

**Exp No: 1**

## **Develop a personal profile website using HTML and CSS**

### **Objective:**

The objective is to design and develop a personal profile website using the common tags of HTML (HyperText Markup Language) such as link, table, form etc., and CSS (Cascading Style Sheets). All 3 types of CSS will be used. The website will be hosted in the Github pages.

### **Procedure :**

1. Create HTML File: Save the document as `resume.html` in project.
2. Basic Structure: Set up the standard HTML structure with the head and body.
3. Page Layout: Split the page into header and main sections for content.
4. Insert Content: Add profile, education, skills, projects, and contact form sections.
5. Add CSS: Use CSS for layout, color, fonts, and overall styling.
6. Test in Browser: Open the file in a browser to review and adjust the layout.

### **Complete Source code :**

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Abinaya P - Resume</title>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }
    body {
      font-family: 'Times New Roman';
      line-height: 1.6;
      color: #333;
      max-width: 1200px;
    }
  </style>
</head>
<body>
  <h1>Abinaya P</h1>
  <p>Software Developer</p>
  <table>
    <thead>
      <tr>
        <th>Section</th>
        <th>Content</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Profile</td>
        <td>Abinaya P</td>
      </tr>
      <tr>
        <td>Education</td>
        <td>B.Tech in Computer Science</td>
      </tr>
      <tr>
        <td>Skills</td>
        <td>HTML, CSS, JavaScript, Python</td>
      </tr>
      <tr>
        <td>Projects</td>
        <td>Web Development, Data Structures, Machine Learning</td>
      </tr>
      <tr>
        <td>Contact</td>
        <td>Email: abinaya.p@outlook.com</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

Rollno: 22AD002

Name: ABINAYA P

```
margin: 0 auto;
padding: 40px;
display: flex;
background-color: #f4f4f4;
}
h1, h2 {
  color: #2c3e50;
}
header {
  width: 30%;
  padding: 30px;
  background-color: #fff;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
.photo-container {
  text-align: center;
  margin-bottom: 30px;
}
.photo-container img {
  width: 180px;
  height: 180px;
  border-radius: 50%;
  object-fit: cover;
  border: 3px solid #0077b5;
}
h1 {
  margin: 20px 0 10px;
  font-size: 28px;
}
.student-title {
  font-style: italic;
  margin-bottom: 20px;
  color: #7f8c8d;
}
.contact-info p {
  margin: 10px 0;
}
.contact-info a {
  text-decoration: none;
}
.linkedin {
  color: #0077b5;
}
.linkedin:hover {
  text-decoration: underline;
}
.github {
```

Rollno: 22AD002

Name: ABINAYA P

```
        color: #0077b5;
    }
.github:hover {
    text-decoration: underline;
}
main {
    width: 70%;
    padding: 30px 40px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
.section {
    margin-bottom: 30px;
}
.section h2 {
    border-bottom: 2px solid #3498db;
    padding-bottom: 10px;
    margin-bottom: 15px;
}
ul {
    padding-left: 20px;
}
li {
    margin-bottom: 5px;
}
table {
    width: 100%;
    border-collapse: collapse;
    margin-bottom: 30px;
}
table, th, td {
    border: 1px solid #ddd;
}
th, td {
    padding: 10px;
    text-align: left;
}
th {
    background-color: #3498db;
    color: white;
}
form {
    margin-top: 30px;
    background-color: #f9f9f9;
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
```

Rollno: 22AD002

Name: ABINAYA P

```
        }
    form h2 {
        margin-bottom: 20px;
        border-bottom: none;
    }
    form label {
        display: block;
        margin-bottom: 10px;
        font-weight: bold;
    }
    form input, form textarea {
        width: 100%;
        padding: 10px;
        margin-bottom: 20px;
        border: 1px solid #ddd;
        border-radius: 5px;
        font-size: 16px;
    }
    form button {
        padding: 10px 20px;
        background-color: #3498db;
        color: white;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
    }
    form button:hover {
        background-color: #2980b9;
    }
</style>
</head>
<body>
<header>
    <div class="photo-container">
        
    </div>
    <h1>ABINAYA P</h1>
    <p class="student-title">Student</p>
    <div class="contact-info">
        <p>+91 88704 31789</p>
        <p>pabinayaabinaya04@gmail.com</p>
        <p><a href="https://www.linkedin.com/in/abinaya-p-43287526a" class="linkedin">LinkedIn</a></p>
        <p><a href="https://github.com/Abi-1724" class="github">GitHub</a></p>
    </div>
</header>
```

Rollno: 22AD002

Name: ABINAYA P

```
<main>
  <section class="section">
    <h2>Profile</h2>
    <p>Talented and dedicated Frontend Developer with expertise in SQL, Python, Java, and data science, coupled with intermediate skills in machine learning. Proven experience in developing user-friendly applications using Streamlit for real-time data prediction and analysis, and a strong background in database management and operations. Highly motivated to leverage technical and analytical skills to solve complex problems and drive innovation.</p>
  </section>

  <section class="section">
    <h2>Education</h2>
    <table>
      <tr>

        <th>School & Institution Name</th>
        <th>Year</th>
        <th>Percentage</th>
      </tr>
      <tr>
        <td>KPR Institute of Engineering & Technology, Coimbatore, (Bachelor of Technology in AI & DS).</td>
        <td>2022-2026</td>
        <td>8.356</td>
      </tr>
      <tr>
        <td>Vidhya Parthi Higher Secondary School, Dindigul, (12th Standard).</td>
        <td>2021-2022</td>
        <td>89%</td>
      </tr>
      <tr>
        <td>Vidhya Parthi Higher Secondary School, Dindigul, (10th Standard).</td>
        <td>2019-2020</td>
        <td>92%</td>
      </tr>
    </table>
  </section>

  <section class="section">
    <h2>Tech Stack</h2>
```

Rollno: 22AD002

Name: ABINAYA P

```
<ul>
    <li>Programming (Java, Python, SQL, C)</li>
    <li>Frontend Coding (HTML, CSS, JavaScript)</li>
    <li>Data Science</li>
    <li>Website Development</li>
    <li>Machine Learning</li>
    <li>Exploratory Data analysis</li>
</ul>
</section>

<section class="section">
    <h2>Skills</h2>
    <ul>
        <li>Strong problem-solving abilities and analytical thinking</li>
        <li>Dynamic person</li>
        <li>Strong communication skills</li>
    </ul>
</section>

<section class="section">
    <h2>Certificates</h2>
    <ul>
        <li>National Hackfest Finalist Participation</li>
        <li>C Programming by Barathidhasan University</li>
        <li>Machine learning by INTEL</li>
        <li>NPTEL (Data Analytics with Python, English for Competitive Exam)</li>
    </ul>
</section>

<section class="section">
    <h2>Internships</h2>
    <ul>
        <li>Yaane Technologies, Coimbatore (Image Processing application)</li>
        <li>CodTech It Solutions, Hyderabad (Basic Authentication System Project)</li>
    </ul>
</section>

<section class="section">
    <h2>Participation & Projects</h2>
    <ul>
        <li>NATIONAL HACKFEST Hackathon at PSG itech college (Finalist), Coimbatore</li>
        <li>Presentation on TRASH TERMINATOR: THE AI POWERED ROBOTIC BOAT</li>
        <li>INFO HUNT - news collection webapp using API</li>
    </ul>
</section>
```

Rollno: 22AD002

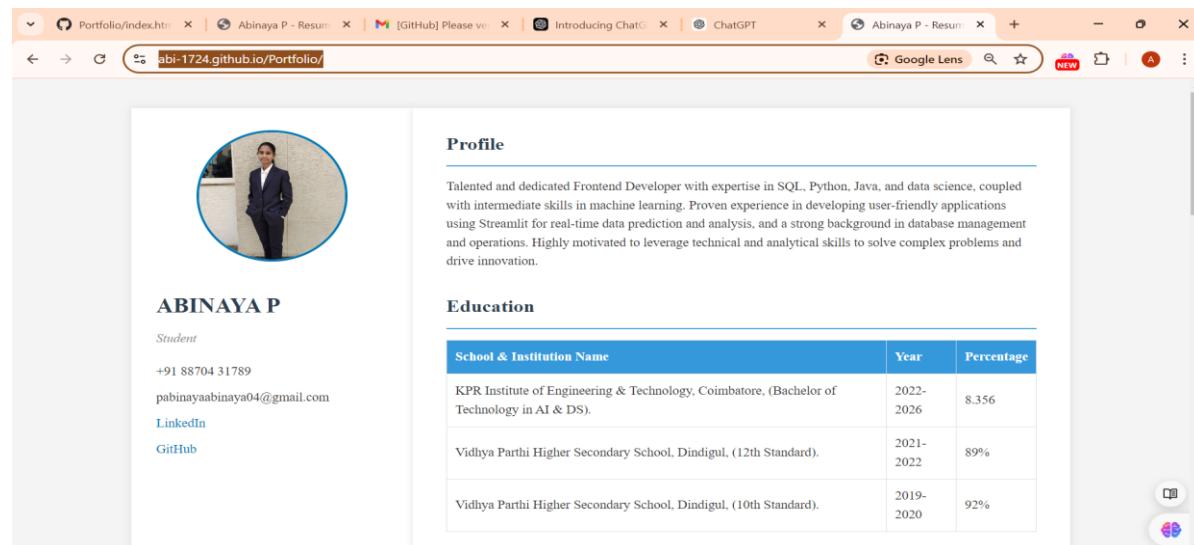
Name: ABINAYA P

```
<li>Energy Consumption - predict power consumption using Machine  
Learning</li>  
</ul>  
</section>  
  
<section class="section">  
  <h2>Contact Me</h2>  
  <form action="/submit" method="post">  
    <h2>Contact Form</h2>  
    <label for="name">Name:</label>  
    <input type="text" id="name" name="name" required>  
  
    <label for="email">Email:</label>  
    <input type="email" id="email" name="email" required>  
  
    <label for="message">Message:</label>  
    <textarea id="message" name="message" rows="5" required></textarea>  
  
    <button type="submit">Submit</button>  
  </form>  
</section>  
</main>  
</body>  
</html>
```

## Output:

URL : <https://abi-1724.github.io/Portfolio/>

### Screen 1 : About me page and Education



The screenshot shows a web browser window with multiple tabs open. The active tab displays a resume page for 'Abinaya P'. The page features a circular profile picture of a woman, her name 'ABINAYA P' in bold capital letters, and her title 'Student'. Below this, there are links for 'LinkedIn' and 'GitHub'. The main content area is divided into two sections: 'Profile' and 'Education'. The 'Profile' section contains a brief biography: 'Talented and dedicated Frontend Developer with expertise in SQL, Python, Java, and data science, coupled with intermediate skills in machine learning. Proven experience in developing user-friendly applications using Streamlit for real-time data prediction and analysis, and a strong background in database management and operations. Highly motivated to leverage technical and analytical skills to solve complex problems and drive innovation.' The 'Education' section is presented in a table format:

School & Institution Name	Year	Percentage
KPR Institute of Engineering & Technology, Coimbatore, (Bachelor of Technology in AI & DS).	2022-2026	8.356
Vidhya Parthi Higher Secondary School, Dindigul, (12th Standard).	2021-2022	89%
Vidhya Parthi Higher Secondary School, Dindigul, (10th Standard).	2019-2020	92%

Rollno: 22AD002

Name: ABINAYA P

## Tech Stack and Certificates

The screenshot shows a resume page titled "Abinaya P - Resume" on a browser window. The content is organized into sections:

- Tech Stack**
  - Programming (Java, Python, SQL, C)
  - Frontend Coding (HTML, CSS, JavaScript)
  - Data Science
  - Website Development
  - Machine Learning
  - Exploratory Data analysis
- Skills**
  - Strong problem-solving abilities and analytical thinking
  - Dynamic person
  - Strong communication skills
- Certificates**
  - National Hackfest Finalist Participation
  - C Programming by Barathidhasan University
  - Machine learning by INTEL
  - NPTEL (Data Analytics with Python, English for Competitive Exam)

The browser's address bar shows the URL: `abi-1724.github.io/Portfolio/`. The system tray at the bottom indicates the date as 24-10-2024 and the time as 09:20.

## Internship ,Projects and contact me

The screenshot shows a resume page titled "Abinaya P - Resume" on a browser window. The content is organized into sections:

- Internships**
  - C Programming by Barathidhasan University
  - Machine learning by INTEL
  - NPTEL (Data Analytics with Python, English for Competitive Exam)
- Participation & Projects**
  - Yann Technologies, Coimbatore (Image Processing application)
  - CodTech It Solutions, Hyderabad (Basic Authentication System Project)
- Contact Me**

**Contact Form**

Name:

Email:

Message:

The browser's address bar shows the URL: `abi-1724.github.io/Portfolio/`. The system tray at the bottom indicates the date as 24-10-2024 and the time as 09:21.

Rollno: 22AD002

Name: ABINAYA P

Exp No: 2	Develop a website with client side form validation
-----------	--

### **Objective:**

The objective of the code is to create a simple website with client side form validation

### **Procedure :**

- Structure: Design separate forms for login and signup within a container element to keep them visually organized.
- Form Fields: Include fields for username, password, email, age, gender, book name, author, book format, and role (reader or creator).
- Global Styling: Add a background image, center the container, and give it a polished look with transparency and shadow.
- Form Fields: Style inputs, selects, and buttons to be full-width with padding and border-radius for user-friendly interaction.
- Radio Buttons: Align radio buttons and labels vertically for better readability.
- Modal Styling: Set up styles for a modal popup (optional) for additional user interactions or alerts.
- Form Toggle: `toggleForms()` switches visibility between the login and signup forms.
- Login: `login(event)` prevents form submission, checks if all fields are filled, and shows a success or error message based on input.
- Signup: `signup(event)` prevents form submission, validates email and password with regular expressions, and shows alerts for valid or invalid input.

