

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)



B.Tech(CSE-DS) II Year / II Semester (R22)

LAB MANUAL

| | |
|---------------------|---|
| Name of the Faculty | KISHORE K |
| Department | CSE(Data Science) |
| Year & Semester | B.Tech-II & II Sem |
| Regulation | R22 |
| Lab Name | NODE JS (SKILL DEVELOPMENT COURSE) |

DEPARTMENT OF CSE (Data Science)

VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

An Autonomous Institution

AZIZ NAGAR, C B POST, HYDERABAD-500075

2024-2025

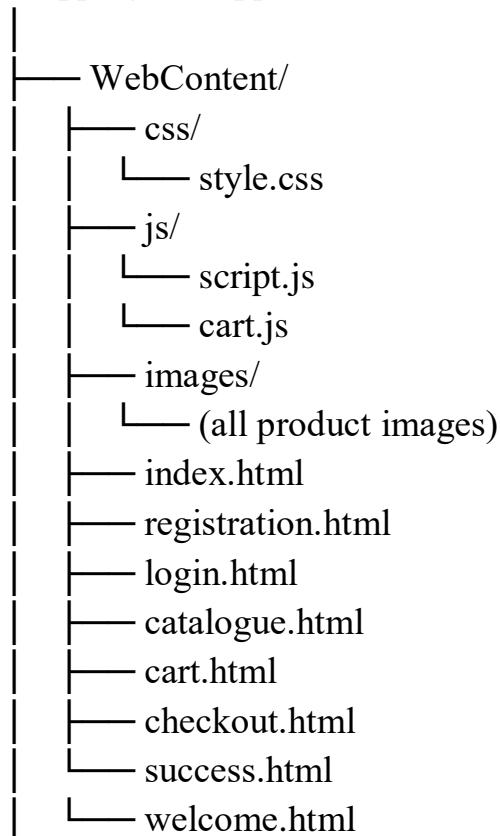
INDEX

| S. No. | Title of the Experiment | Page |
|--------------------|--|----------------|
| NODE JS LAB | | |
| 1. | Build a Responsive Web Application for Shopping Cart with Registration, Login, Catalog and Cart pages using CSS3 features, Flex and Grid. | 1-31 |
| 2. | Make the above Responsive Web Application using Bootstrap Framework. | 32-46 |
| 3. | Use JavaScript for doing Client – Side Validation of the pages for the above 2 Web Applications. | 47-70 |
| 4. | Develop a Java Stand Alone Application that connects with the Database (Oracle / mySql) and perform the CRUD operation on the database tables. | 71-79 |
| 5. | Create an XML for the Bookstore. Validate the same using both DTD and XSD. | 80-86 |
| 6. | Design a Controller with Servlet that provides the interaction with Web Application | 87-91 |
| 7. | Maintaining the transactional history of any user is very important. Explore the various Session Tracking mechanism using Cookies | 92-95 |
| 8. | Maintaining the transactional history of any user is very important. Explore the various Session Tracking mechanism using Sessions. | 96-101 |
| 9. | Develop an Express Web Application that can interact with REST API to perform CRUD operations on Student Data | 102-108 |
| 10. | Create a REACT Application for the Student Management System having Registration, Login, Contact, About pages and implement ROUTING to navigate through these pages. | 109-119 |
| 11. | Create a Custom Server using HTTP Module and explore the other modules of Node JS like OS, PATH, EVENT. | 120-122 |

1. Build a **Responsive Web Application** for **Shopping Cart** with **Registration, Login, Catalog** and **Cart** pages using **CSS3 features, Flex and Grid**.

Project Structure

ShoppingCartApp/



index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
    <title>Registration Form</title>
<style>

    .container1 {
        width: 50%;
        margin: 10px auto;
        background-color: #fff;
        padding: 20px;
```

```
border-radius: 8px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

h2 {
    text-align: center;
    margin-top: 20px;
    margin-left: 60px;
    margin-bottom: 20px;
    width: 80%;
    padding: 10px;
    background-color: #b91df5;
    color: white;
    font-size: 22px;
    border: none;
    border-radius: 4px;
}

img {
    width: 100%;
    border: none;
    border-radius: 4px;
}

.topnav {
    overflow: hidden;
    background-color: #3498db;
}

.topnav a {
    float: left;
    color: #f2f2f2;
    text-align: center;
    padding: 16px 18px;
    text-decoration: none;
    font-size: 19px;
}

.topnav a:hover {
    background-color: #ebdef0;
    color: black;
}

.topnav a.active {
    background-color: #f7f45d;
    color: black;
}
```

```
body {
    background-color: #f8f9fa;
}

</style>
</head>
<body>

    <div class="container1">
        
        <div class="topnav">
            <a class="active" href="index.html">Home</a>
            <a href="registration.html">Registration</a>
            <a href="login.html">Login</a>
            <a href="catalogue.html">Catalogue</a>
            <a href="cart.html">Cart</a>
            <a href="checkout.html">Buy</a>
        </div>
        <h2>Online Catalogue</h2>
        <div id="items-list" class="container">

            <div class="row__content">
                <div id="grid__column">
                    <!--.card-->
                    <div class="card">
                        
                        <div class="info-card">
                            <h4>HTML5</h4>

                            <p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div>
                    </div> <!--.card-->
                    <div class="card">
                        
                        <div class="info-card">
                            <h4>Cooking</h4>

                            <p class="price">Rs.2000 <span class="u-pull-right">Rs.1500</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div>
                    </div> <!--.card-->
                    <div class="card">
```

```

<div class="info-card">
    <h4>Guitar</h4>

    <p class="price">Rs.1200 <span class="u-pull-right">
        "Rs.750</span></p>
        <button class="button button1">Add to Cart</button>
    </div> <!--.card-->
<div class="card">
    
    <div class="info-card">
        <h4>Organic</h4>

        <p class="price">Rs.800 <span class="u-pull-right">
            "Rs.750</span></p>
            <button class="button button1">Add to Cart</button>
        </div> <!--.card-->
        <div class="card">
            
            <div class="info-card">
                <h4>Crafts</h4>

                <p class="price">Rs.2000 <span class="u-pull-right">
                    "Rs.1750</span></p>
                    <button class="button button1">Add to Cart</button>
                </div> <!--.card-->
                <div class="card">
                    
