

Useful Links

1. Github Profile:-
 - a. <https://github.com/abhisheknaidu/awesome-github-profile-readme>
 - i. <https://zzetao.github.io/awesome-github-profile/>
2. Readme File Generator:-
 - a. <https://readme.so/>
3. ScreenShot Editor:-
 - a. <https://uizard.io/screenshot-scanner/>
4. Different AI Models in a single App:-
 - a. <https://play.google.com/store/apps/details?id=us.valkon.privateai>
5. Free Projects with Code:-
 - a. CollegeBasic Projects:-
 - i. <https://www.codewithrandom.com/>
 - ii. <https://www.enggroom.com/>
 - iii. <https://github.com/codecrafters-io/build-your-own-x>
 - iv. Python Project:-
 1. <https://copyassignment.com/top-100-python-projects-with-source-code/>
 - b. ML Projects with DataSet:-
 - i. <https://www.kaggle.com/>
 - ii. <https://data-flair.training/blogs/machine-learning-project-ideas/>
 - c. Useful Python Scripts:-
 - i. <https://github.com/avinashkranjan/Amazing-Python-Scripts>
 - ii. <https://github.com/larymak/Python-project-Scripts>
 - d. React Project with Code:-
 - i. <https://reactjsexample.com/>
6. Article Rewriter Tool:-
 - a. <https://articlerewritertool.com/>
7. Download Published Papers:-
 - a. Sci-hub:-
 - i. <https://pism.in/>
8. Free EBooks:-
 - a. <https://www.pdfdrive.com/>
 - b. Programming books:-
 - i. <https://github.com/EbookFoundation/free-programming-books/tree/main>
9. RoadMap:-
 - a. <https://roadmap.sh/>
10. Cheat Sheet:-
 - a. <https://www.datacamp.com/cheat-sheet>
 - b. <https://overapi.com/>
 - c. <https://devhints.io/>
 - d. <https://shortcuts.design/>
 - e. <https://cheatography.com/>
 - f. <https://levelup.gitconnected.com/35-website-cheat-sheets-for-developers-f8da96aa84f1>
 - g. <https://cheatsheets.shcodes.io/>

- h. <https://kevincobain2000.github.io/golang-cheat-sheet/>
- i. <https://rstudio.github.io/cheatsheets/>
- j. <https://posit.co/resources/cheatsheets/>
- k. <https://github.com/ikatyang/emoji-cheat-sheet?tab=readme-ov-file#face-smiling>
- l. <https://github.com/ChristianLempa/cheat-sheets>
- m. <https://startupsventurecapital.com/essential-cheat-sheets-for-machine-learning-and-deep-learning-researchers-efb6a8ebd2e5>
- n. <https://github.com/wsargent/docker-cheat-sheet?tab=readme-ov-file>
- o. <https://github.com/georgearun/Data-Science--Cheat-Sheet>

11. Free Programming Websites:-

- a. <https://www.w3schools.com/>
- b. <https://www.javatpoint.com/>
- c. <https://www.geeksforgeeks.org/>
- d. <https://www.programiz.com/>
- e. <https://developer.mozilla.org/en-US/>
- f. <https://caddy.tech/>

12. All Programming Algorithms:-

- a. <https://the-algorithms.com/>
 - i. <https://github.com/TheAlgorithms/Python>
 - 1. <https://thealgorithms.github.io/Python/>

13. Coding Practice:-

- a. <https://www.hackerrank.com/>
- b. <https://leetcode.com/>
- c. <https://www.hackerearth.com/for-developers/>
- d. <https://www.codechef.com/>

14. For Developers:-

- a. World Data API
 - i. <https://github.com/Asabeneh/world-countries-data-api>
- b. <https://github.com/jixserver/free-for-dev>
 - i. <https://free-for.dev/#/>

15. Creating CV:-

- a. <https://flowcv.com/dashboard>

16. Online Java Compiler:-

- a. <https://www.guru99.com/try-java-editor.html>

17. Code Converter:-

- a. <https://ide.onelang.io/>

18. Linkedin Skill Assessment Quizzes Solution:-

- a. <https://github.com/Ebazhanov/linkedin-skill-assessments-quizzes>

19. Snippets for JavaScript & React use cases

- a. <https://codetogo.io/>

20. Study Material:-

- a. <https://github.com/Asabeneh/30-Days-Of-JavaScript>
- b. <https://github.com/Asabeneh/30-Days-Of-Python>
- c. <https://github.com/Asabeneh/30-Days-Of-React>
- d. <https://github.com/Asabeneh/10-days-of-git-and-github>
- e. <https://github.com/Asabeneh/30-Days-Of-HTML>
- f. <https://github.com/Asabeneh/JavaScript-for-Everyone>