### **Complete Source code :**

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
```

Rollno: 22AD002

Name: ABINAYA P

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Comic Book Registration and Login</title>
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.1/css/all.min.css" rel="stylesheet">
<link href="style.css" rel="stylesheet">
<style>
.form-group.radio-group {
    text-align: left; /* Align text to the left */
    margin-left: 0; /* Align to the left edge of the form */
    margin-bottom: 16px; /* Add space below the radio buttons */
}

.radio-group label {
    display: inline-block; /* Keep label and radio buttons inline */
    margin-right: 10px; /* Add space between radio buttons */
}
</style>
</head>
<body>
<div class="container">
<div id="form-container">
<!-- Login Form -->
<div id="login-form">
    <h1 class="text-2xl font-bold mb-4">Welcome Back to Comic Book
Registration</h1>
    <form onsubmit="return login(event)">
        <div class="mb-4">
            <label for="login-username" class="block text-sm font-medium text-
gray-700">Username</label>
            <input type="text" id="login-username" required class="mt-1 block w-
full border border-gray-300 rounded-md p-2" />
        </div>
        <div class="mb-4">
            <label for="login-password" class="block text-sm font-medium text-
gray-700">Password</label>
            <input type="password" id="login-password" required class="mt-1
block w-full border border-gray-300 rounded-md p-2" />
        </div>
        <br>
        <button type="submit" class="bg-blue-500 text-white py-2 rounded w-full
hover:bg-blue-600">Login</button>
        <p class="mt-4 text-center">Don't have an account? <a href="#">
        onclick="toggleForms()" class="text-blue-500">Sign Up</a></p>
    </form>
</div>

<!-- Signup Form -->
```

Rollno: 22AD002

Name: ABINAYA P

```
<div id="signup-form" class="hidden">
  <h1 class="text-2xl font-bold mb-4">Comic Book Registration</h1>
  <form onsubmit="return signup(event)">
    <div class="mb-4">
      <label for="signup-username" class="block text-sm font-medium text-gray-700">Username</label>
      <input type="text" id="signup-username" required class="mt-1 block w-full border border-gray-300 rounded-md p-2" />
    </div>
    <div class="mb-4">
      <label for="signup-email" class="block text-sm font-medium text-gray-700">Email</label>
      <input type="email" id="signup-email" required class="mt-1 block w-full border border-gray-300 rounded-md p-2" />
    </div>
    <div class="mb-4">
      <label for="signup-password" class="block text-sm font-medium text-gray-700">Password</label>
      <input type="password" id="signup-password" required class="mt-1 block w-full border border-gray-300 rounded-md p-2" />
    </div>
    <div class="mb-4">
      <label for="signup-age" class="block text-sm font-medium text-gray-700">Age</label>
      <input type="number" id="signup-age" required class="mt-1 block w-full border border-gray-300 rounded-md p-2" />
    </div>
    <div class="mb-4">
      <label for="gender" class="block text-sm font-medium text-gray-700">Gender</label>
      <select id="gender" name="gender" required class="mt-1 block w-full border border-gray-300 rounded-md p-2">
        <option value="select">Select</option>
        <option value="male">Male</option>
        <option value="female">Female</option>
        <option value="other">Other</option>
      </select>
    </div>
    <div class="mb-4">
      <label for="bookName" class="block text-sm font-medium text-gray-700">Book Name</label>
      <input type="text" id="bookName" name="bookName" required class="mt-1 block w-full border border-gray-300 rounded-md p-2" />
    </div>
    <div class="mb-4">
      <label for="author" class="block text-sm font-medium text-gray-700">Author</label>
```

Rollno: 22AD002

Name: ABINAYA P

```
<input type="text" id="author" name="author" required class="mt-1  
block w-full border border-gray-300 rounded-md p-2" />  
</div>  
<div class="mb-4">  
    <label for="format" class="block text-sm font-medium text-gray-  
700">Preferred Book Format</label>  
    <select id="format" name="format" required class="mt-1 block w-full  
border border-gray-300 rounded-md p-2">  
        <option value="select">Select</option>  
        <option value="pdf">PDF</option>  
        <option value="epub">EPUB</option>  
        <option value="mobi">MOBI</option>  
    </select>  
</div>  
<p>Select Your Role</p>  
<div class="radio-group">  
    <label>  
        <input type="radio" name="role" value="reader">Reader  
    </label>  
    <label>  
        <input type="radio" name="role" value="creator">Creator  
    </label>  
</div>
```

```
<button type="submit" class="bg-green-500 text-white py-2 rounded w-  
full hover:bg-green-600">Sign Up</button>  
<p class="mt-4 text-center">Already have an account? <a href="#"  
onclick="toggleForms()" class="text-blue-500">Login</a></p>  
</form>  
</div>  
</div>  
</div>  
<script src="script.js"></script>  
</body>  
</html>
```

- **Script.js**

```
function toggleForms() {  
    const loginForm = document.getElementById('login-form');  
    const signupForm = document.getElementById('signup-form');  
    loginForm.classList.toggle('hidden');  
    signupForm.classList.toggle('hidden');
```

Rollno: 22AD002

Name: ABINAYA P

```
}

function login(event) {
    event.preventDefault();
    const username = document.getElementById('login-username').value;
    const password = document.getElementById('login-password').value;

    // Simple validation (you can add more complex validation)
    if (username && password) {
        alert(`Login successful!\nUsername: ${username}`);
        // Here you can add code to send data to your server
    } else {
        alert('Please fill in all fields.');
    }
    return false; // Prevent form submission
}

function signup(event) {
    event.preventDefault();
    const username = document.getElementById('signup-username').value;
    const email = document.getElementById('signup-email').value;
    const password = document.getElementById('signup-password').value;

    // Regular expression for email validation
    const emailRegex = /^[^@\s]+@[^\s@]+\.\[^@\s@]+\$/;
    if (!emailRegex.test(email)) {
        alert('Please enter a valid email address.');
        return false;
    }

    // Regular expression for password validation
    const passwordRegex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{6,}$/;
    if (!passwordRegex.test(password)) {
        alert('Password must be at least 6 characters long, contain at least one uppercase letter, one lowercase letter, one digit, and one special character.');
        return false;
    }

    alert(`Registration successful! ${username}`);
    // Here you can add code to send data to your server
    return false; // Prevent form submission
}
```

Rollno: 22AD002

Name: ABINAYA P

- **Style.css**

```
body {  
    background-color: #f8d3e0; /* Light pink background */  
}  
  
.container {  
    max-width: 400px;  
    margin: auto;  
    padding: 20px;  
    background: white;  
    border-radius: 10px;  
    box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);  
}  
  
.hidden {  
    display: none;  
}  
  
/* General styles for radio buttons */  
input[type="radio"] {  
    vertical-align: middle; /* Aligns radio buttons with text vertically */  
    margin-right: 0; /* Space between radio button and label text */  
}  
  
/* Container for radio buttons to ensure alignment */  
.radio-group {  
    display: flex;  
    flex-direction: column; /* Align radio buttons vertically */  
    align-items: flex-start; /* Align radio buttons to the left */  
    margin-bottom: 15px; /* Space below the radio button group */  
}  
  
/* Style for label to ensure proper spacing */  
label {  
    display: flex;  
    align-items: center; /* Aligns radio button and text vertically */  
    margin-bottom: 10px;  
    white-space: nowrap; /* Space below each radio button option */  
}  
  
  
body {  
    font-family: Arial, sans-serif;  
    margin: 0;  
    padding: 0;
```

Rollno: 22AD002

Name: ABINAYA P

```
display: flex;
justify-content: center;
align-items: center;
min-height: 100vh;
background: url('pexels-niezam-16516.jpg') no-repeat center center fixed;
background-size: cover;
}

.container {
background: rgba(255, 255, 255, 0.8);
border-radius: 8px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
padding: 20px;
width: 100%;
max-width: 600px;
}

h1 {
text-align: center;
margin-bottom: 20px;
}

.form-group {
margin-bottom: 15px;
}

label {
display: block;
margin-bottom: 5px;
}

input, select {
width: 100%;
padding: 8px;
box-sizing: border-box;
}

button {
display: block;
width: 100%;
padding: 10px;
background-color: #666;
color: #fff;
border: none;
border-radius: 4px;
font-size: 16px;
```

Rollno: 22AD002

Name: ABINAYA P

```
        cursor: pointer;
    }

button:hover {
    background-color: #666;
}

/* Modal styles */
.modal {
    display: none;
    position: fixed;
    z-index: 1;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    overflow: auto;
    background-color: rgba(0,0,0,0.4);
}

.modal-content {
    background-color: #fefefe;
    margin: 15% auto;
    padding: 20px;
    border: 1px solid #888;
    width: 80%;
    max-width: 400px;
    text-align: center;
    border-radius: 8px;
}

.close {
    color: #aaa;
    float: right;
    font-size: 28px;
    font-weight: bold;
    cursor: pointer;
}

.close:hover,
.close:focus {
    color: black;
    text-decoration: none;
    cursor: pointer;
}
```

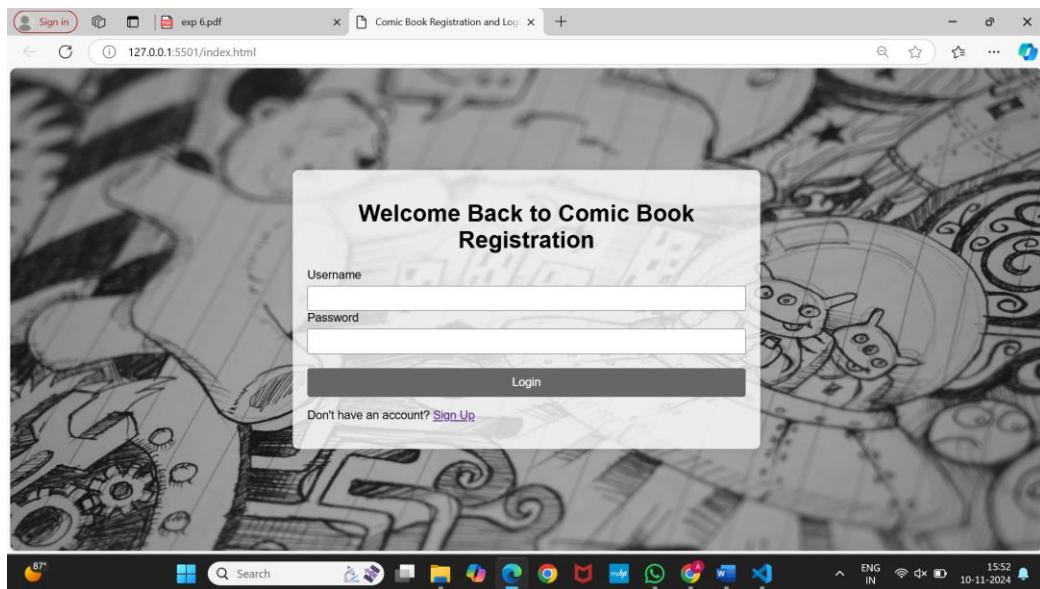
Rollno: 22AD002

Name: ABINAYA P

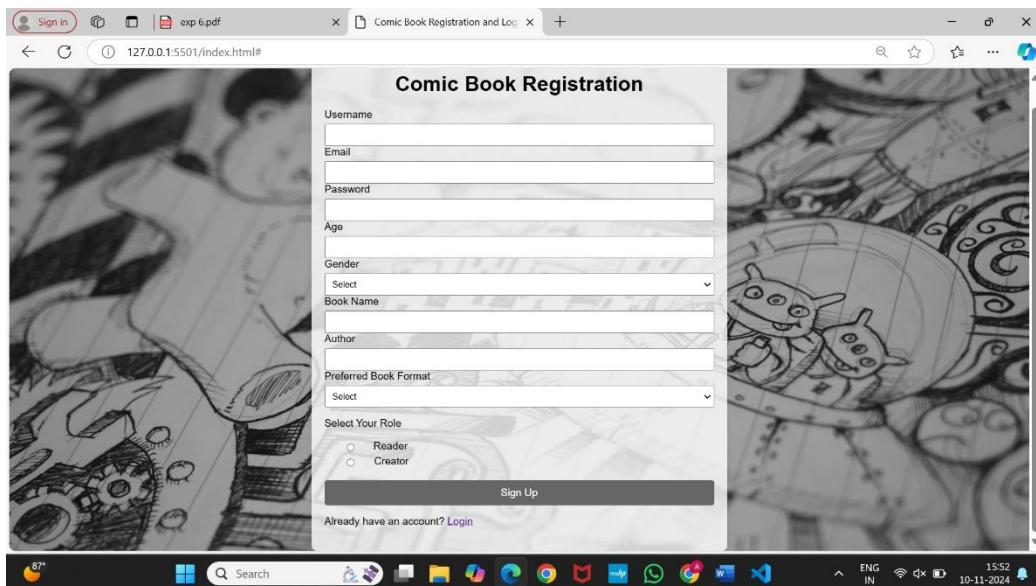
**Output:**

URL : <https://abi-1724.github.io/comic-book-form/>

**Screen 1 : Login Page**



**Screen 2 : Sign Up page**



**Exp No: 3**

**Develop an application for task management using Java Script and convert it to mobile app**

**Objective:**

The objective of the code is to create a simple task management web app that allows users to add, complete, or delete tasks, track progress, and organize tasks by their due dates.

**Procedure :**

- HTML Structure: Defines the layout for task management, including task input, lists, and progress tracking.
- CSS Styling: Styles the page with a transparent task container, responsive layout, and interactive buttons.
- Add Task Functionality: Captures task input and due date, then displays it in the appropriate list (current or later).
- Progress Bar: Tracks task completion progress, updating the progress bar and percentage based on completed tasks.
- Task Completion & Deletion: Allows users to mark tasks as complete or delete them from the list.
- Event Listeners: Adds interactivity to buttons for task management actions like adding, completing, or deleting tasks.