                    <div class="info-card">
                        <h4>FastFoods</h4>

                        <p class="price">Rs.2200 <span class="u-pull-right">
                            "Rs.1750</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div> <!--.card-->
                        </div>
                    </div>
                    </div>
                    </body>
                </html>
```

registration.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Registration Form</title>
<style>

.container {
    width: 50%;
    margin: 10px auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

h2 {
    text-align: center;
    margin-top: 20px;
    margin-left: 60px;
    margin-bottom: 20px;
    width: 80%;
    padding: 10px;
    background-color: #b91df5;
    color: white;
    font-size: 22px;
    border: none;
    border-radius: 4px;
}

img {
    width: 100%;
    border: none;
    border-radius: 4px;
}

.topnav {
    overflow: hidden;
    background-color: #3498db;
}
```

```
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}  
.form-container {  
    max-width: 500px;  
    margin: 50px auto;  
    background-color: white;  
    padding: 30px;  
    border-radius: 15px;  
    box-shadow: 0 0 20px rgba(0,0,0,0.1);  
}  
.form-title {  
    text-align: center;  
    margin-bottom: 25px;  
    font-weight: bold;  
}  
a {  
    text-decoration: none;  
    color: #0FA0CE;  
}  
a:hover {  
    color: #f82a0a; }  
</style>  
</head>  
<body>
```

```
<div class="container">
    
    <div class="topnav">
        <a href="index.html">Home</a>
        <a class="active" href="registration.html">Registration</a>
        <a href="login.html">Login</a>
        <a href="catalogue.html">Catalogue</a>
        <a href="cart.html">Cart</a>
        <a href="checkout.html">Buy</a>
    </div>
    <h2>Registration Form</h2>
    <form name="reg_form" action="success.html" method="post">

        <!-- Username -->
        <div class="mb-3">
            <label for="username" class="form-label">Username</label>
            <input type="text" class="form-control border border-primary border-2 rounded-2" id="username" placeholder="Enter username" required>
        </div>

        <!-- Password -->
        <div class="mb-3">
            <label for="password" class="form-label">Password</label>
            <input type="password" class="form-control border border-primary border-2 rounded-2" id="password" placeholder="Enter password" required>
        </div>

        <!-- Address -->
        <div class="mb-3">
            <label for="address" class="form-label">Address</label>
            <textarea class="form-control border border-primary border-2 rounded-2" id="address" rows="3" placeholder="Enter your address"></textarea>
        </div>

        <!-- Gender -->
        <div class="mb-3">
            <label class="form-label d-block">Gender</label>
            <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio" name="gender" id="male-inline" value="male" checked>
                <label class="form-check-label" for="male-inline">Male</label>
            </div>
            <div class="form-check form-check-inline">
                <input class="form-check-input" type="radio" name="gender" id="female-inline" value="female">
            </div>
        </div>
```

```

<label class="form-check-label" for="female-inline">Female</label>
</div>
<div class="form-check form-check-inline">
  <input class="form-check-input" type="radio" name="gender" id="other-
  inline" value="other">
    <label class="form-check-label" for="other-inline">Other</label>
  </div>

</div>

<!-- Email -->
<div class="mb-3">
  <label for="email" class="form-label">Email ID</label>
  <input type="email" class="form-control border border-primary border-2
rounded-2" id="email" placeholder="Enter your email" required>
</div>

<!-- Mobile Number -->
<div class="mb-3">
  <label for="mobile" class="form-label">Mobile Number</label>
  <input type="tel" class="form-control border border-primary border-2
rounded-2" id="mobile" placeholder="Enter your mobile number" required>
</div>

<!-- Submit Button -->
<div class="d-flex justify-content-center">
<button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
<button type="submit" class="btn btn-success w-50">Register</button><br>
</div>
<h6>Already Registered ! -><a href="login.html">Login here !</a></h6>

</form>
</div>

</body>
</html>

```

success.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/bootstrap@5.3.3/dist/css/bootstrap.min.css"
  rel="stylesheet">
  <title>Registration Success</title>
  <style>
    .container {
      width: 50%;
      margin: 10px auto;
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }

    h2 {
      text-align: center;
      margin-top: 20px;
      margin-left: 60px;
      margin-bottom: 20px;
      width: 80%;
      padding: 10px;
      background-color: #b91df5;
      color: white;
      font-size: 22px;
      border: none;
      border-radius: 4px;
    }

    img {
      width: 100%;
      border: none;
      border-radius: 4px;
    }

    .topnav {
      overflow: hidden;
      background-color: #3498db;
    }

    .topnav a {
      float: left;
      color: #f2f2f2;
      text-align: center;
      padding: 16px 18px;
      text-decoration: none;
      font-size: 19px;
    }
  </style>
```

```
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}  
a {  
    text-decoration: none;  
    font-size: 30px;  
    color: #0FA0CE;  
}  
a:hover {  
    color: #f82a0a; }  
</style>  
</head>  
<body>  
  
<div class="container">  
      
    <div class="topnav">  
        <a href="index.html">Home</a>  
        <a class="active" href="registration.html">Registration</a>  
        <a href="login.html">Login</a>  
        <a href="catalogue.html">Catalogue</a>  
    </div>  
  
    <p><h3>Registration Successful! <a href="login.html">Login here  
!</a></h3></p>  
  
    </div>  
    </body>  
    </html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Login Form</title>
<style>

    .container {
        width: 50%;
        margin: 10px auto;
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }

    h2 {
        text-align: center;
        margin-top: 20px;
        margin-left: 60px;
        margin-bottom: 20px;
        width: 80%;
        padding: 10px;
        background-color: #b91df5;
        color: white;
        font-size: 22px;
        border: none;
        border-radius: 4px;
    }

    img {
        width: 100%;
        border: none;
        border-radius: 4px;
    }

    .topnav {
        overflow: hidden;
        background-color: #3498db;
    }


```

```
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}  
  
a {  
    text-decoration: none;  
    color: #0FA0CE;  
}  
a:hover {  
    color: #f82a0a; }  
</style>  
</head>  
<body>  
  
<div class="container">  
      
        <div class="topnav">  
    <a href="index.html">Home</a>  
    <a href="registration.html">Registration</a>  
    <a class="active" href="login.html">Login</a>  
    <a href="catalogue.html">Catalogue</a>  
    <a href="cart.html">Cart</a>  
    <a href="checkout.html">Buy</a>  
</div>
```

```
<h2>Login Form</h2>
<form action="welcome.html" method="post">

<!-- Username -->

    <label for="username">Username</label>
    <input type="text" id="username" placeholder="Enter username"
required><br>

<!-- Password -->

    <label for="password">Password</label>
    <input type="password" id="password" placeholder="Enter password"
required><br>

<!-- Submit Button -->

<button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
<button type="submit" class="btn btn-success w-50">Login</button></br>

