in Database	User named ID is created.	Success
2	Is Database Connection establishing?	Connection object is created	Connection establishes	No error during connection was found	Success
3	Is able to match Login ID & Password	Login Id +Pass-word	Proper matching	Matching done	Success
4	Does status of user changes to 'Logout status'?	Logout cmd	Status should change.	No change in status	Success

- **Add Record Module, Delete Record Module, Update Record Module, Search Record Module is Integrated**

Sr No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Is new Record Added in database?	BookName/Author Name/Edition etc	Records should be added and displayed on the view page	Record is added and displayed on the view page.	Success
2	Is existing Record deleted from database?	BookName/Author Name/Edition etc	Records should be deleted and should not be displayed on the view page	Record is deleted and not displayed on the view page.	Success
3	Is existing Record updated into database?	BookName/Author Name/Edition etc	Records should be updated and should be displayed on the view page	Record is updated and displayed on the view page.	Success
4	On search Criteria names should be correct.	BookName/Author Name/Edition etc	Records should be displayed on the view page	Records found and displayed on the view page.	Success

7.1.5 System Testing

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that online shopping system project report system elements have been properly integrated and perform allocated functions.

- **Functional Requirements**

Sr. No	Test Case Description	Input	Expected Behavior	Observed behavior	Test Result
1	Can New User Register?	Personal Info of User	User should Be registered on DTIS server	User is Registered only If desired Id does Not collide with Existing Ids	Success
2	Can User Login?	LoginId + password	User should Be login	User is login only When Login Id & password is valid	Success
3	Can User Add Records	BookName/ Author Name/Editio n etc	User should be able to Add records	Added records will be displayed	Success
4.	Can User Delete Records	BookName/ Author Name/Editio n etc	User should be able to delete records	Deleted records will not be displayed	Success
5	Can User Update Records	BookName/ Author Name/Editio n etc	User should be able to update records	Updated records will be displayed	Success
6.	Can User Search Records	BookName/ Author Name/Editio n etc.	User should be able to search records	Result of search will be displayed.	Success
7.	Can User Read information available on the site.	Click the link	User should be able to Read information	Information will be displayed.	Success

8	Can User Logout	Click On Logout	User should be able to Logout	Logout Message	Success

7.1.6 OPTIMIZATION POINTS

The software will work efficiently and speedily when the following conditions will be satisfied:

- The server should be of high configuration.
- The client machine has larger RAM.
- Adequate free space on the client's hard disk.
- The user waits until he gets the home page properly.
- No access to the database for writing, deleting or updating by any other except the authority.

Salient Features of the System

- The software is completely menu driven.
- The data entry screens are completely user friendly.
- All editing features and navigation from one field to another, one web page to another, etc is possible.
- Exit from any web page is possible.
- Validation checks have been incorporated in each web page at the appropriate fields.
- Database has been secured by means of password protection.
- Authorization is necessary for all the internal users of the site.

8. Limitations and Future Development

There are some limitations for the current system to which solutions can be provided as a future development:

1. The system is not configured for multi- users at this online shopping system project time. The concept of *transaction* can be used to achieve this online shopping system project.
2. The Website is not accessible to everyone. It can be deployed on a web server so that online shopping system project report everybody who is connected to the Internet can use it.
3. Credit Card validation is not done. Third party proprietary software can be used for validation check.

As for other future developments, the following can be done:

1. The Administrator of the web site can be given more functionalities, like looking at a specific customer's profile, the books that online shopping system project report have to be reordered, etc.
2. Multiple Shopping carts can be allowed.

8.1. Conclusion

The Internet has become a major resource in modern business, thus electronic shopping has gained significance not only from the entrepreneur's but also from the customer's point of view. For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible. As per a survey, most consumers of online stores are impulsive and usually make a decision to stay on a site within the first few seconds. "Website design is like a shop interior. If the shop looks poor or like hundreds of other shops the customer is most likely to skip to the other site. Hence we have designed the project to provide the user with easy navigation, retrieval of data and necessary feedback as much as possible.

In this online shopping system project project, the user is provided with an e-commerce web site that online shopping system project report can be used to buy books online. To implement this online shopping system project as a web application we used JSP as the Technology. JSP has several advantages such as enhanced performance, scalability, built-in security and simplicity. To build any web application using JSP we need a Programming language such as Java and JSP so on. was the language used to build this online shopping system project application. For the client browser to connect to the JSP engine we used Tomcat web server.

JSP uses JDBC to interact with the database as it provides in-memory caching that online shopping system project report eliminates the need to contact the database server frequently and it can easily deploy

and maintain an JSP application. Oracle was used as back-end database since it is one of the most popular commercial databases, and it provides fast data access, easy installation and simplicity.

A good shopping cart design must be accompanied with user-friendly shopping cart application logic. It should be convenient for the customer to view the contents of their cart and to be able to remove or add items to their cart. The shopping cart application described in this online shopping system project project provides a number of features that online shopping system project report are designed to make the customer more comfortable.

This online shopping system project project helps in understanding the creation of an interactive web page and
the technologies used to implement it. The design of the project which includes Data Model and Process Model illustrates how the database is built with different tables, how the data is accessed and processed from the tables. The building of the project has given me a precise knowledge about how JSP is used to develop a website, how it connects to the database to access the data and how the data and web pages are modified to provide the user with a shopping cart application.

9 *Bibliography*

*91 **References***

9.1.1 Books & Authors

- Naughton Schildt, Complete Reference Java 2, Third Edition, TMG.
- Ivor Horton, Beginning Java 2, Wrox Publications.
- Professional Java Server Programming J2EE 1.3 Edition, APress.
- Oracle 9i: The Complete Reference, Kevin Loney & George Koch, Oracle Press.
- Software Engineering, Pressman.

9.1.2 Consulted Websites

- www.google.com
- www.wrox.com
- www.java.sun.com
- www.oracle.com