**Complete Source code :**

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
  <title>Task Management</title>
</head>
<body class="bg-gray-100 p-6">
```

Rollno: 22AD002

Name: ABINAYA P

```
<style>
body {
    font-family: 'Arial', sans-serif;
    background-image: url('pexels-polina-kovaleva-5546907.jpg'); /* Replace with
your image URL */
    background-size: cover; /* Cover the entire background */
    background-position: center; /* Center the background image */
    background-repeat: no-repeat; /* Prevent repeating the image */
}

.container {
    max-width: 600px;
    padding: 20px;
    background-color: rgba(255, 255, 255, 0.8); /* White background with
transparency */
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Subtle shadow */
    margin: 0; /* No automatic centering */
}

h1, h2 {
    color: #2d3748; /* Dark gray for headings */
}

input[type="text"], input[type="date"] {
    width: calc(100% - 40px); /* Full width minus button space */
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #cbd5e0; /* Light gray border */
    border-radius: 4px;
    transition: border-color 0.3s;
}

input[type="text"]:focus, input[type="date"]:focus {
    border-color: #8A816E; /* Blue border on focus */
    outline: none; /* Remove default outline */
}

button {
    background-color: #8A816E; /* Blue background */
    color: white; /* White text */
    padding: 10px 15px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s;
}
```

Rollno: 22AD002

Name: ABINAYA P

```
button:hover {  
    background-color: #8A816E; /* Darker blue on hover */  
}  
  
.task-item {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 10px;  
    border-bottom: 1px solid #e2e8f0; /* Light gray border */  
}  
  
.task-item span {  
    flex-grow: 1; /* Allow text to take available space */  
}  
  
.line-through {  
    text-decoration: line-through; /* Strikethrough for completed tasks */  
    color:#8A816E; /* Gray color for completed tasks */  
}  
  
.progress-container {  
    background-color: #edf2f7; /* Light gray for progress bar background */  
    border-radius: 4px;  
    overflow: hidden; /* Ensure rounded corners */  
}  
  
.progress-bar {  
    background-color: #816a37; /* Blue for progress */  
    height: 8px; /* Height of the progress bar */  
    transition: width 0.3s; /* Smooth transition for progress */  
}  
.input-container {  
    display: flex;  
    gap: 10px;  
}  
  
.input-container input {  
    flex-grow: 1;  
}  
.alert {  
    color: #8A816E; /* Red color for alert messages */  
    font-weight: bold;  
}  
#addTaskBtn {  
    background-color: #816a37; /* Blue background */
```

Rollno: 22AD002

Name: ABINAYA P

```
color: white; /* White text */
padding: 0%;
border: none;
border-radius: 10%; /* Round button */
border-top-left-radius: 50%;
border-top-right-radius: 50%;
border-bottom-right-radius: 50%;
border-bottom-left-radius: 50%;
width: 30px; /* Set fixed size for width */
height: 40px; /* Set fixed size for height */
display: flex;
justify-content: center;
align-items: center;
cursor: pointer;
transition: background-color 0.3s, transform 0.3s;
box-shadow: 0 2px 5px rgba(0, 0, 0, 0.2); /* Add subtle shadow */
}

/* Plus button hover effect */
#addTaskBtn:hover {
background-color: #816a37; /* Darker blue on hover */
transform: scale(1.1); /* Slight zoom on hover */
}

</style>
<div class="max-w-md bg-white rounded-lg shadow-md p-4 container">
<h1 class="text-xl font-bold mb-2">Tasks</h1>
<div class="mb-4">
<div class="relative">
<input type="text" id="taskInput" class="border border-gray-300 rounded-md p-2 w-full" placeholder="Add new item" />
<input type="date" id="dueDateInput" class="border border-gray-300 rounded-md p-2 w-full mt-2" />
<button id="addTaskBtn" class="absolute right-0 top-0 mt-2 mr-2 bg-blue-500 text-white rounded-md px-4">+</button>
</div>
</div>
<div class="mb-4">
<div class="flex items-center">
<div class="w-full bg-gray-200 rounded-full h-2">
<div id="progressBar" class="bg-blue-500 h-2 rounded-full" style="width: 0%;"></div>
<span id="progressText" class="ml-2 text-sm">0%</span>
</div>
</div>
<h2 class="text-lg font-semibold mb-2">Current Tasks</h2>
<ul id="taskList" class="space-y-2">
```

Rollno: 22AD002

Name: ABINAYA P

```
<!-- Current task items will be added here -->
</ul>
<h2 class="text-lg font-semibold mt-4 mb-2">Do Later</h2>
<ul id="doLaterList" class="space-y-2">
    <!-- Do later task items will be added here -->
</ul>
</div>

<script src="script.js"></script>
</body>
</html>
```

- **Script.js**

```
// Get DOM elements
const taskInput = document.getElementById('taskInput');
const dueDateInput = document.getElementById('dueDateInput');
const addTaskBtn = document.getElementById('addTaskBtn');
const taskList = document.getElementById('taskList');
const doLaterList = document.getElementById('doLaterList');
const progressBar = document.getElementById('progressBar');
const progressText = document.getElementById('progressText');

// Add event listener to the add task button
addTaskBtn.addEventListener('click', addTask);

function addTask() {
    const taskText = taskInput.value.trim();
    const dueDate = dueDateInput.value;

    // Check if both fields are filled
    if (taskText === "" || dueDate === "") {
        alert('Please fill in both the task and due date fields');
        return;
    }

    const taskItem = document.createElement('li');
    taskItem.className = 'task-item';
    taskItem.innerHTML = `
        <span>${taskText}</span>
        <span>${dueDate}</span>
        <button class="complete-btn">Complete</button>
        <button class="delete-btn">Delete</button>
    `;

    if (new Date(dueDate) > new Date()) {
        doLaterList.appendChild(taskItem);
    }
}
```

Rollno: 22AD002

Name: ABINAYA P

```
    } else {
      taskList.appendChild(taskItem);
    }

    taskInput.value = "";
    dueDateInput.value = "";

    updateProgress();
}

function updateProgress() {
  const totalTasks = taskList.children.length + doLaterList.children.length;
  const completedTasks = document.querySelectorAll('.task-item.completed').length;
  const progress = totalTasks === 0 ? 0 : (completedTasks / totalTasks) * 100;

  progressBar.style.width = `${progress}%`;
  progressText.textContent = `${Math.round(progress)}%`;
}

// Event delegation for complete and delete buttons
document.addEventListener('click', function(e) {
  if (e.target.classList.contains('complete-btn')) {
    e.target.parentElement.classList.toggle('completed');
    updateProgress();
  } else if (e.target.classList.contains('delete-btn')) {
    e.target.parentElement.remove();
    updateProgress();
  }
});
```

- **Style.css**

```
<style>
body {
  font-family: 'Arial', sans-serif;
  background-image: url('pexels-polina-kovaleva-5546907.jpg'); /* Replace with
your image URL */

  background-size: cover; /* Cover the entire background */
  background-position: center; /* Center the background image */
  background-repeat: no-repeat; /* Prevent repeating the image */
}

.container {
  max-width: 600px;
  margin: 0 auto;
  padding: 20px;
```

Rollno: 22AD002

Name: ABINAYA P

```
background-color: rgba(255, 255, 255, 0.8); /* White background with transparency */
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Subtle shadow */
}

h1, h2 {
    color: #2d3748; /* Dark gray for headings */
}

input[type="text"], input[type="date"] {
    width: calc(100% - 40px); /* Full width minus button space */
    padding: 10px;
    margin-bottom: 10px;
    border: 1px solid #cbd5e0; /* Light gray border */
    border-radius: 4px;
    transition: border-color 0.3s;
}

input[type="text"]:focus, input[type="date"]:focus {
    border-color: #8A816E; /* Blue border on focus */
    outline: none; /* Remove default outline */
}

button {
    background-color: #8A816E; /* Blue background */
    color: white; /* White text */
    padding: 10px 15px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s;
}

button:hover {
    background-color: #8A816E; /* Darker blue on hover */
}

.task-item {
    display: flex;
    justify-content: space-between;
    align-items: center;
    padding: 10px;
    border-bottom: 1px solid #e2e8f0; /* Light gray border */
}

.task-item span {
```

Rollno: 22AD002

Name: ABINAYA P

```
flex-grow: 1; /* Allow text to take available space */  
}  
  
.line-through {  
    text-decoration: line-through; /* Strikethrough for completed tasks */  
    color: #8A816E; /* Gray color for completed tasks */  
}  
  
.progress-container {  
    background-color: #edf2f7; /* Light gray for progress bar background */  
    border-radius: 4px;  
    overflow: hidden; /* Ensure rounded corners */  
}  
  
.progress-bar {  
    background-color: #8A816E; /* Blue for progress */  
    height: 8px; /* Height of the progress bar */  
    transition: width 0.3s; /* Smooth transition for progress */  
}  
  
.alert {  
    color:#8A816E; /* Red color for alert messages */  
    font-weight: bold;  
}  
  
</style>
```

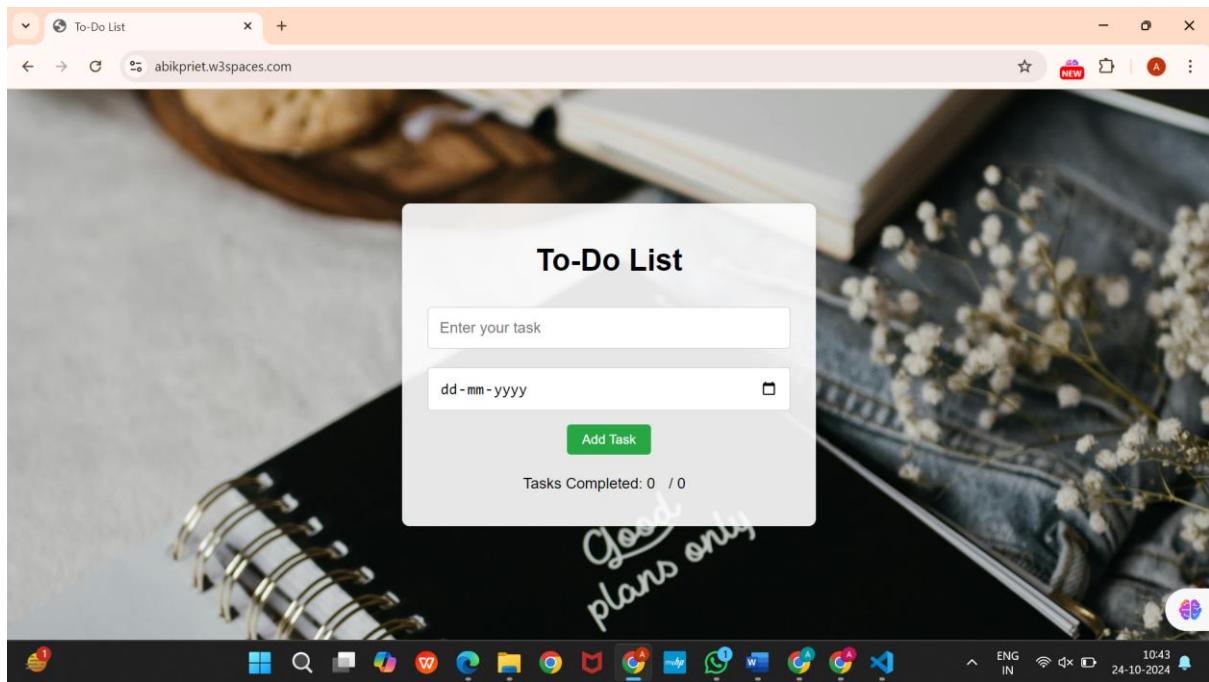
**Output:**

URL : <https://abikpriet.w3spaces.com/>

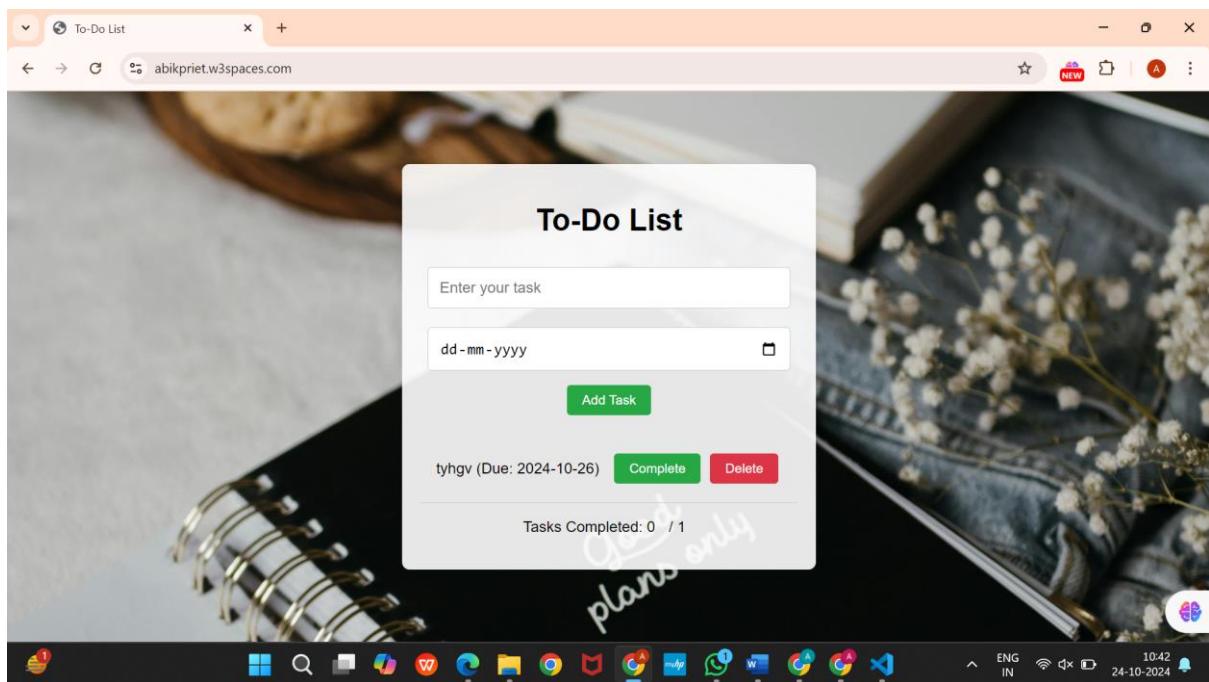
Rollno: 22AD002

Name: ABINAYA P

### Screen 1 : Home Page



### Screen 2 : Main page



Exp No: 4

**Develop a client application to make use of API services**

**Objective:**

The objective of the code is to create a currency exchange rate checker that allows users to input a base currency, select a target currency, and fetch the current exchange rate.

**Procedure :**

- **HTML Structure:** Defines the layout with input fields for base and target currencies and a button to fetch rates.
- **JavaScript Functionality:** Handles the button click event, validates user input, fetches exchange rates, and displays results.
- **Currency Validation:** Uses a regular expression to ensure the base currency input is a valid three-letter code.
- **Fetching Exchange Rate:** Constructs an API URL to retrieve exchange rates and handles responses or errors appropriately.
- **Displaying Results:** Updates the page with the exchange rate or an error message based on the user's input.

**Complete Source code :**

- index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Currency Exchange Rate Checker</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Currency Exchange Rate Checker</h1>
        <div class="input-section">
            <label for="baseCurrency">Base Currency:</label>
```