<h6>Not Registered -> <a href="registration.html">Register here !</a></h6>

</form>

</body>
</html>
```

welcome.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/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">
        <title>Login Success</title>
        <style>

            .container {
                width: 50%;
```

```
        margin: 10px auto;
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }

h2 {
    text-align: center;
    margin-top: 20px;
    margin-left: 60px;
    margin-bottom: 20px;
    width: 80%;
    padding: 10px;
    background-color: #b91df5;
    color: white;
    font-size: 22px;
    border: none;
    border-radius: 4px;
}

img {
    width: 100%;
    border: none;
    border-radius: 4px;
}

.topnav {
    overflow: hidden;
    background-color: #3498db;
}

.topnav a {
    float: left;
    color: #f2f2f2;
    text-align: center;
    padding: 16px 18px;
    text-decoration: none;
    font-size: 19px;
}

.topnav a:hover {
    background-color: #ebdef0;
    color: black;
}
```

```
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}  
a {  
    text-decoration: none;  
    font-size: 30px;  
    color: #0FA0CE;  
}  
a:hover {  
    color: #f82a0a; }  
</style>  
</head>  
<body>  
  
<div class="container">  
      
    <div class="topnav">  
        <a href="index.html">Home</a>  
        <a href="registration.html">Registration</a>  
        <a class="active" href="login.html">Login</a>  
        <a href="catalogue.html">Catalogue</a>  
        <a href="cart.html">Cart</a>  
        <a href="checkout.html">Buy</a>  
    </div>  
  
    <p><h3>Login Successful! <a href="catalogue.html">Lets Shopping now  
!</a></h3></p>  
  
    </div>  
    </body>  
    </html>
```

catalogue.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
        <script src="script.js"></script>
        <title>Online Catalogue</title>
<style>

.container1 {
    width: 50%;
    margin: 10px auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

h2 {
    text-align: center;
        margin-top: 20px;
        margin-left: 60px;
    margin-bottom: 20px;
        width: 80%;
    padding: 10px;
    background-color: #b91df5;
    color: white;
    font-size: 22px;
    border: none;
    border-radius: 4px;
}
img {
    width: 100%;
    border: none;
    border-radius: 4px;
}

.topnav {
    overflow: hidden;
    background-color: #3498db;
}
```

```
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}  
  
</style>  
</head>  
<body>  
  
<div class="container1">  
      
    <div class="topnav">  
        <a href="index.html">Home</a>  
        <a href="registration.html">Registration</a>  
        <a href="login.html">Login</a>  
        <a class="active" href="catalogue.html">Catalogue</a>  
        <a href="cart.html">Cart</a>  
        <a href="checkout.html">Buy</a>  
    </div>  
    <h2>Online Catalogue</h2>  
    <div id="items-list" class="container">  
        <div class="row_content">  
            <div id="grid_column">
```

```
<div class="card">

<div class="info-card">
    <h4>HTML5</h4>

    <p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
<div class="card">
    
    <div class="info-card">
        <h4>Cooking</h4>

        <p class="price">Rs.2000 <span class="u-pull-right">Rs.1500</span></p>
            <button class="button button1">Add to Cart</button>
        </div>
    </div> <!--.card-->
    <div class="card">
        
        <div class="info-card">
            <h4>Guitar</h4>

            <p class="price">Rs.1200 <span class="u-pull-right">Rs.750</span></p>
                <button class="button button1">Add to Cart</button>
            </div>
        </div> <!--.card-->
        <div class="card">
            
            <div class="info-card">
                <h4>Organic</h4>

                <p class="price">Rs.800 <span class="u-pull-right">Rs.750</span></p>
                    <button class="button button1">Add to Cart</button>
                </div>
            </div> <!--.card-->
                <div class="card">
                    
                    <div class="info-card">
                        <h4>Crafts</h4>
```

```
<p class="price">Rs.2000 <span class="u-pull-right">  
">Rs.1750</span></p>  
    <button class="button button1">Add to Cart</button>  
    </div>  
</div> <!--.card-->  
    <div class="card">  
          
        <div class="info-card">  
            <h4>FastFoods</h4>  
  
            <p class="price">Rs.2200 <span class="u-pull-right">  
">Rs.1750</span></p>  
            <button class="button button1">Add to Cart</button>  
            </div>  
</div> <!--.card-->  
  
    </div>  
    </div>  
    </div>  
  
</body>  
</html>
```

cart.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Cart</title>  
  
<script defer src="cart.js"></script>  
    <style>  
  
.container1 {  
    width: 50%;  
    margin: 10px auto;  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 8px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  
}
```

```
h2 {  
    text-align: center;  
    margin-top: 20px;  
    margin-left: 60px;  
    margin-bottom: 20px;  
    width: 80%;  
    padding: 10px;  
    background-color: #b91df5;  
    color: white;  
    font-size: 22px;  
    border: none;  
    border-radius: 4px;  
  
}  
img {  
    width: 100%;  
    border: none;  
    border-radius: 4px;  
  
}  
.topnav {  
    overflow: hidden;  
    background-color: #3498db;  
}  
  
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}
```

```
body {  
    background-color: #f8f9fa;  
}  
  
</style>  
</head>  
<body>  
<div class="container1">  
      
    <div class="topnav">  
        <a href="index.html">Home</a>  
        <a href="registration.html">Registration</a>  
        <a href="login.html">Login</a>  
        <a href="catalogue.html">Catalogue</a>  
        <a class="active" href="cart.html">Cart</a>  
        <a href="checkout.html">Buy</a>  
    </div>  
  
    <h2>Your Cart</h2>  
    <table>  
        <thead>  
            <tr>  
                <th>Item</th>  
                <th>Image</th>  
                <th>Price</th>  
                <th>Quantity</th>  
                <th>Subtotal</th>  
            </tr>  
        </thead>  
        <tbody id="cart-items"></tbody>  
    </table>  
    <h4 id="total-price">Total: Rs.0</h4>  
    <a href="checkout.html">Proceed to Checkout</a>  
</div>  
</body>  
</html>
```

checkout.html

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

<style>

    .container1 {
        width: 50%;
        margin: 10px auto;
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }

    h2 {
        text-align: center;
        margin-top: 20px;
        margin-left: 60px;
        margin-bottom: 20px;
        width: 80%;
        padding: 10px;
        background-color: #b91df5;
        color: white;
        font-size: 22px;
        border: none;
        border-radius: 4px;
    }

    img {
        width: 100%;
        border: none;
        border-radius: 4px;
    }

    .topnav {
        overflow: hidden;
        background-color: #3498db;
    }


```

```
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}
```

```
body {  
    background-color: #f8f9fa;  
}  
  
</style>  
</head>  
<body>  
<div class="container1">  
      
    <div class="topnav">  
        <a href="index.html">Home</a>  
        <a href="registration.html">Registration</a>  
        <a href="login.html">Login</a>  
        <a href="catalogue.html">Catalogue</a>  
        <a href="cart.html">Cart</a>  
        <a class="active" href="checkout.html">Buy</a>  
    </div>  
  
    <h2>Checkout</h2>  
    <form>  
        <label for="name">Full Name:</label>  
        <input type="text" id="name" required><br>
```

```

<label for="address">Shipping Address:</label>
<textarea id="address" rows="3" required></textarea><br>