21. Useful Multi-purpose Website:-

- a. <https://www.photopoea.com/>
- b. <https://www.typingclub.com/>
- c. <https://www.sejda.com/>
- d. <https://tinywow.com/>
- e. <https://www.ilovepdf.com/>

22. Front-End Development:-

a. UI Components:-

- i. <https://tailwindflex.com/>
- ii. <https://kitwind.io/products/kometa/components>
- iii. <https://mambau.com/components>
- iv. <https://fancytailwind.com/>
- v. <https://github.com/unlight/tailwind-components>
- vi. <https://freefrontend.com/>
- vii. <https://www.tailwindmasterkit.com/components>
- viii. <https://tailblocks.cc/>
- ix. <https://flowrift.com/c/banner>
- x. <https://github.com/brillout/awesome-react-components>
- xi. <https://uiverse.io/>
- xii. <https://v0.dev/>
- xiii. <https://magicui.design/>
- xiv. <https://app.haikei.app/>

b. Box Shadows:-

- i. <https://manuarora.in/boxshadows>

c. Fonts Preview:-

- i. <https://wordmark.it/>

d. Images/PSD/Vector:-

- i. <https://www.freepik.com/>

e. Icons:-

- i. <https://react-icons.github.io/react-icons/>
- ii. <https://mui.com/material-ui/material-icons/>
- iii. <https://www.flaticon.com/>
- iv. <https://icon-sets.iconify.design/>

f. Gradient:-

- i. <https://hypercolor.dev/>

g. Meta Tags:-

- i. <https://metatags.io/>

h. Programing Language Documentations:-

- i. <https://devdocs.io/>

i. Dev Tools:-

- i. <https://smalldev.tools/>

j. CSS Class Extractor:-

- i. <http://extractcss.com/>

k. Privacy Policy Generator: -

- i. <https://www.privacypolicyonline.com/>

l. Terms and Condition Generator: -

- i. <https://www.termsandconditionsgenerator.com/>

Useful Code Snippets

1. Validation Schema used for input in forms using Formik and yup:-

a. Email:-

```
email: Yup.string()  
  .email("Invalid email address")  
  .required("E-Mail address is required."),
```

b. Phone Number:-

```
phoneNumber: Yup.string()  
  .matches(/^[0-9]+$/, "Invalid phone number")  
  .min(10, "Phone number must be at least 10 characters")  
  .max(15, "Phone number can't exceed 15 characters")  
  .required("Phone Number is required."),
```

c. Message:-

```
message: Yup.string().required("Message is required."),
```

d. Bank Details:-

i. Aadhar No:-

```
validationSchema: Yup.string()  
  .required("Aadhar Number is Required")  
  .matches(  
    /^[2-9]{1}[0-9]{3}[0-9]{4}[0-9]{4}$/,  
    "Invalid aadhaar number"  
  ),
```

ii. Pan No:-

```
validationSchema: Yup.string()  
  .required("Pan Number is Required")  
  .matches(  
    /^[A-Z]{5}[0-9]{4}[A-Z]{1}$/,  
    "Invalid PAN number. Use capital letters and numbers."  
  ),
```

iii. Bank Name:-

```
validationSchema: Yup.string()  
  .required("BankName is Required")  
  .max(50, "Bank Name must be less than 50 characters")  
  .min(5, "Bank Name must be greater than 5 characters"),
```

iv. Account Number:-

```
validationSchema: Yup.string()  
  .required("Account Number is Required")  
  .matches(/^[0-9]{9,18}$/, "Invalid Account Number"),
```

v. IFSC Code:-

```
validationSchema: Yup.string()  
  .required("IFSC Code is Required")  
  .matches(/^[A-Z]{4}[0][A-Z0-9]{6}$/, "Invalid IFSC Code"),
```

e. Password

i. New Password:-

```
validationSchema: Yup.string()
  .min(8, "Password must be at least 8 characters")
  .matches(/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/, "Password must include at least 1 uppercase letter, 1 lowercase letter, 1 symbol, and 1 number")
  .required("Password is required"),
```