Rollno: 22AD002

Name: ABINAYA P

```
<input type="text" id="baseCurrency" placeholder="e.g., USD">

<label for="targetCurrency">Target Currency:</label>
<select id="targetCurrency">
    <option value="EUR">EUR</option>
    <option value="GBP">GBP</option>
    <option value="JPY">JPY</option>
    <option value="INR">INR</option>
    <option value="CAD">CAD</option>
</select>

    <button id="getRate">Get Exchange Rate</button>
</div>
<div class="result-section">
    <h2>Exchange Rate</h2>
    <p id="rateDisplay">-</p>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```

- **Script.js**

```
// script.js
document.getElementById('getRate').addEventListener('click', function () {
    const baseCurrency =
document.getElementById('baseCurrency').value.toUpperCase();
    const targetCurrency = document.getElementById('targetCurrency').value;

    // Regular expression to check if the base currency is a valid format (e.g., 3
uppercase letters)
    const currencyPattern = /^[A-Z]{3}$/;

    // Validate base currency input
    if (!currencyPattern.test(baseCurrency)) {
        alert("Please enter a valid 3-letter currency code (e.g., USD, EUR).");
        document.getElementById('rateDisplay').textContent = ""; // Clear previous
display
        return; // Stop further execution
    }

    const apiUrl = `https://api.exchangerate-api.com/v4/latest/${baseCurrency}`;

    fetch(apiUrl)
        .then(response => {
            if (!response.ok) {
```

Rollno: 22AD002

Name: ABINAYA P

```
        throw new Error(`Failed to fetch data for ${baseCurrency}`);
    }
    return response.json();
})
.then(data => {
    const rate = data.rates[targetCurrency];
    if (rate) {
        document.getElementById('rateDisplay').textContent = `1 ${baseCurrency} =
${rate} ${targetCurrency}`;
    } else {
        document.getElementById('rateDisplay').textContent = `Rate not available
for ${targetCurrency}`;
    }
})
.catch(error => {
    document.getElementById('rateDisplay').textContent = error.message;
});
});
```

- **Style.css**

```
/* style.css */

body {
    font-family: Arial, sans-serif;
    background-image: url('pexels-pixabay-259249.jpg'); /* Change this path as
needed */
    background-size: cover; /* Cover the entire viewport */
    background-position: center; /* Center the background image */
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;
}

.container {
    background-color: rgba(255, 255, 255, 0.8); /* Slight transparency for better
readability */
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    width: 300px;
    text-align: center;
}

h1 {
```

Rollno: 22AD002

Name: ABINAYA P

```
        color: #333;
    }

    .input-section {
        margin-bottom: 20px;
    }

    label {
        display: block;
        margin: 10px 0 5px;
    }

    input, select {
        width: 100%;
        padding: 8px;
        margin-bottom: 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
    }

    button {
        width: 100%;
        padding: 10px;
        background-color: #007BFF;
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        font-size: 16px;
    }

    button:hover {
        background-color: #0056b3;
    }

    .result-section {
        margin-top: 20px;
    }

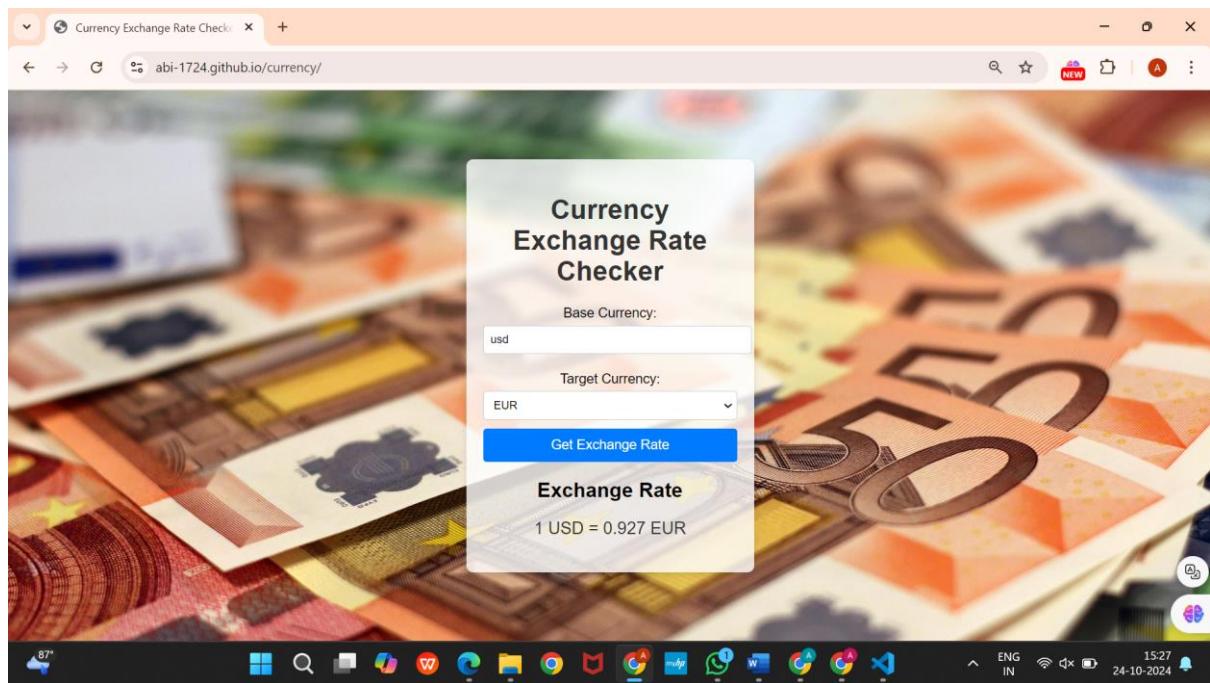
    #rateDisplay {
        font-size: 20px;
        color: #333;
    }
```

**Output:**

URL : <https://abi-1724.github.io/currency/>

Rollno: 22AD002

Name: ABINAYA P



**Exp No: 5**

**Demonstrate a basic web server and file operations with Node.js**

**Objective:**

This code creates an HTTP server that provides file operations (read, write, delete, list) on example.txt through an interactive HTML interface with buttons for each action.

**Procedure :**

1. **Module Setup:** Imports necessary modules (http, fs, path, url) and defines the target file (example.txt) and the working directory.
2. **Response Handling:** Defines a sendResponse function to streamline sending HTTP responses with appropriate status codes and content types.
3. **Server Creation:** Sets up an HTTP server that listens on port 3000 and handles incoming requests based on the URL path.
4. **Route Handling:**
  - **/**: Serves an HTML interface with buttons for file operations (Read, Write, Delete, List) and includes JavaScript for interaction.
  - **/read**: Reads and returns the contents of example.txt.
  - **/write**: Accepts POST requests to write content to example.txt.
  - **/delete**: Deletes example.txt.
  - **/list**: Lists all files in the current directory.
  - **404**: Returns a "404 Not Found" message for invalid routes.
5. **Server Listening:** Starts the server and logs a message confirming it's running.

**Complete Source code :**

- index.js

```
const http = require('http');
const fs = require('fs');
const path = require('path');
const url = require('url');

// Define the filename to operate on
const filename = 'example.txt';
const directory = __dirname; // The current directory
```

Rollno: 22AD002

Name: ABINAYA P

```
// Function to send a response with status and message
function sendResponse(res, statusCode, message, contentType = 'text/plain') {
  res.writeHead(statusCode, { 'Content-Type': contentType });
  res.end(message);
}

// Create the server
http.createServer((req, res) => {
  const parsedUrl = url.parse(req.url, true);
  const pathname = parsedUrl.pathname;

  if (pathname === '/') {
    // Serve the main HTML page with buttons
    const html = `
      <!DOCTYPE html>
      <html lang="en">
      <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>File Operations</title>
        <style>
          /* Keyframes for animated background gradient */
          @keyframes gradientBackground {
            0% { background-position: 0% 50%; }
            50% { background-position: 100% 50%; }
            100% { background-position: 0% 50%; }
          }

          /* Keyframes for button pulsate effect */
          @keyframes pulsate {
            0% { box-shadow: 0 0 8px rgba(95, 186, 233, 0.5); }
            50% { box-shadow: 0 0 18px rgba(95, 186, 233, 1); }
            100% { box-shadow: 0 0 8px rgba(95, 186, 233, 0.5); }
          }

        </style>
      <body>
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        background: linear-gradient(135deg, #ff7e5f, #feb47b, #86a8e7, #91eae4);
        background-size: 200% 200%;
        animation: gradientBackground 10s ease infinite;
        color: #343a40;
        display: flex;
        flex-direction: column;
        align-items: center;
      </body>
    `;
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.end(html);
  }
})
```

Rollno: 22AD002

Name: ABINAYA P

```
        justify-content: center;
        height: 100vh;
    }

    h1 {
        color: #ffffff;
        font-size: 2.5em;
        margin-bottom: 20px;
        text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.2);
    }

    button {
        margin: 10px;
        padding: 12px 20px;
        font-size: 18px;
        cursor: pointer;
        background: linear-gradient(135deg, #56ccf2, #2f80ed);
        color: white;
        border: none;
        border-radius: 8px;
        animation: pulsate 2s infinite; /* Apply pulsate animation */
        transition: background 0.3s, transform 0.3s;
    }

    button:hover {
        background: linear-gradient(135deg, #2f80ed, #56ccf2);
        transform: scale(1.05); /* Slightly enlarge button on hover */
    }

    .result {
        margin-top: 20px;
        padding: 15px;
        width: 80%;
        max-width: 500px;
        background-color: rgba(255, 255, 255, 0.8);
        border: 1px solid #ddd;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
        color: #333;
        white-space: pre-wrap;
    }

    @media (max-width: 600px) {
        button {
            width: 100%;
        }
    }
```

Rollno: 22AD002

Name: ABINAYA P

```
</style>

</head>
<body>
    <h1>File Operations Server</h1>
    <button onclick="performOperation('read')">Read File</button>
    <button onclick="performOperation('write')">Write File</button>
    <button onclick="performOperation('delete')">Delete File</button>
    <button onclick="performOperation('list')">List Files</button>
    <div class="result" id="result"></div>
    <script>
        function performOperation(operation) {
            let url;
            if (operation === 'write') {
                const content = prompt("Enter content to write to example.txt:");
                if (!content) return; // Cancel if no content entered
                url = '/write';
                fetch(url, {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'text/plain',
                    },
                    body: content
                }).then(response => response.text()).then(data => {
                    document.getElementById('result').innerText = data;
                });
            } else {
                url = '/' + operation;
                fetch(url)
                    .then(response => response.text())
                    .then(data => {
                        document.getElementById('result').innerText = data;
                    });
            }
        }
    </script>
</body>
</html>
`;
sendResponse(res, 200, html, 'text/html');
} else if (pathname === '/read') {
    // Read the contents of example.txt
    fs.readFile(path.join(directory, filename), 'utf8', (err, data) => {
        if (err) {
            return sendResponse(res, 500, 'Error reading file: ' + err.message);
        }
        sendResponse(res, 200, 'File Contents:\n' + data);
    })
}
```

Rollno: 22AD002

Name: ABINAYA P