<label for="payment">Payment Method:</label>
<select id="payment">
    <option value="COD">Cash on Delivery</option>
    <option value="card">Credit/Debit Card</option>
</select><br>

<button type="reset">Cancel Order</button>
<button type="submit">Place Order</button><br>

</form>
<script>
    document.getElementById("checkout-form").addEventListener("submit",
function(e) {
    e.preventDefault();
    localStorage.removeItem("cart");
    document.getElementById("confirmation").innerHTML =
        <div class="alert alert-success">
            Order placed successfully! Thank you for shopping.
        </div>;
});
</script>
</body>
</html>

```

script.js

```

// script.js
document.addEventListener("DOMContentLoaded", () => {
    const buttons = document.querySelectorAll(".button1");

    buttons.forEach((button, index) => {
        button.addEventListener("click", () => {
            const card = button.closest(".card");
            const name = card.querySelector("h4").innerText;
            const price = parseFloat(card.querySelector(".u-pull-
right").innerText.replace("Rs.", ""));
            const image = card.querySelector("img").src;

```

```
const item = { name, price, image, quantity: 1 };

let cart = JSON.parse(localStorage.getItem("cart")) || [];

const existingItem = cart.find(i => i.name === item.name);
if (existingItem) {
    existingItem.quantity += 1;
} else {
    cart.push(item);
}

localStorage.setItem("cart", JSON.stringify(cart));
alert(` ${item.name} added to cart.`);
});  
});  
});
```

style.css

```
//style.css

h1
{
    font-weight: 600;
    font-size: 28px;
}
h4
{
    font-weight: 600;
    font-size: 18px;
}
p {
    margin-top: 0;
    color: #f82a0a;
}
header {
    padding: 20px 0 0 0;
    background: white;
    text-align: center;
}
.flex__navbar{
    display: flex;
    justify-content: space-between;
}
```

```
#heading {  
    margin: 30px 0;  
}  
/* Links */  
  
.logo > a {  
    color: #222;  
    list-style: none;  
    text-decoration: none;  
}  
a {  
    text-decoration: none;  
    font-size: 30px;  
    color: #f82a0a;  
}  
a:hover {  
    color: #0FA0CE; }  
/* Grid */  
  
.container {  
    position: relative;  
    width: 100%;  
    max-width: 960px;  
    margin: 0 auto;  
    padding: 0 20px;  
    box-sizing: border-box; }  
.column,  
.columns {  
    width: 100%;  
    float: left;  
    box-sizing: border-box; }  
  
#grid__column{  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-gap: 2rem;  
}  
/*----- */  
  
#banner{  
    width: 100%;  
    background: url('./img/banner.jpg') no-repeat center center/cover;  
    height: 350px;  
}
```

```
/*-----Catalogue ---- */
#items-list .row {
    margin-bottom:20px;
}

.row__content > #grid__column{
    grid-template-columns: repeat(3, 1fr);
    margin: 0 auto;
}

.card {
    text-align: center;
    border: 1px solid #e1e1e1;
    background: white;
}

.item-image{
    width: 100%;
    height: 10rem;
}

.card h4 {
    margin-bottom: 5px;
}
.info-card {
    padding: 10px 20px;
}
.price {
    text-decoration: line-through;
    font-size: 18px;
    margin-top: 10px;
}
.u-pull-right {
    float: right; }
.price span {
    font-weight: 600;
    font-size: 18px;
}

.button {
    border: none;
    color: white;
    padding: 10px 15px;
```

```
text-align: center;
text-decoration: none;
display: inline-block;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}

.button1 {background-color: #008CBA;} /* Blue */

.footer {
margin-top: 40px;
border-top: 1px solid #aaa;
padding-top: 40px;
}
```

cart.js

```
// cart.js
document.addEventListener("DOMContentLoaded", () => {
  const cart = JSON.parse(localStorage.getItem("cart")) || [];
  const cartItems = document.getElementById("cart-items");
  const totalPrice = document.getElementById("total-price");