ii. Confirm Password:-

```
validationSchema: Yup.string()
  .min(8, "Confirm password must be at least 8 characters")
  .required("Confirm password is required")
  .oneOf([Yup.ref("password"), null], "Passwords must match"),
```

f. Price

i. Mrp Price:-

```
validationSchema: yup.string()
  .required("Mrp is required")
  .min(1, "Minimum Amount Should be 1")
  .test("non-negative", "Price should be positive",
    (value) => {
      return parseFloat(value) >= 0;
    }
  )
```

ii. Sale Price:-

```
validationSchema: yup
  .string()
  .required("Sale Price is required")
  .min(1, "Minimum Amount Should be 1")
  .test("non-negative", "Price should be positive",
    (value) => {
      return parseFloat(value) >= 0;
    }
  )
  .test(
    "less-than-or-equal-to-cost-price",
    "Sale Price must be less than or equal to Cost Price",
    function (value) {
      const mrpPrice = this.parent.mrpPrice;
      return parseFloat(value) <= parseFloat(mrpPrice);
    }
  ),
```

g. Time and date

i. Start Time:-

```
startTime: Yup.string()
  .test(
    "is-future",
    "Start time must be in the future",
    function (value) {
      const startDateTime = new Date(value as string);
      return startDateTime > new Date();
    }
  )
  .required("Start Time Required!"),
```

ii. End Time:-

```
endTime: Yup.string()
  .test(
    "is-after-start",
    "End time must be after start time",
    function (value) {
      const startDateTime = new
Date(`${this.parent.startTime}`);
      const endDateTime = new Date(`${value}`);
      return endDateTime > startDateTime;
    }
  )
  .required("End Time Required!"),
```

2. Hide Scroll Bar in tailwind CSS:-

```
/* Hide scrollbar for Chrome, Safari and Opera */
.no-scrollbar::-webkit-scrollbar {
  display: none;
}
/* Hide scrollbar for IE, Edge and Firefox */
.no-scrollbar {
  -ms-overflow-style: none; /* IE and Edge */
  scrollbar-width: none; /* Firefox */
}
```

Simple Roadmap for Full Stack Developer

1. Frontend Development:

a. HTML, CSS, JavaScript (Basic and Advanced)

b. React.js (Basics to Advanced)

- i. React Components, Props, State
- ii. React Hooks
- iii. Redux or Context API for state management
- iv. React Router for client-side routing
- v. Styled Components or CSS Modules for styling

c. Next.js

- i. Server-side Rendering (SSR)
- ii. Static Site Generation (SSG)
- iii. Data Fetching (getStaticProps, getServerSideProps)
- iv. API Routes
- v. Image Optimization

2. Backend Development:

a. Node.js Core Concepts and Modules

- i. Event Loop
- ii. Asynchronous Programming
- iii. File System, HTTP, and other built-in modules

b. Express.js

- i. Routing

- ii. Middleware
- iii. Error Handling
- iv. RESTful API Design

c. Database Fundamentals

- i. Types of Databases (Relational, NoSQL)
- ii. Choose one Database (e.g., PostgreSQL, MongoDB)
- iii. ORM (e.g., Sequelize for PostgreSQL, Mongoose for MongoDB)

d. Database Operations

- i. CRUD Operations
- ii. Data Validation
- iii. Transactions

e. Query Optimization

- i. Indexing
- ii. Query Tuning
- iii. Caching Strategies

f. Advanced CRUD Operations

- i. Joins
- ii. Aggregation

- iii. Pipelines

3. Architecture and Deployment:

a. Backend Architecture

- i. Monolithic Architecture
- ii. Microservices Architecture

b. Cloud Computing

- i. AWS, Azure, Google Cloud Platform
- ii. Deploying applications on cloud services
- iii. Scalability and Load Balancing

c. Continuous Integration and Deployment (CI/CD)

- i. Basics of CI/CD
- ii. Jenkins, Travis CI, GitLab CI/CD
- iii. Automation of Testing and Deployment Processes

d. Testing

- i. Unit Testing (e.g., Jest, Mocha)
- ii. Integration Testing
- iii. API Testing (e.g., Postman, Supertest)

4. Networking and Security:

a. Basic Networking

- i. TCP/IP, HTTP/HTTPS
- ii. DNS, IP addressing
- iii. OSI Model

b. Secure Protocols

- i. SSL/TLS
- ii. OAuth, JWT for authentication
- iii. HTTPS configuration

c. Security Best Practices

- i. Input Validation
- ii. Authentication and Authorization
- iii. Encryption