```
});  
} else if (pathname === '/write' && req.method === 'POST') {  
    // Write content to example.txt  
    let body = '';  
    req.on('data', chunk => {  
        body += chunk.toString();  
    });  
    req.on('end', () => {  
        fs.writeFile(path.join(directory, filename), body, (err) => {  
            if (err) {  
                return sendResponse(res, 500, 'Error writing to file: ' + err.message);  
            }  
            sendResponse(res, 200, 'File written successfully!');  
        });  
    });  
} else if (pathname === '/delete') {  
    // Delete example.txt  
    fs.unlink(path.join(directory, filename), (err) => {  
        if (err) {  
            return sendResponse(res, 500, 'Error deleting file: ' + err.message);  
        }  
        sendResponse(res, 200, 'File deleted successfully!');  
    });  
} else if (pathname === '/list') {  
    // List all files in the directory  
    fs.readdir(directory, (err, files) => {  
        if (err) {  
            return sendResponse(res, 500, 'Error reading directory: ' + err.message);  
        }  
        sendResponse(res, 200, 'Files in directory:\n' + files.join('\n'));  
    });  
} else {  
    // Handle 404 - Not Found  
    sendResponse(res, 404, '404 Not Found');  
}  
}).listen(3000, () => {  
    console.log('Server running on http://localhost:3000');  
});
```

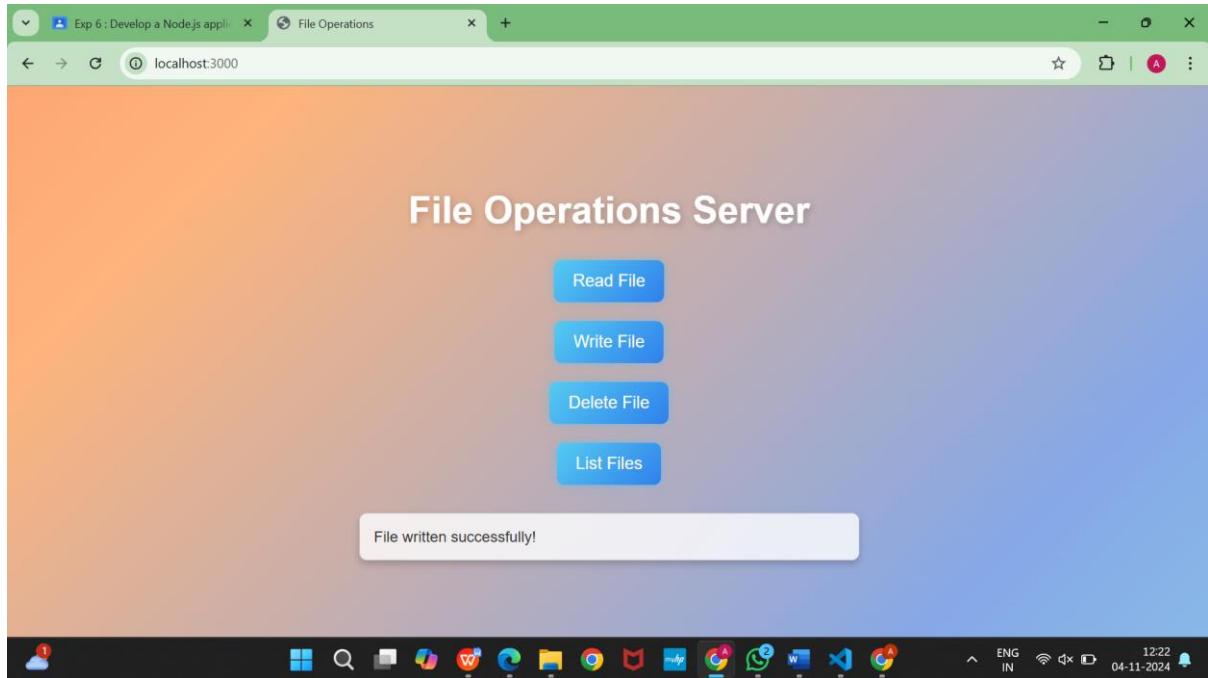
### Output:

URL : [https://abi-1724.github.io/server\\_fileoperation/](https://abi-1724.github.io/server_fileoperation/)

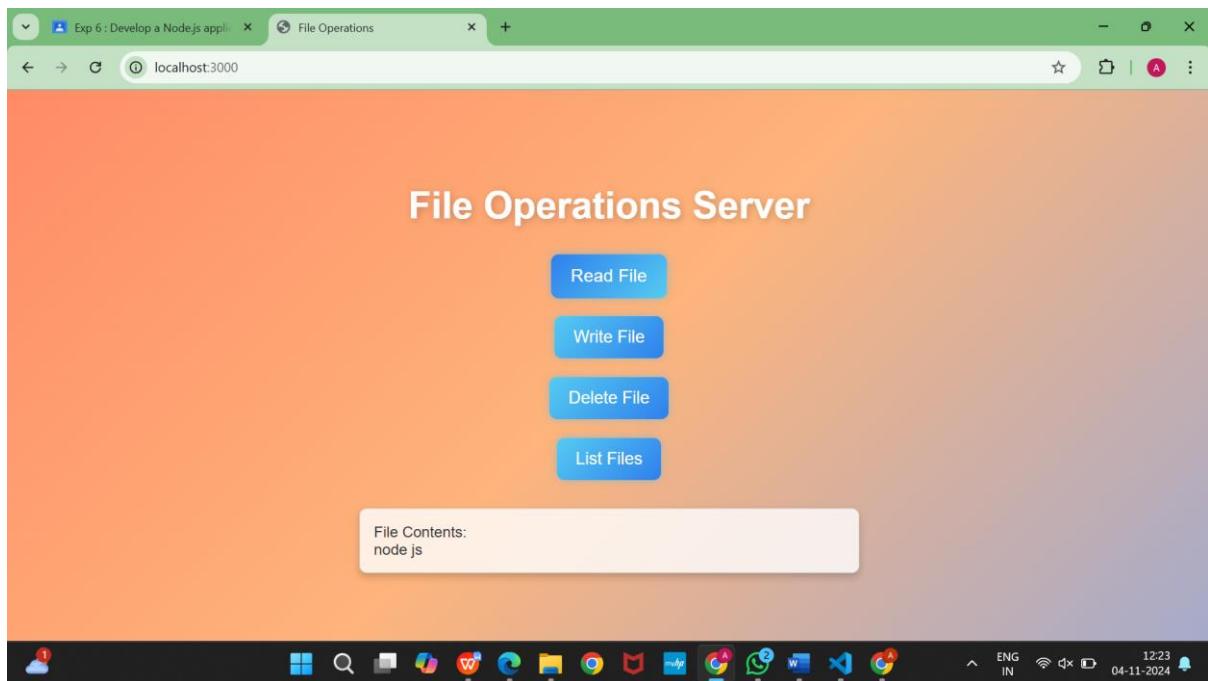
Rollno: 22AD002

Name: ABINAYA P

### File Writing :



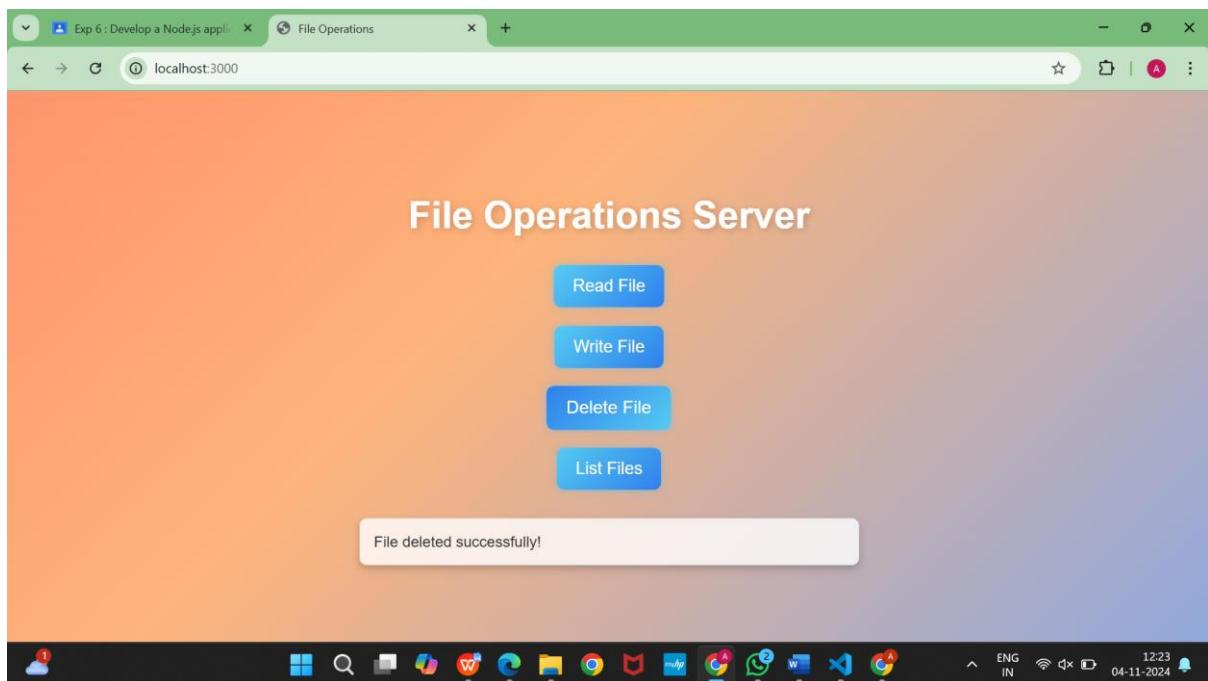
### File Reading



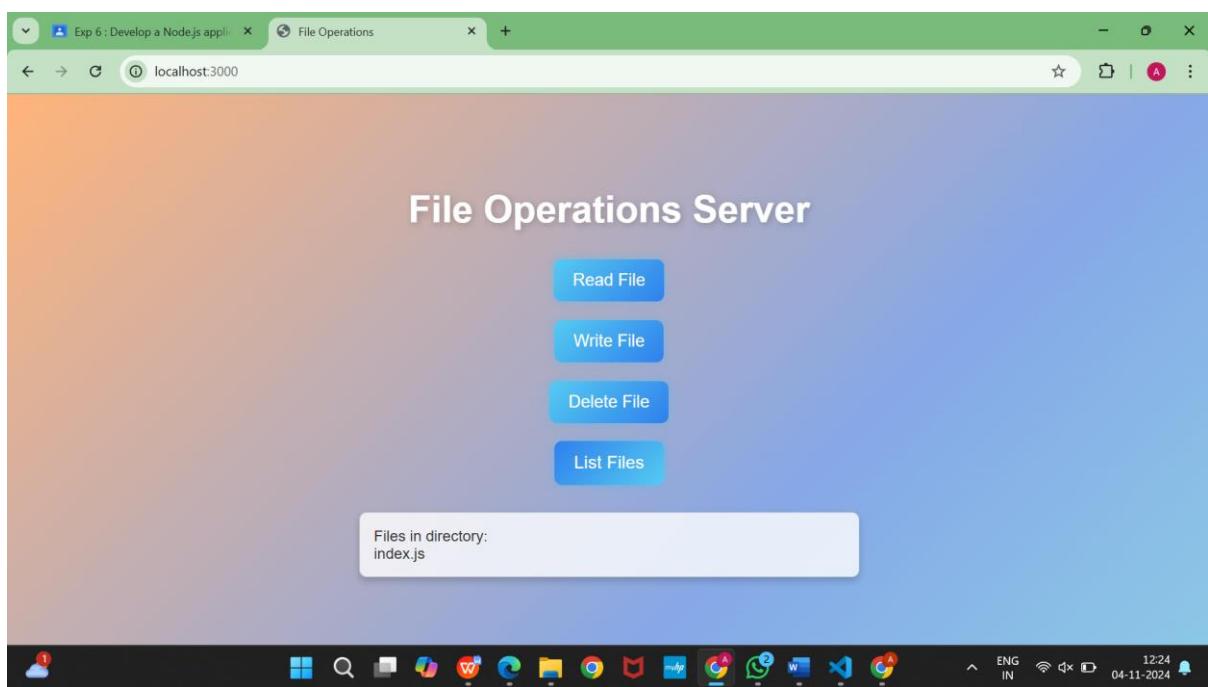
Rollno: 22AD002

Name: ABINAYA P

### File Delete



### List of files



<b>Exp No: 06</b>	<b>Develop a Node.js application that serves static JSON data via API</b>
-------------------	---

### **Objective:**

The objective is to build a Node.js application that serves static JSON data via a REST API endpoint using Express. The API should handle HTTP GET requests and respond with the JSON data in the proper format. This setup will allow clients to retrieve the data seamlessly over HTTP.

### **Procedure :**

#### **Set Up Node.js Environment:**

Install Node.js if it's not already installed on your computer.

Initialize a new Node.js project by creating a folder and running `npm init -y` to generate a `package.json` file.

#### **Install Required Dependencies:**

Use the Express framework to create the server.

In the terminal, navigate to your project folder and install Express by running  
`npm install express`

#### **Create Project Files:**

Inside project directory, create the following files:

`server.js` (for the main server code).

`public/index.html` (for a simple HTML page).

`public/style.css` (for styling the HTML page).

#### **Start the Server:**

Run the server with the command:

`node server.js`

Open a browser and navigate to `http://localhost:3000/api/data` to see the JSON data response.

#### **Verify the Output:**

Check if the JSON data is served correctly via the `/api/data` endpoint.

Confirm that the HTML page (at `http://localhost:3000/`) is styled as expected.

### **Complete Source code :**

- `Public/styles.css`
- `Server.js`

#### **Server.js:**

```
const express = require('express');
const path = require('path');
const fs = require('fs');
```

```
// Create an Express application
```

Rollno: 22AD002

Name: ABINAYA P

```
const app = express();

// Define the port the server will run on
const PORT = process.env.PORT || 3000;

// Define the path to the JSON data file
const booksDataPath = path.join(__dirname, 'books.json');

// Serve static files (CSS) from the 'public' directory
app.use(express.static(path.join(__dirname, 'public')));

// Route for the root URL ('/')
app.get('/', (req, res) => {
  res.send(`

    <!DOCTYPE html>
    <html lang="en">
      <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Book Store API</title>
        <link rel="stylesheet" href="/styles.css">
      </head>
      <body>
        <div class="container">
          <h1>Welcome to the Book Store API!</h1>
          <p>Use <code>/api/books</code> to get all books or <code>/api/books/:id</code> to
get a specific book.</p>
        </div>
      </body>
    </html>
  `);
});

// API endpoint to fetch all books
app.get('/api/books', (req, res) => {
  fs.readFile(booksDataPath, 'utf8', (err, data) => {
    if (err) {
      return res.status(500).json({ error: 'Unable to read books data' });
    }
    res.json(JSON.parse(data));
  });
});

// API endpoint to fetch a specific book by ID
app.get('/api/books/:id', (req, res) => {
  const bookId = parseInt(req.params.id, 10);
```

Rollno: 22AD002

Name: ABINAYA P

```
fs.readFile(booksDataPath, 'utf8', (err, data) => {
  if (err) {
    return res.status(500).json({ error: 'Unable to read books data' });
  }

  const books = JSON.parse(data);
  const book = books.find(b => b.id === bookId);

  if (!book) {
    return res.status(404).json({ error: 'Book not found' });
  }

  res.json(book);
});

});

// Start the server
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

#### **Public/styles.css:**

```
/* Midnight Teal and Soft Charcoal Style */
body {
  background: linear-gradient(135deg, #1a2a36, #4a6572); /* Midnight teal to soft charcoal
*/
  color: #e1e1e1;
  font-family: 'Arial', sans-serif;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  max-width: 600px;
  padding: 20px;
  border-radius: 10px;
  background: linear-gradient(135deg, #4a6572, #3c4f56); /* Soft charcoal gradient for
container */
  box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.4);
  text-align: center;
}

h1 {
```

```

Rollno: 22AD002
Name: ABINAYA P

font-size: 2em;
color: #ffffff;
margin-bottom: 10px;
}

p {
font-size: 1.2em;
color: #b0c4b1; /* Soft sage green accent */
}

code {
background-color: rgba(255, 255, 255, 0.1); /* Semi-transparent white overlay */
color: #e1e1e1;
padding: 2px 5px;
border-radius: 5px;
font-size: 1em;
}

```

### **Explanation for each file:**

#### **1. Server.js**

- **App Initialization & Configuration:** Sets up an Express server, configures the port, and defines the path to the books.json data file.
- **Static and API Routes:** Serves static CSS files and sets up two main routes: the root route (/) displays an HTML welcome page, and API routes (/api/books and /api/books/:id) serve JSON data for all books or a specific book by ID.
- **Error Handling & Server Start:** Includes basic error handling for file reading issues and starts the server on the specified port, logging a confirmation message.