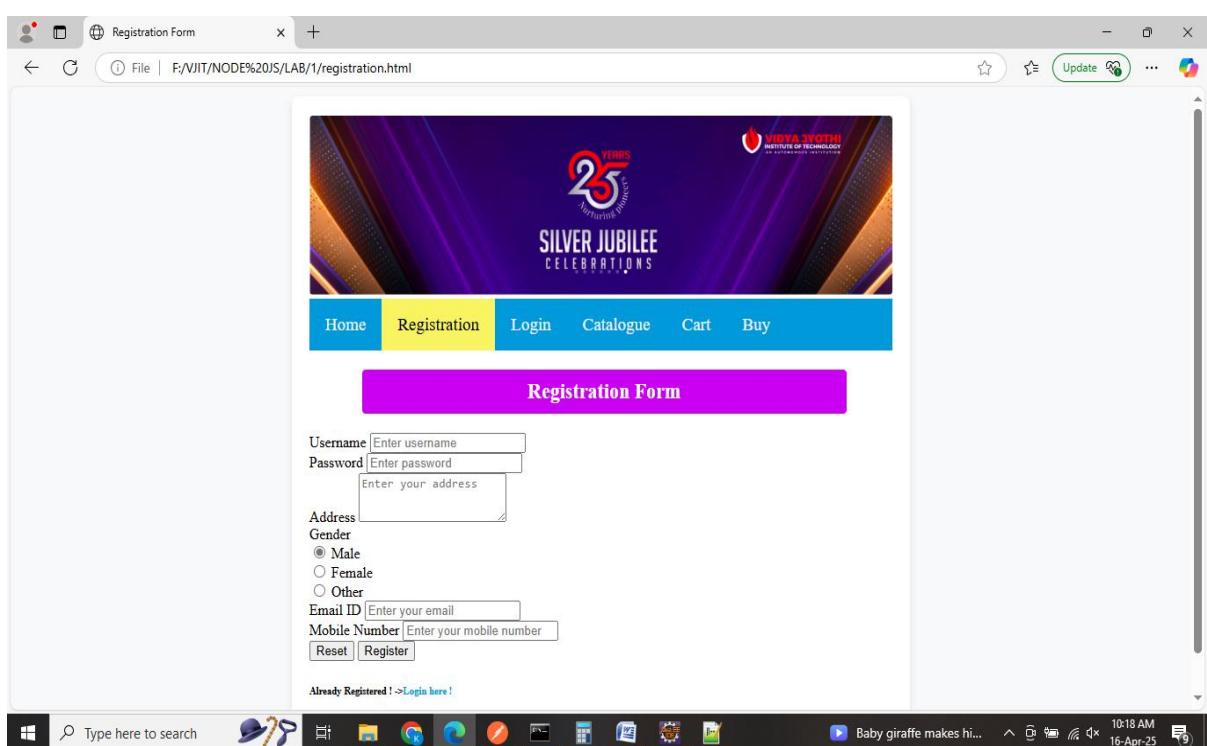
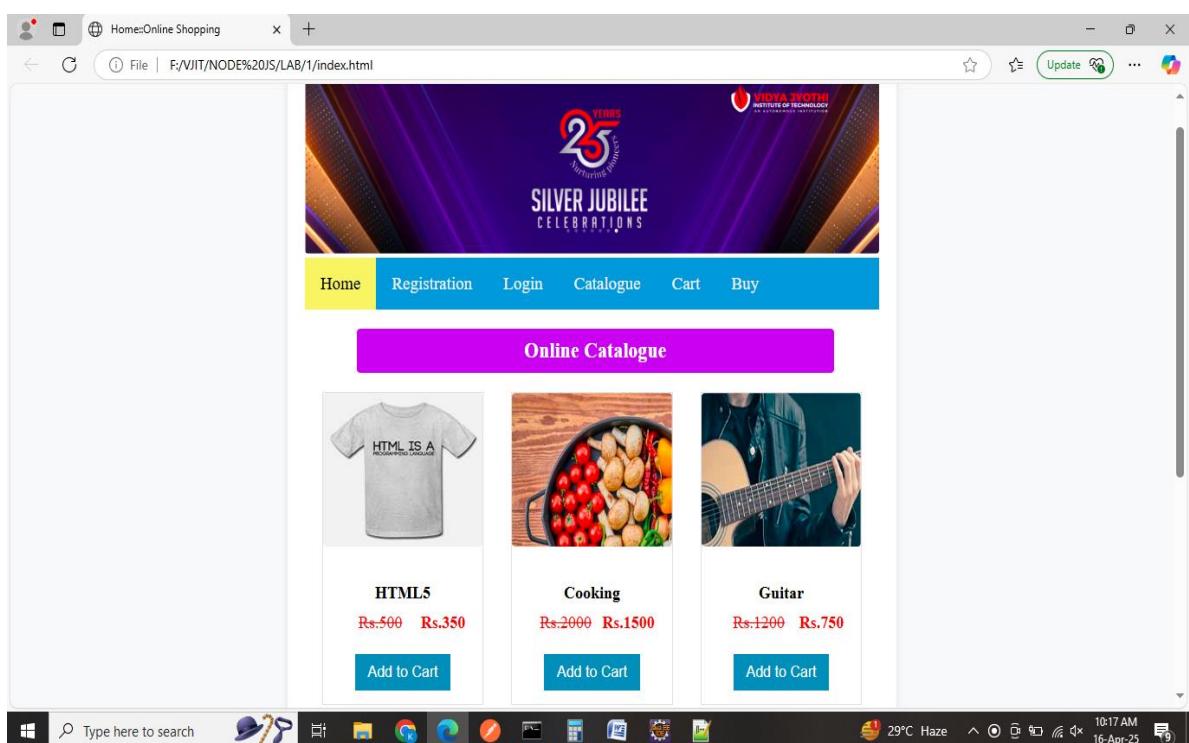
  let total = 0;
  cart.forEach(item => {
    const row = document.createElement("tr");

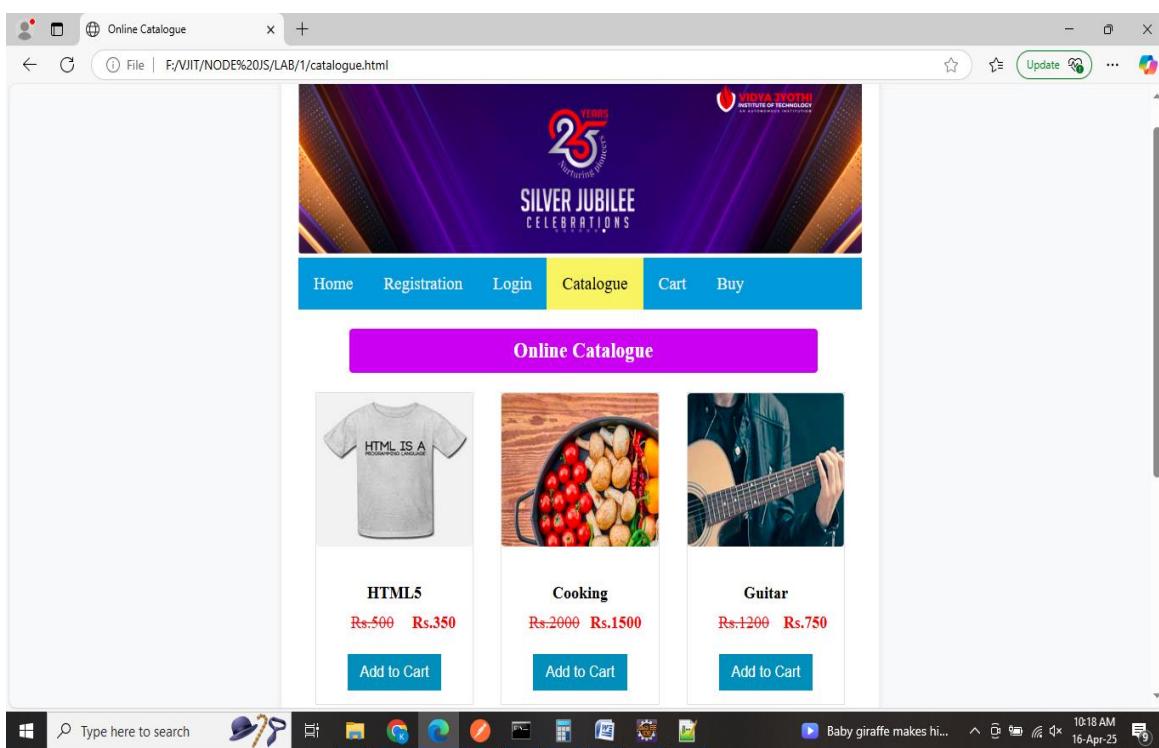
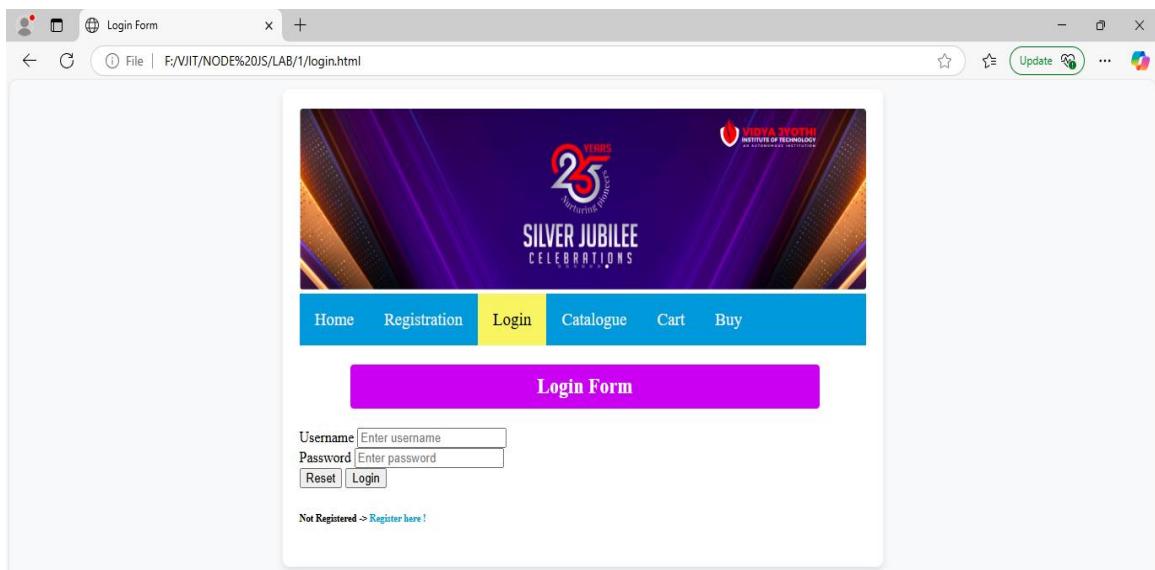
    const subtotal = item.price * item.quantity;
    total += subtotal;