#### **2. Public/styles.css**

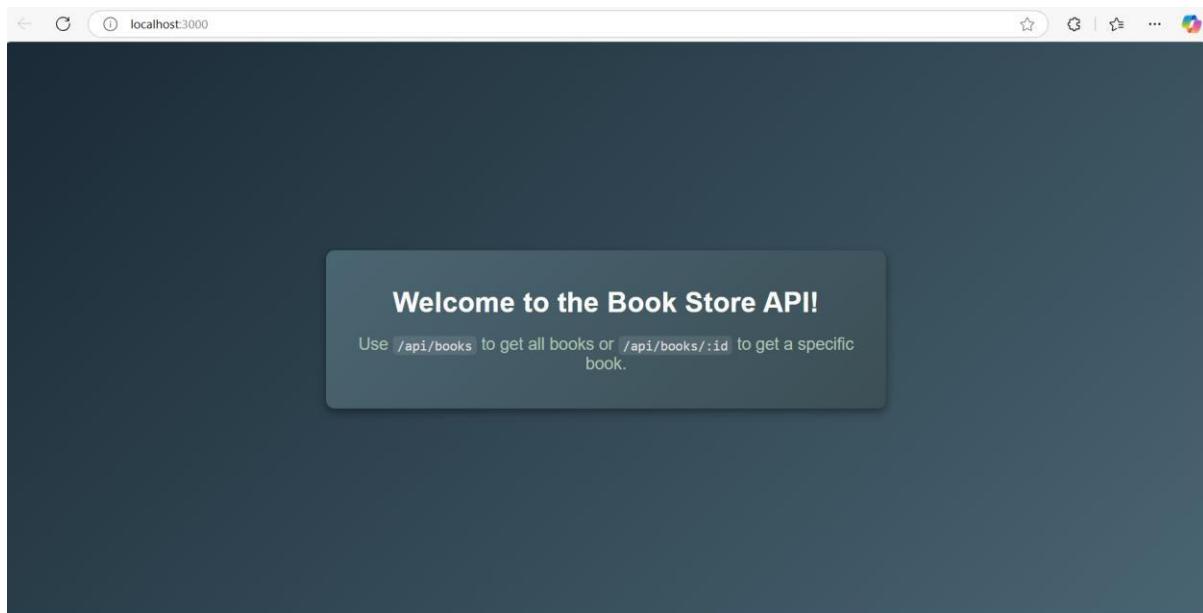
- **Background & Container Styling:** Sets a gradient background from midnight teal to soft charcoal, with a centered container styled with a complementary gradient and box shadow for a clean look.
- **Typography & Colors:** Applies a light font color for text, with larger, soft colors for headers (h1) and paragraph text (p), creating a professional look.
- **Code Block Styling:** Adds styling for <code> elements, giving them a semi-transparent overlay and padding, enhancing readability for instructions and code snippets on the HTML page.

### **Output:**

visit <http://localhost:3000/>, you should see a webpage with: A title "Welcome to the Book store API".

Rollno: 22AD002  
Name: ABINAYA P

### Screen 1 :



### Screen 2:

The screenshot shows a code editor interface (VS Code) with the following details:

- File Explorer:** Shows files in the 'book-store-api' directory, including 'books.json', '# styles.css', and 'server.js'.
- Editor:** The 'server.js' file is open, displaying the following code:

```
49 app.get('/api/books/:id', (req, res) => {
50   fs.readFile(booksDataPath, 'utf8', (err, data) => {
51     if (err) {
52       return res.status(500).json({ error: 'Unable to read books data' });
53     }
54
55     const books = JSON.parse(data);
56     const book = books.find(b => b.id === bookId);
57
58     if (!book) {
59       return res.status(404).json({ error: 'Book not found' });
60     }
61
62     res.json(book);
63   });
64 });
65 });
66 });
67
68 // Start the server
69 app.listen(PORT, () => {
70   console.log(`Server is running on http://localhost:${PORT}`);
71 });
72 }
```

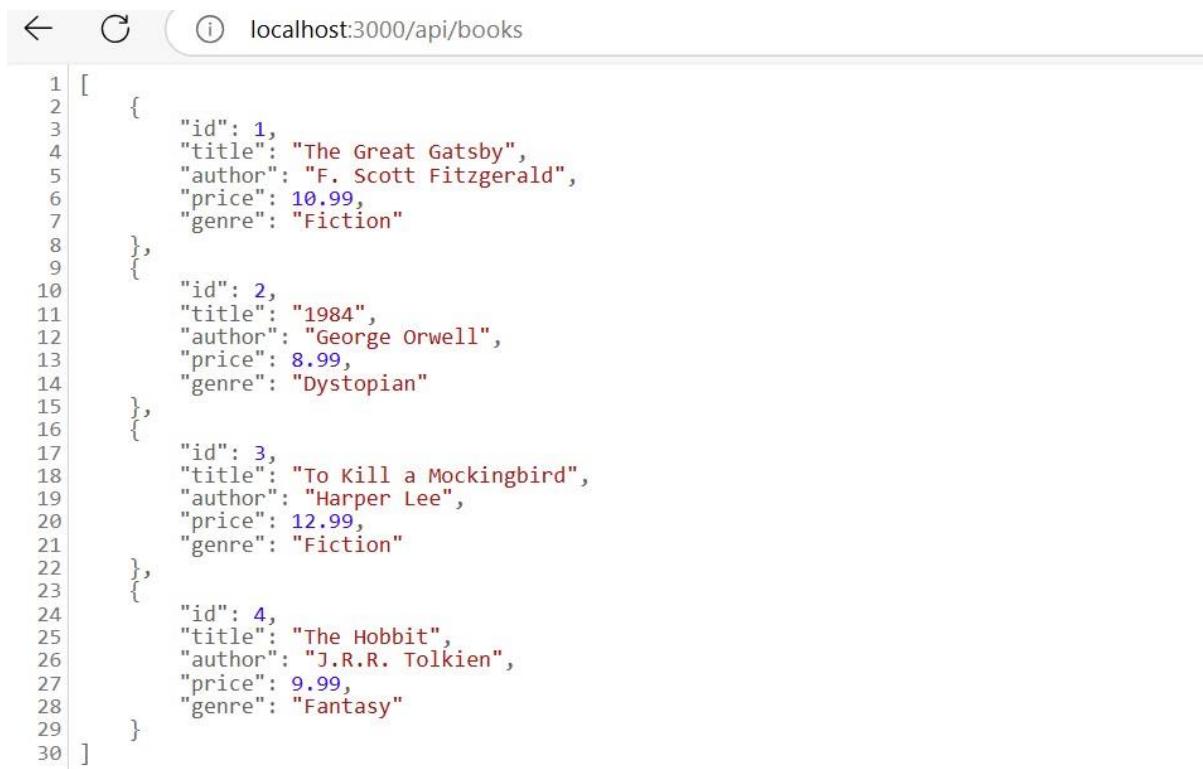
- Terminal:** The terminal window shows the command line output:

```
PS C:\Users\thirupathi\OneDrive\Desktop\html6> cd book-store-api
PS C:\Users\thirupathi\OneDrive\Desktop\html6\book-store-api> node server.js
PS C:\Users\thirupathi\OneDrive\Desktop\html6\book-store-api> node server.js
>>
server is running on http://localhost:3000
```

Rollno: 22AD002

Name: ABINAYA P

**Screen 3:**



```
← ⌂ ⓘ localhost:3000/api/books
[{"id": 1, "title": "The Great Gatsby", "author": "F. Scott Fitzgerald", "price": 10.99, "genre": "Fiction"}, {"id": 2, "title": "1984", "author": "George Orwell", "price": 8.99, "genre": "Dystopian"}, {"id": 3, "title": "To Kill a Mockingbird", "author": "Harper Lee", "price": 12.99, "genre": "Fiction"}, {"id": 4, "title": "The Hobbit", "author": "J.R.R. Tolkien", "price": 9.99, "genre": "Fantasy"}]
```

Rollno: 22AD002

Name: ABINAYA P

Exp No: 7

## Develop a quiz application using React.JS

### Objective:

The goal is to create a responsive quiz application using React.js that provides an engaging experience for users to test their knowledge on various topics. The app will feature multiple-choice questions, track the user's progress and score, and display the results upon completion. Key features will include a visually appealing UI, real-time feedback on answers, and a streak counter to encourage accuracy. The project aims to enhance React.js skills, specifically in state management, component design, and event handling.

### Procedure :

#### 1. Set Up the Project:

Begin by creating a new React project using `create-react-app`. Organize folders for components, styles, and data to maintain clean and modular code structure.

#### 2. Create Question Data:

Define an array of question objects, each containing a question, options, and the correct answer. This data will be used to render each question dynamically.

#### 3. Build Components:

Develop reusable components like `Question`, `Result`, and `ProgressBar` to handle different parts of the app. This modular approach makes the application easier to maintain and expand.

#### 4. Implement State Management:

Use React's `useState` hook to manage states such as the current question index, user's score, selected options, and streak count. State management is key to tracking user progress and interaction.

#### 5. Handle Answer Selection and Scoring:

Add logic to validate selected answers, update the score, and display immediate feedback. This includes highlighting correct/incorrect answers to improve the user experience.

#### 6. Display Streak and Progress Bar:

Implement a progress bar to show quiz completion status and a streak counter in the top-right to motivate users with a visual streak indicator, accompanied by an emoji.

#### 7. Show Results and Play Again Option:

Once the quiz is complete, display the final score with an option to play again, resetting states for a fresh attempt.

**Complete Source code :**

- App.js

```
import React, { useState } from 'react';
import Question from './Question';
import Result from './Result';
import VantaBackground from './VantaBackground';
import './index.css';

const quizData = [
  {
    question: "What is the capital of France?",
    options: ["Paris", "London", "Berlin", "Madrid"],
    answer: "Paris"
  },
  {
    question: "Who wrote 'To Kill a Mockingbird?'",
    options: ["Harper Lee", "J.K. Rowling", "Ernest Hemingway", "Jane Austen"],
    answer: "Harper Lee"
  },
  {
    question: "What is 2 + 2?",
    options: ["3", "4", "5", "6"],
    answer: "4"
  },
  {
    question: "What is the chemical symbol for water?",
    options: ["H2O", "CO2", "O2", "NaCl"],
    answer: "H2O"
  },
  {
    question: "Who painted the Mona Lisa?",
    options: ["Vincent van Gogh", "Leonardo da Vinci", "Pablo Picasso", "Claude Monet"],
    answer: "Leonardo da Vinci"
  },
  {
    question: "What planet is known as the Red Planet?",
    options: ["Earth", "Mars", "Jupiter", "Saturn"],
    answer: "Mars"
  },
  {
    question: "What is the largest mammal?",
    options: ["Elephant", "Blue Whale", "Giraffe", "Shark"],
    answer: "Blue Whale"
  }
]
```

Rollno: 22AD002

Name: ABINAYA P

```
        answer: "Blue Whale"
    },
    {
        question: "What is the smallest prime number?",
        options: ["0", "1", "2", "3"],
        answer: "2"
    },
    {
        question: "Who was the first President of the United States?",
        options: ["Abraham Lincoln", "George Washington", "Thomas Jefferson", "John Adams"],
        answer: "George Washington"
    },
    {
        question: "What is the square root of 64?",
        options: ["6", "7", "8", "9"],
        answer: "8"
    }
];
}

function App() {
    const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0);
    const [score, setScore] = useState(0);
    const [showResult, setShowResult] = useState(false);
    const [selectedOption, setSelectedOption] = useState(null);
    const [showAnswer, setShowAnswer] = useState(false);

    const handleAnswerSelection = (option) => {
        setSelectedOption(option);
        setShowAnswer(true);
        if (option === quizData[currentQuestionIndex].answer) {
            setScore(score + 1);
        }
    };

    const handleNextQuestion = () => {
        setShowAnswer(false);
        setSelectedOption(null);
        if (currentQuestionIndex < quizData.length - 1) {
            setCurrentQuestionIndex(currentQuestionIndex + 1);
        } else {
            setShowResult(true);
        }
    };

    const handlePlayAgain = () => {
        setCurrentQuestionIndex(0);
    };
}
```

Rollno: 22AD002

Name: ABINAYA P

```
        setScore(0);
        setShowResult(false);
        setSelectedOption(null);
        setShowAnswer(false);
    };

    return (
        <VantaBackground>
            <div className="App">
                <div className="streak-count">Streak: {score} 🔥 </div>
                {showResult ? (
                    <Result score={score} totalQuestions={quizData.length}
                        onPlayAgain={handlePlayAgain} />
                ) : (
                    <Question
                        data={quizData[currentQuestionIndex]}
                        selectedOption={selectedOption}
                        showAnswer={showAnswer}
                        onAnswerSelect={handleAnswerSelection}
                        onNextQuestion={handleNextQuestion}
                    />
                )}
            <div className="progress-bar">
                <div
                    className="progress-bar-fill"
                    style={{
                        width: `${((currentQuestionIndex + 1) / quizData.length) * 100}%`,
                    }}
                ></div>
            </div>
        </VantaBackground>
    );
}

export default App;
```

- index.css

```
/* src/index.css */

.App {
    text-align: center;
    font-family: Arial, sans-serif;
```

Rollno: 22AD002

Name: ABINAYA P

```
display: flex;
flex-direction: column;
min-height: 100vh;
justify-content: center;
position: relative;
padding-bottom: 50px; /* Space above progress bar */
}

.options {
display: flex;
flex-direction: column;
align-items: center;
margin-top: 20px;
}

.option-btn {
padding: 10px 20px;
margin: 5px 0;
width: 60%;
font-size: 16px;
cursor: pointer;
border: none;
border-radius: 5px;
}

.option-btn.correct {
background-color: #905bf8;
color: white;
}

.option-btn.incorrect {
background-color: #f44336;
color: white;
}

.next-btn {
margin-top: 20px;
padding: 10px 20px;
font-size: 16px;
cursor: pointer;
border: none;
background-color: rgb(33, 150, 243);
color: white;
border-radius: 5px;
}

.result {
```

Rollno: 22AD002

Name: ABINAYA P

```
    font-size: 18px;
    margin-top: 20px;
}

.play-again-btn {
    margin-top: 20px;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    border: none;
    background-color: #4caf50;
    color: white;
    border-radius: 5px;
}

/* src/index.css */

.progress-bar {
    height: 15px;
    width: 75%; /* 75% width of the page */
    background: rgba(255, 215, 0, 0.3);
    border-radius: 8px;
    overflow: hidden;
    box-shadow: 0px 0px 10px rgba(255, 223, 0, 0.6);
    position: fixed;
    bottom: 20px;
    left: 50%;
    transform: translateX(-50%);
}

.progress-bar-fill {
    height: 100%;
    background: linear-gradient(90deg, #ffd700, #ffcc00, #ffe680);
    border-radius: 8px;
    animation: shimmer 1.5s infinite alternate ease-in-out;
}

.streak-count {
    font-size: 20px;
    font-weight: bold;
    color: #ffd700; /* Golden color for streak */
    position: absolute;
    top: 10px;
    right: 20px;
    display: flex;
    align-items: center;
    justify-content: center;
```