    row.innerHTML = `
      <td>${item.name}</td>
      <td></td>
      <td>Rs.${item.price}</td>
      <td>${item.quantity}</td>
      <td>Rs.${subtotal}</td>
    `;
    cartItems.appendChild(row);
  });

  totalPrice.innerText = `Total: Rs.${total}`;
});
```

Output:





25 Years Silver Jubilee Celebrations

VJIT INSTITUTE OF TECHNOLOGY

Home Registration Login Catalogue Cart Buy

Your Cart

| Item | Image | Price | Quantity | Subtotal |
|---------|-------|---------|----------|----------|
| HTML5 | | Rs.350 | 1 | Rs.350 |
| Cooking | | Rs.1500 | 1 | Rs.1500 |

Total: Rs.1850

[Proceed to Checkout](#)



25 Years Silver Jubilee Celebrations

VJIT INSTITUTE OF TECHNOLOGY

Home Registration Login Catalogue Cart Buy

Checkout

Full Name:

Shipping Address:

Payment Method: Cash on Delivery

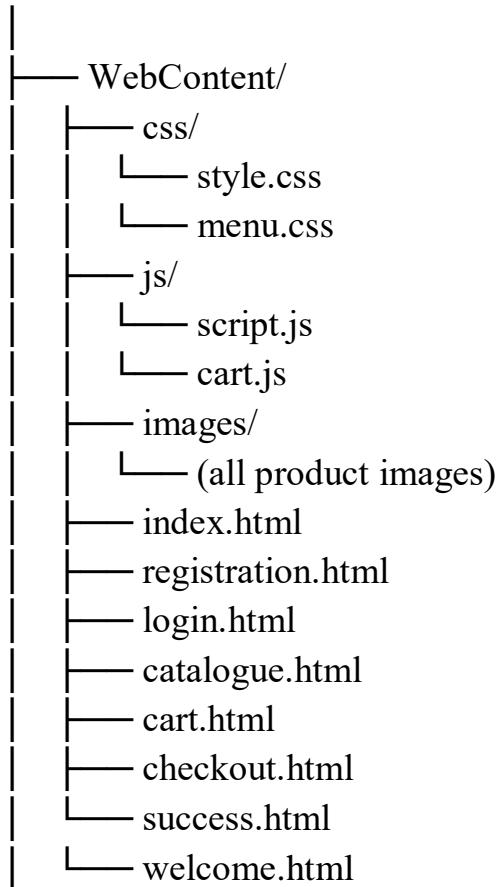
[Cancel Order](#) [Place Order](#)



2. Make the above Responsive Web Application using Bootstrap Framework.

Project Structure

ShoppingCartApp/



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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="./css/style.css">
        <link rel="stylesheet" href="./css/menu.css">
        <script src="script.js"></script>
        <title>Registration Form</title>

</head>
<body>

    <div class="container1">
        
```

```
<div class="topnav">
<a class="active" href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
        <h2>Online Catalogue</h2>
        <div id="items-list" class="container">

<div class="row__content">
    <div id="grid__column">
        <!--.card-->
            <div class="card">
                
                <div class="info-card">
                    <h4>HTML5</h4>

                    <p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
                    <button class="button button1">Add to Cart</button>
                </div>
            </div> <!--.card-->
            <div class="card">
                
                <div class="info-card">
                    <h4>Cooking</h4>

                    <p class="price">Rs.2000 <span class="u-pull-right">Rs.1500</span></p>
                    <button class="button button1">Add to Cart</button>
                </div>
            </div> <!--.card-->
            <div class="card">
                
                <div class="info-card">
                    <h4>Guitar</h4>

                    <p class="price">Rs.1200 <span class="u-pull-right">Rs.750</span></p>
                    <button class="button button1">Add to Cart</button>
                </div>
            </div> <!--.card-->
            <div class="card">
```

```

<div class="info-card">
    <h4>Organic</h4>

        <p class="price">Rs.800 <span class="u-pull-right ">Rs.750</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
    <div class="card">

<div class="info-card">
    <h4>Crafts</h4>

        <p class="price">Rs.2000 <span class="u-pull-right
">Rs.1750</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
    <div class="card">

<div class="info-card">
    <h4>FastFoods</h4>

        <p class="price">Rs.2200 <span class="u-pull-right
">Rs.1750</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->

    </div>
</div>
</div>

</body>
</html>
```

Registration.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="./css/menu.css">
        <title>Registration Form</title>
</head>
<body>
    <div class="container1">
        
            <div class="topnav">
<a href="index.html">Home</a>
<a class="active" href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
            <h2>Registration Form</h2>
            <form name="reg_form" action="success.html" method="post">
                <!-- Username -->
                <div class="mb-3">
                    <label for="username" class="form-label">Username</label>
                    <input type="text" class="form-control border border-primary border-2 rounded-2" id="username" placeholder="Enter username" required>
                </div>
                <!-- Password -->
                <div class="mb-3">
                    <label for="password" class="form-label">Password</label>
                    <input type="password" class="form-control border border-primary border-2 rounded-2" id="password" placeholder="Enter password" required>
                </div>
                <!-- Address -->
                <div class="mb-3">
                    <label for="address" class="form-label">Address</label>
                    <textarea class="form-control border border-primary border-2 rounded-2" id="address" rows="3" placeholder="Enter your address"></textarea>
                </div>
                <!-- Gender -->
                <div class="mb-3">
```