Rollno: 22AD002

Name: ABINAYA P

```
    gap: 8px; /* Space between streak number and emoji */  
}
```

```
/* Shimmer effect */  
@keyframes shimmer {  
  0% {  
    box-shadow: 0 0 8px #ffd700, 0 0 12px #ffec8b;  
  }  
  100% {  
    box-shadow: 0 0 8px #ffec8b, 0 0 12px #ffd700;  
  }  
}
```

- Questions.js

```
// src/Question.js  
import React from 'react';  
  
function Question({ data, selectedOption, showAnswer, onAnswerSelect, onNextQuestion }) {  
  return (  
    <div>  
      <h2>{data.question}</h2>  
      <div className="options">  
        {data.options.map((option, index) => (  
          <button  
            key={index}  
            onClick={() => onAnswerSelect(option)}  
            className={`option-btn ${  
              showAnswer  
                ? option === data.answer  
                ? 'correct'  
                : option === selectedOption  
                ? 'incorrect'  
                : ""  
                : ""  
            }`}  
            disabled={showAnswer}  
          >  
            {option}  
          </button>  
        ))}  
      </div>  
      {showAnswer && (  
        <button className="next-btn" onClick={onNextQuestion}>  
          Next Question  
        </button>  
      )}  
    </div>  
  );  
}
```

Rollno: 22AD002  
Name: ABINAYA P

```
        })
      </div>
    );
}

export default Question;
```

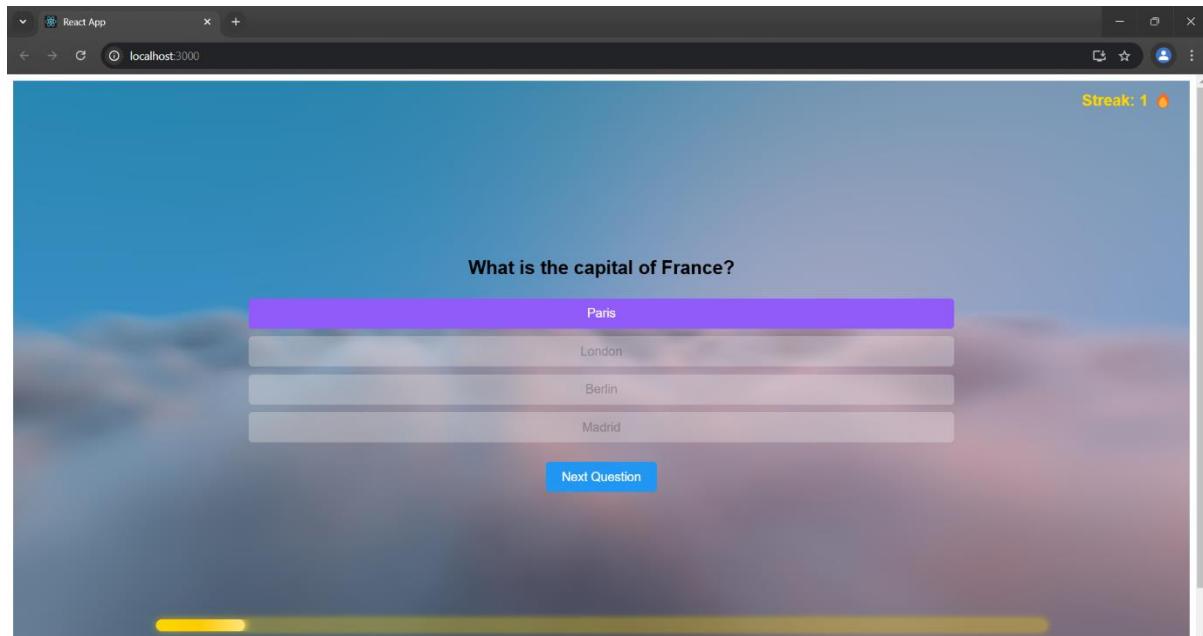
- Result.js

```
// src/Result.js
import React from 'react';

function Result({ score, totalQuestions, onPlayAgain }) {
  return (
    <div className="result">
      <h2>Quiz Completed!</h2>
      <p>
        You scored {score} out of {totalQuestions}.
      </p>
      <button className="play-again-btn" onClick={onPlayAgain}>
        Play Again
      </button>
    </div>
  );
}

export default Result;
```

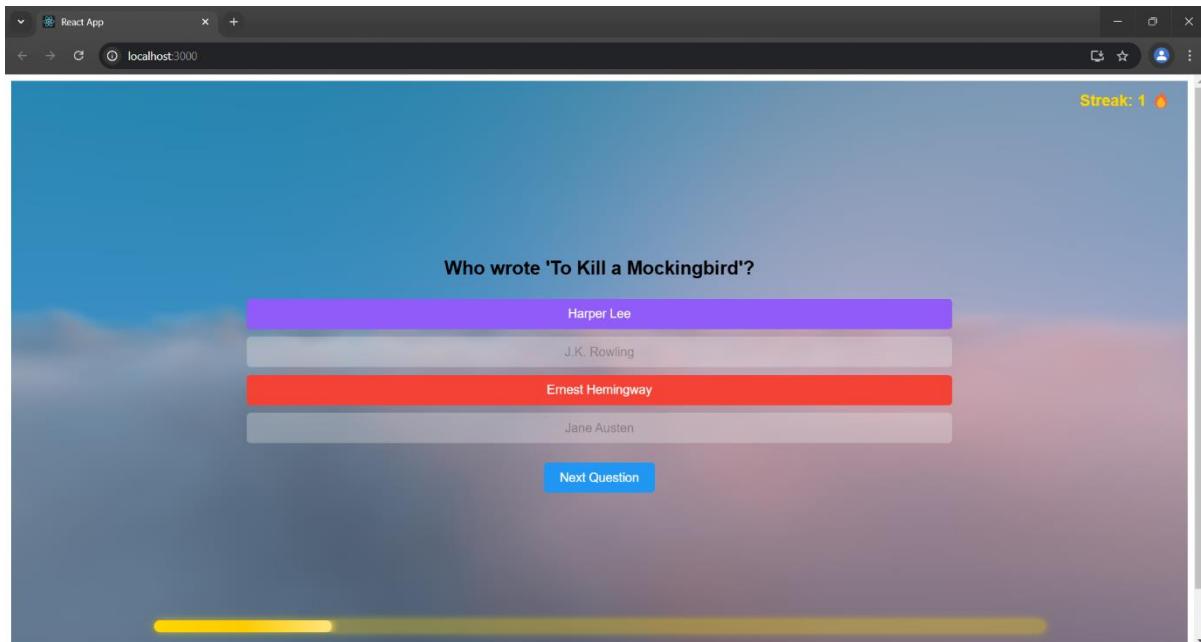
**Screen 1 : Home page**



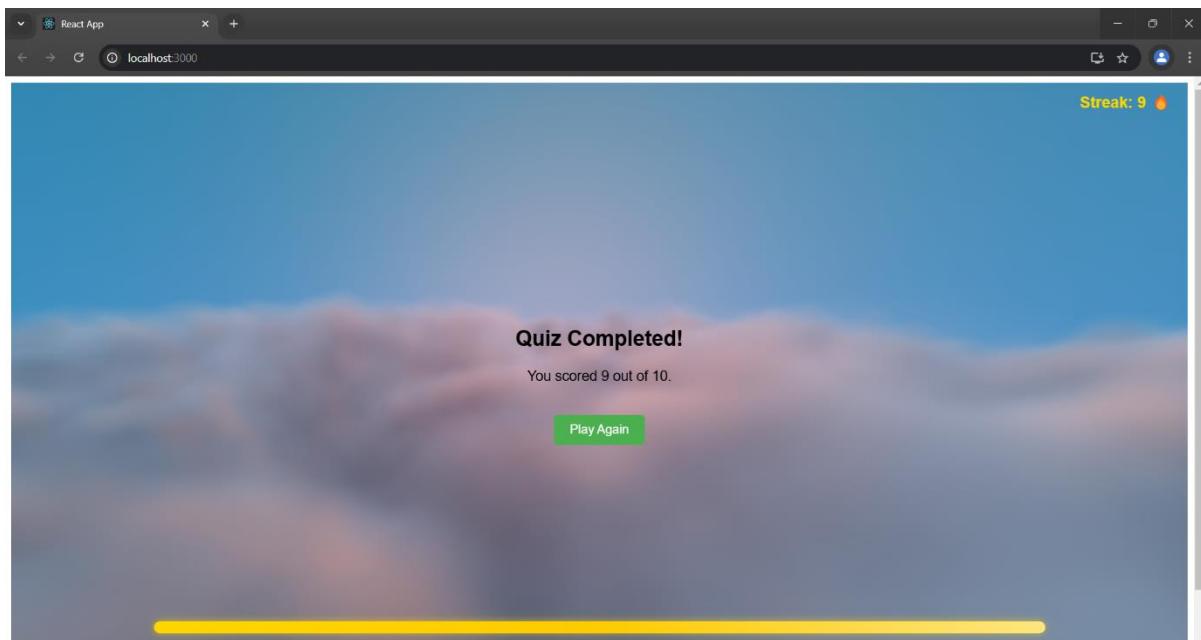
Rollno: 22AD002

Name: ABINAYA P

### Screen 2 : Incorrect answer



### Screen 3 : Play Again



**Exp No: 8**

**Develop a simple login and registration application using React.Js and MongoDB**

**Objective:**

The objective of this application is to develop a simple and user-friendly login and registration system using React.js for the frontend and MongoDB for data storage. It enables users to securely sign up with unique credentials, log in to access personalized content, and ensures data privacy with features like password hashing using bcrypt and token-based authentication via JWT. The system demonstrates effective client-server communication and a seamless user experience with form validations and intuitive navigation. It serves as a foundational project for learning full-stack development.

**Procedure :**

- **Set Up the Project Environment**

Initialize a React.js project using create-react-app for the frontend and configure a Node.js + Express.js backend. Install required dependencies like axios, mongoose, bcrypt, and jsonwebtoken.

- **Design the Frontend UI**

Create functional React components for Login, Signup, and Home pages. Use react-router-dom for navigation and CSS for styling to enhance the user experience.

- **Set Up MongoDB Database**

Install and configure MongoDB, either locally or via a cloud service like MongoDB Atlas. Define a User schema in MongoDB with fields for username and password.

- **Create Backend API Endpoints**

Develop API routes for user registration (/register) and login (/login) using Express.js. Use bcrypt to hash passwords during registration and validate them during login.

- **Implement Token-Based Authentication**

Use jsonwebtoken to generate a secure token upon successful login. This token can be used for protected routes and validating user sessions.

- **Connect Frontend with Backend**

Use axios in React components to send HTTP requests to backend APIs. Handle responses to update the UI dynamically based on user actions.

Rollno: 22AD002

Name: ABINAYA P

### Complete Source code :

- App.js

```
// import './App.css'
import Home from "./components/Home"
import Login from "./components/Login"
import Signup from "./components/Signup"
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import { useState } from 'react';
import './App.css';

function App() {
  return (
    <div className="App">
      <Router>
        <Routes>
          <Route path="/" element={<Login/>}/>
          <Route path="/signup" element={<Signup/>}/>
          <Route path="/home" element={<Home/>}/>
        </Routes>
      </Router>
    </div>
  );
}

export default App;
```

- App.css

```
/* General Reset */
body, html {
  margin: 0;
  padding: 0;
  font-family: 'Arial', sans-serif;
  background: linear-gradient(135deg, #f3f4f7, #d9e8f5);
  color: #333;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

/* General Form Styling */
form {
  display: flex;
  flex-direction: column;
```

Rollno: 22AD002

Name: ABINAYA P

```
gap: 15px;
width: 100%;
max-width: 300px;
}

input[type="email"],
input[type="password"],
input[type="submit"] {
padding: 10px;
border: 2px solid #ddd;
border-radius: 5px;
font-size: 16px;
}

input[type="email"]:focus,
input[type="password"]:focus {
border-color: #6aafe3;
outline: none;
box-shadow: 0 0 5px #6aafe3;
}

input[type="submit"] {
background: #6aafe3;
color: white;
border: none;
cursor: pointer;
transition: background 0.3s ease;
}

input[type="submit"]:hover {
background: #558bc2;
}

/* Links */
a {
color: #6aafe3;
text-decoration: none;
font-size: 14px;
}

a:hover {
text-decoration: underline;
}

/* Container for Login/Signup */
.login, .signup {
background: white;
```

Rollno: 22AD002

Name: ABINAYA P

```
padding: 30px;
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
text-align: center;
width: 100%;
max-width: 400px;
}

.login h1,
.signup h1 {
font-size: 24px;
margin-bottom: 20px;
color: #333;
}

/* "OR" Section */
p {
font-size: 16px;
color: #666;
margin: 10px 0;
}

/* Home Page */
.homepage {
text-align: center;
background: white;
padding: 40px;
border-radius: 15px;
box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
}

.homepage h1 {
font-size: 26px;
color: #333;
}

/* Responsive Styling */
@media (max-width: 768px) {
.login, .signup, .homepage {
padding: 20px;
}

form {
max-width: 100%;
}

input[type="email"],
```

Rollno: 22AD002

Name: ABINAYA P

```
input[type="password"],  
input[type="submit"] {  
    font-size: 14px;  
}  
}
```

- Home.js

```
import React from "react"  
import {useLocation, useNavigate} from 'react-router-dom';  
  
function Home (){  
    const location=useLocation()  
  
    return (  
        <div className="homepage">  
  
            <h1>Hello {location.state.id} and welcome to the home</h1>  
  
        </div>  
    )  
}  
  
export default Home
```

- Login.js

```
import React, { useEffect, useState } from "react"  
import axios from "axios"  
import { useNavigate, Link } from "react-router-dom"  
  
function Login() {  
  
    const history=useNavigate();  
  
    const [email,setEmail]=useState("")  
    const [password.setPassword]=useState("")  
  
    async function submit(e){  
        e.preventDefault();  
  
        try{  
  
            await axios.post("http://localhost:8000/",{  
                email,password  
            })  
        }  
    }  
}
```