```
<label class="form-label d-block">Gender</label>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="gender" id="male-inline" value="male" checked>
<label class="form-check-label" for="male-inline">Male</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="gender" id="female-inline" value="female">
<label class="form-check-label" for="female-inline">Female</label>
</div>
<div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="gender" id="other-inline" value="other">
<label class="form-check-label" for="other-inline">Other</label>
</div>
</div>

<!-- Email -->
<div class="mb-3">
<label for="email" class="form-label">Email ID</label>
<input type="email" class="form-control border border-primary border-2 rounded-2" id="email" placeholder="Enter your email" required>
</div>

<!-- Mobile Number -->
<div class="mb-3">
<label for="mobile" class="form-label">Mobile Number</label>
<input type="tel" class="form-control border border-primary border-2 rounded-2" id="mobile" placeholder="Enter your mobile number" required>
</div>

<!-- Submit Button -->
<div class="d-flex justify-content-center">
<button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
<button type="submit" class="btn btn-success w-50">Register</button><br>
</div>
<h6>Already Registered ! -><a href="login.html">Login here !</a></h6>

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

Success.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <title>Registration Success</title>
        <link rel="stylesheet" href=".css/menu.css">
</head>
<body>
    <div class="container1">
        
            <div class="topnav">
<a href="index.html">Home</a>
<a class="active" href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
</div>
<p><h3>Registration Successful! <a href="login.html">Login here !</a></h3></p>
</div>
</body>
</html>
```

Login.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href=".css/menu.css">
        <title>Login Form</title>
</head>
<body>
    <div class="container1">
        
            <div class="topnav">
<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a class="active" href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
```

```

<a href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
      <h2>Login Form</h2>
      <form action="welcome.html" method="post">
        <!-- Username -->
        <div class="mb-3">
          <label for="username" class="form-label">Username</label>
          <input type="text" class="form-control border border-primary border-2 rounded-2" id="username" placeholder="Enter username" required>
        </div>
        <!-- Password -->
        <div class="mb-3">
          <label for="password" class="form-label">Password</label>
          <input type="password" class="form-control border border-primary border-2 rounded-2" id="password" placeholder="Enter password" required>
        </div>
        <!-- Submit Button -->
        <div class="d-flex justify-content-center">
          <button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
          <button type="submit" class="btn btn-success w-50">Login</button><br>
        </div>
        <h6>Not Registered -> <a href="registration.html">Register here !</a></h6>
      </form>
    </div>
  </body>
</html>

```

Welcome.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="./css/menu.css">
    <title>Login Success</title>
  </head>
  <body>
    <div class="container1">
      
      <div class="topnav">

```

```

<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a class="active" href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
<p><h3>Login Successful! <a href="catalogue.html">Lets Shopping now
!</a></h3></p>
</div>
</body>
</html>

```

Catalogue.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="./css/style.css">
        <link rel="stylesheet" href="./css/menu.css">
        <script src="script.js"></script>
        <title>Online Catalogue</title>
    </head>
<body>
    <div class="container1">
        
            <div class="topnav">
<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a class="active" href="catalogue.html">Catalogue</a>
<a href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
            <h2>Online Catalogue</h2>
            <div id="items-list" class="container">
                <div class="row__content">
                    <div id="grid__column">
                        <!--.card-->
                            <div class="card">

```

```

<div class="info-card">
    <h4>HTML5</h4>

    <p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
    <button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">
    
    <div class="info-card">
        <h4>Cooking</h4>

        <p class="price">Rs.2000 <span class="u-pull-right
    "Rs.1500</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
<div class="card">
    
    <div class="info-card">
        <h4>Guitar</h4>
        <p class="price">Rs.1200 <span class="u-pull-right ">Rs.750</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
<div class="card">
    
    <div class="info-card">
        <h4>Organic</h4>
        <p class="price">Rs.800 <span class="u-pull-right ">Rs.750</span></p>
        <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
    <div class="card">
        
        <div class="info-card">
            <h4>Crafts</h4>
            <p class="price">Rs.2000 <span class="u-pull-right ">Rs.1750</span></p>
            <button class="button button1">Add to Cart</button>
        </div>
</div> <!--.card-->
    <div class="card">
        
        <div class="info-card">
```

```

<h4>FastFoods</h4>
<p class="price">Rs.2200 <span class="u-pull-right ">Rs.1750</span></p>
    <button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
</div>
</div>
</div>
</div>
</body>
</html>

```

Cart.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cart</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="./css/menu.css">
        <script defer src="cart.js"></script>
</head>
<body>
<div class="container1">
    
        <div class="topnav">
<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a class="active" href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
<div class="container mt-5">
    <h2>Your Cart</h2>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Item</th>
                <th>Image</th>
                <th>Price</th>
                <th>Quantity</th>

```

```

        <th>Subtotal</th>
    </tr>
</thead>
<tbody id="cart-items"></tbody>
</table>
<h4 id="total-price">Total: Rs.0</h4>
<a href="checkout.html" class="btn btn-success">Proceed to Checkout</a>
</div>
</body>
</html>

```

Checkout.html