Rollno: 22AD002

Name: ABINAYA P

```
.then(res=>{
  if(res.data=="exist"){
    history("/home",{state:{id:email}})
  }
  else if(res.data=="notexist"){
    alert("User have not sign up")
  }
})
.catch(e=>{
  alert("wrong details")
  console.log(e);
})

}
catch(e){
  console.log(e);

}

return (
  <div className="login">

    <h1>Login</h1>

    <form action="POST">
      <input type="email" onChange={(e) => { setEmail(e.target.value) }}>
      placeholder="Email" />
      <input type="password" onChange={(e) => { setPassword(e.target.value) }}>
      placeholder="Password" />
      <input type="submit" onClick={submit} />

    </form>

    <br />
    <p>OR</p>
    <br />

    <Link to="/signup">Signup Page</Link>

  </div>
)
}

export default Login
```

Rollno: 22AD002

Name: ABINAYA P

- Signup.js

```
import React, { useEffect, useState } from "react"
import axios from "axios"
import { useNavigate, Link } from "react-router-dom"

function Login() {
  const history=useNavigate();

  const [email,setEmail]=useState("")
  const [password.setPassword]=useState("")

  async function submit(e){
    e.preventDefault();

    try{
      await axios.post("http://localhost:8000/signup",{
        email,password
      })
      .then(res=>{
        if(res.data=="exist"){
          alert("User already exists")
        }
        else if(res.data=="notexist"){
          history("/home",{state:{id:email}})
        }
      })
      .catch(e=>{
        alert("wrong details")
        console.log(e);
      })
    }

    catch(e){
      console.log(e);
    }
  }

  return (
    <div className="login">

      <h1>Signup</h1>
    
```

Rollno: 22AD002

Name: ABINAYA P

```
<form action="POST">
    <input type="email" onChange={(e) => { setEmail(e.target.value) }} placeholder="Email" />
    <input type="password" onChange={(e) => { setPassword(e.target.value) }} placeholder="Password" />
    <input type="submit" onClick={submit} />

</form>

<br />
<p>OR</p>
<br />

<Link to="/">Login Page</Link>

</div>
)
}

export default Login
```

- app.js

```
const express = require("express")
const collection = require("./mongo")
const cors = require("cors")
const app = express()
app.use(express.json())
app.use(express.urlencoded({ extended: true }))
app.use(cors())

app.get("/",cors(),(req,res)=>{

})

app.post("/",async(req,res)=>{
    const{email,password}=req.body

    try{
        const check=await collection.findOne({email:email})

        if(check){
            res.json("exist")
        }
    }
})
```

Rollno: 22AD002

Name: ABINAYA P

```
        }
        else{
            res.json("notexist")
        }

    }
    catch(e){
        res.json("fail")
    }

})

app.post("/signup",async(req,res)=>{
    const{email,password}=req.body

    const data={
        email:email,
        password:password
    }

    try{
        const check=await collection.findOne({email:email})

        if(check){
            res.json("exist")
        }
        else{
            res.json("notexist")
            await collection.insertMany([data])
        }

    }
    catch(e){
        res.json("fail")
    }

})

app.listen(8000,()=>{
    console.log("port connected");
})
```

Rollno: 22AD002

Name: ABINAYA P

- mongo.js

```
const mongoose=require("mongoose")
mongoose.connect("mongodb://0.0.0.0:27017/react-login-tut")
.then(()=>{
  console.log("mongodb connected");
})
.catch(()=>{
  console.log('failed');
})

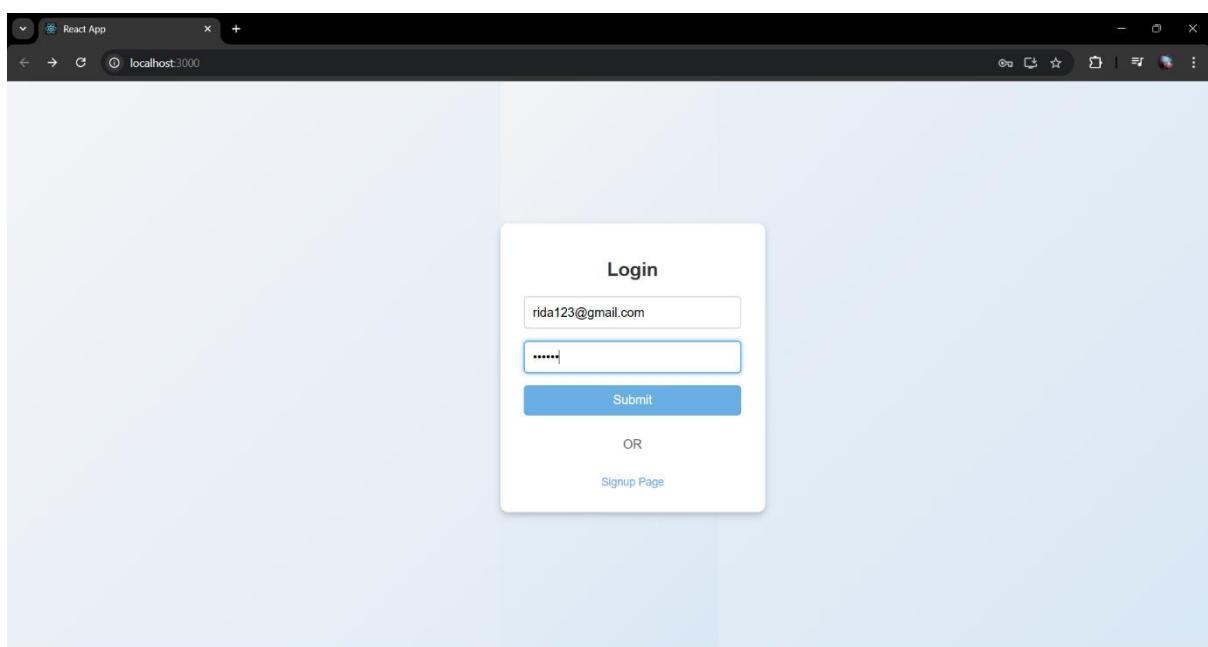
const newSchema=new mongoose.Schema({
  email:{
    type:String,
    required:true
  },
  password:{
    type:String,
    required:true
  }
})

const collection = mongoose.model("collection",newSchema)

module.exports=collection
```

#### Output:

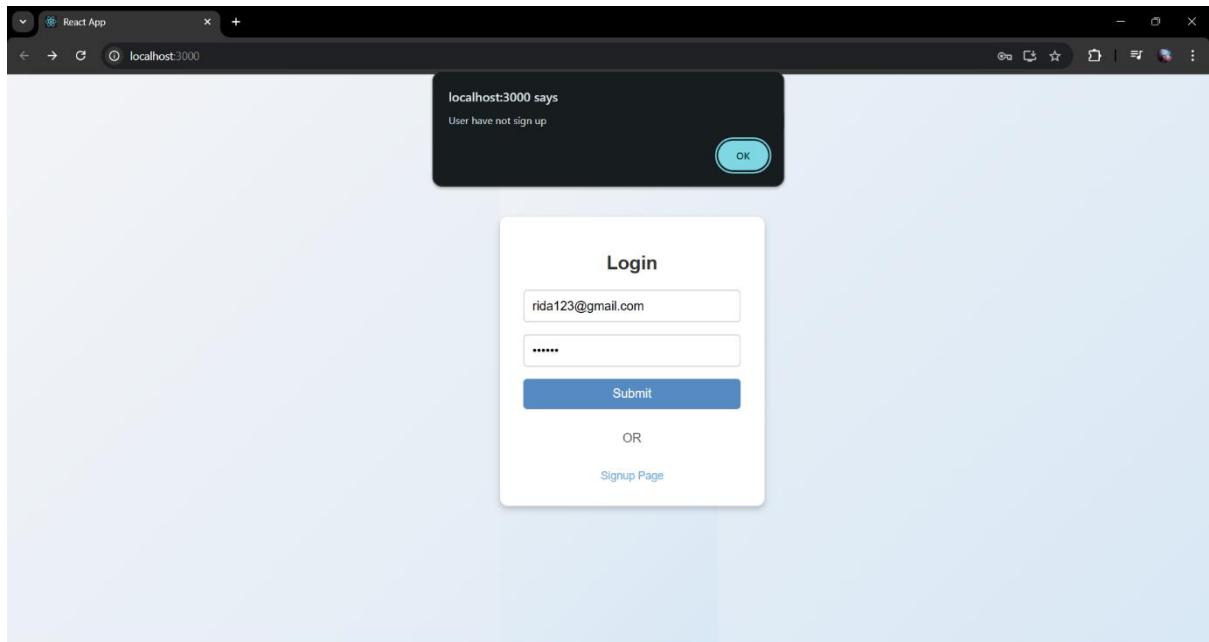
#### Screen 1 : Login Page



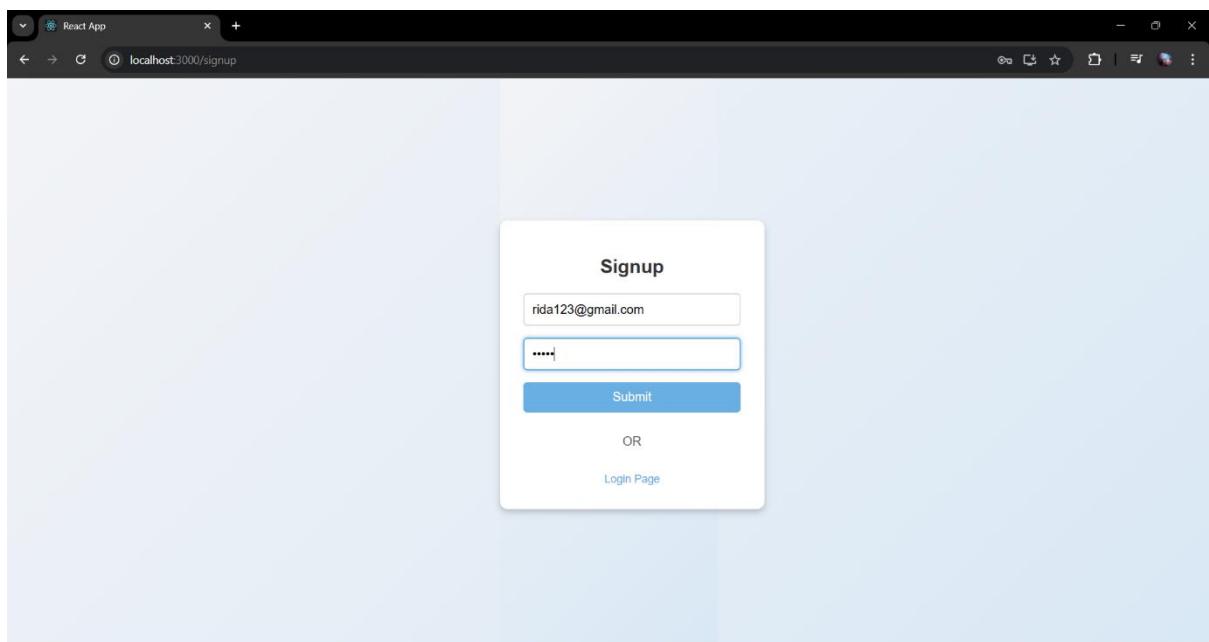
Rollno: 22AD002

Name: ABINAYA P

### Screen 2 : User Not Registered



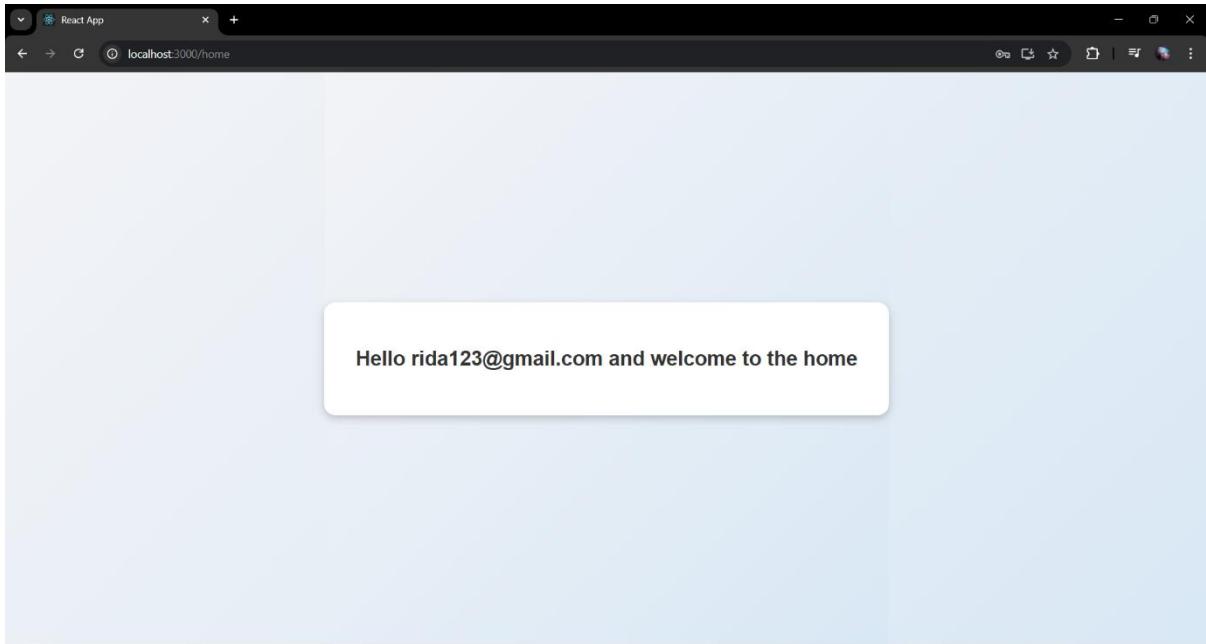
### Screen 3 : Signup Page



### Screen 4 : Home Page

Rollno: 22AD002

Name: ABINAYA P



## Screen 5 : User Data

A screenshot of the MongoDB Compass interface. The left sidebar shows connections and collections, with "react-login-tut" selected. The main pane displays the "collections" tab for the "react-login-tut" database. It shows three documents in the "Documents" list. The first document has the following data:

```
_id: ObjectId('673c11bf9423447b04d87938')
email: "lava@gmail.com"
password: "zxcvbnm"
__v: 0
```

The second document has the following data:

```
_id: ObjectId('673d535cf3bce2ef1e9159d0')
email: "lava2@gmail.com"
password: "123"
__v: 0
```

The third document has the following data:

```
_id: ObjectId('673d55ecf3bce2ef1e9159d4')
email: "rida123@gmail.com"
password: "56789"
__v: 0
```