```

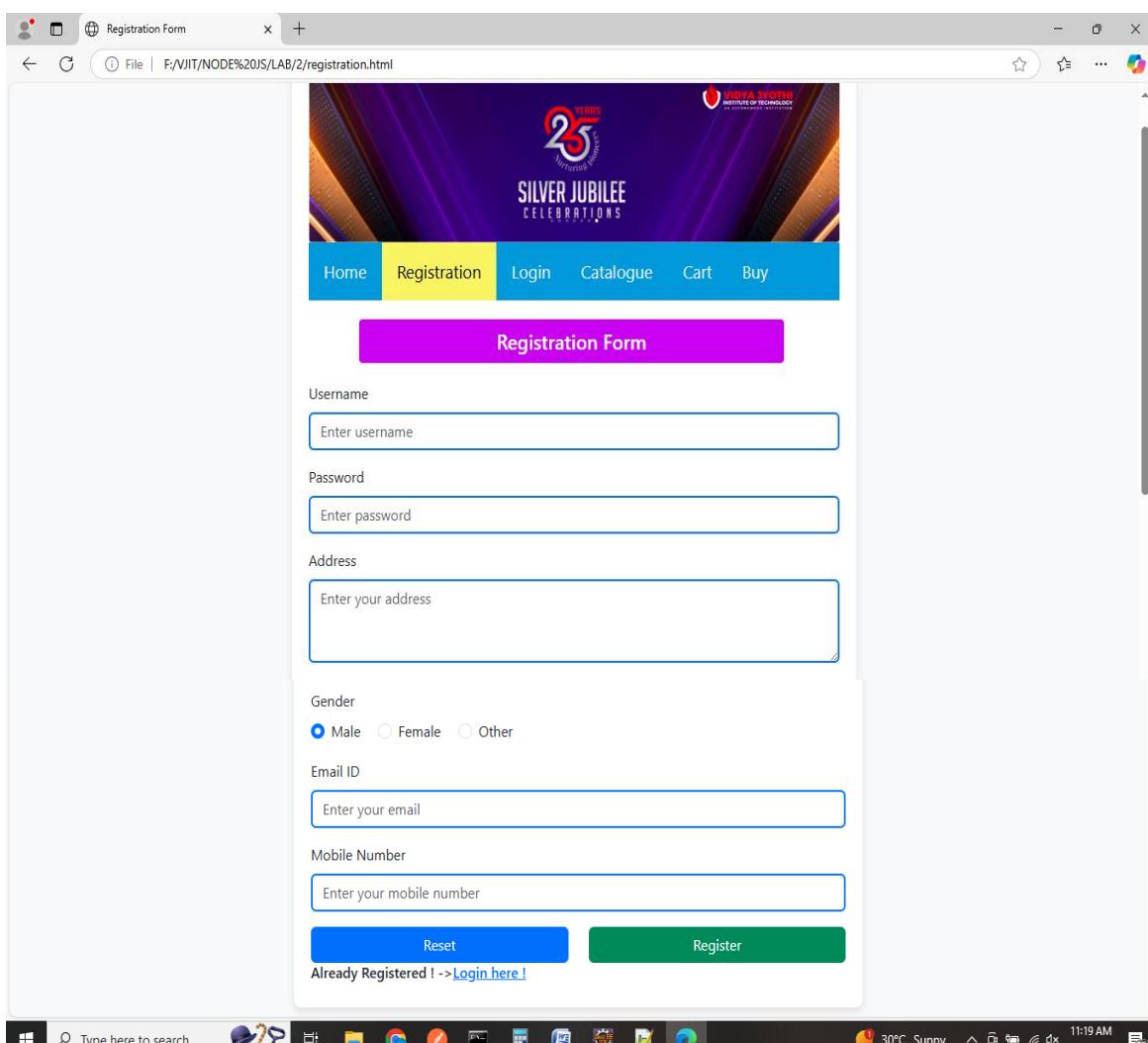
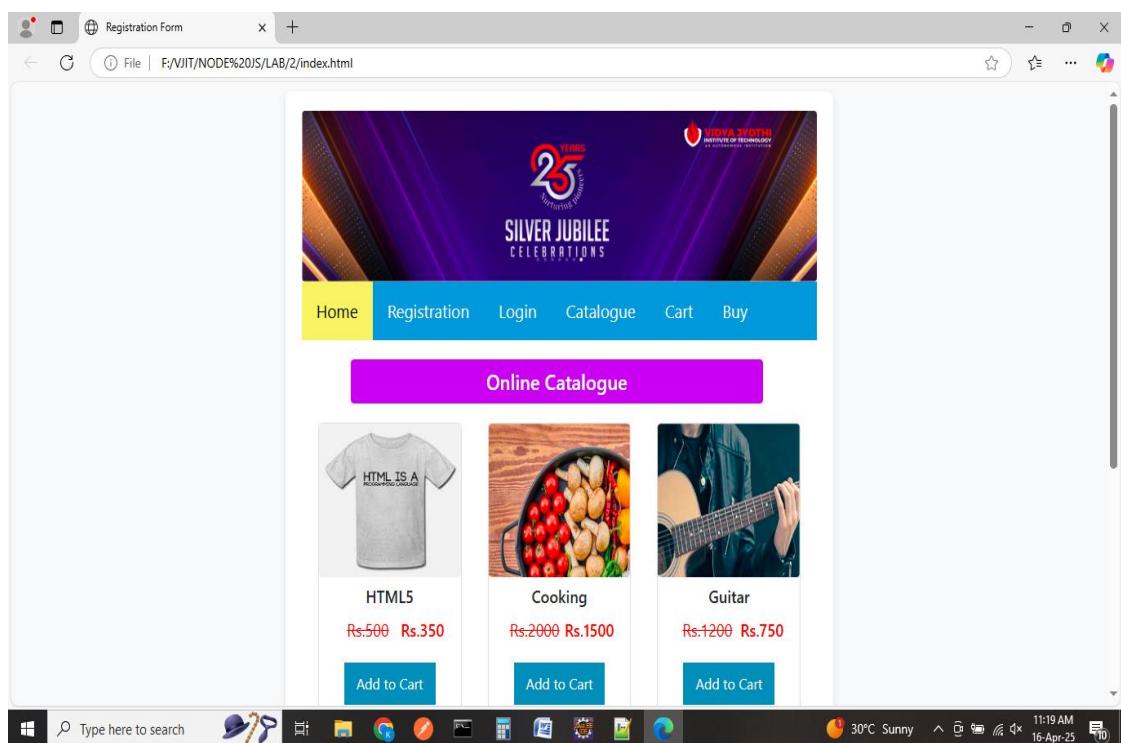
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Checkout</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="./css/menu.css">
</head>
<body>
<div class="container1">
    
        <div class="topnav">
<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a href="cart.html">Cart</a>
<a class="active" href="checkout.html">Buy</a>
</div>
<div class="container mt-5">
    <h2>Checkout</h2>
    <form id="checkout-form">
        <div class="mb-3">
            <label for="name" class="form-label">Full Name:</label>
            <input type="text" id="name" class="form-control border border-primary border-2 rounded-2" required>
        </div>
        <div class="mb-3">
            <label for="address" class="form-label">Shipping Address:</label>
            <textarea id="address" class="form-control border border-primary border-2 rounded-2" rows="3" required></textarea>
        </div>
    </form>
</div>

```

```
</div>
<div class="mb-3">
    <label for="payment" class="form-label">Payment Method:</label>
    <select id="payment" class="form-control border border-primary border-2 rounded-2">
        <option value="COD">Cash on Delivery</option>
        <option value="card">Credit/Debit Card</option>
    </select>
</div>
<div class="d-flex justify-content-center">
    <button type="reset" class="btn btn-primary w-50 me-4">Cancel Order</button>
    <button type="submit" class="btn btn-success w-50 me-4">Place Order</button><br>
</div>
</form>
<div id="confirmation" class="mt-4"></div>
</div>

<script>
    document.getElementById("checkout-form").addEventListener("submit",
function(e) {
    e.preventDefault();
    localStorage.removeItem("cart");
    document.getElementById("confirmation").innerHTML = `<div class="alert alert-success">
        Order placed successfully! Thank you for shopping.
    </div>`;
});
</script>
</body>
</html>
```

Output:



Login Form

F:\VJIT\NODE%20JS\LAB\2\login.html

25 Years
SILVER JUBILEE
CELEBRATIONS

Home Registration Login Catalogue Cart Buy

Login Form

Username
Enter username

Password
Enter password

Reset Login

Not Registered -> [Register here!](#)



Online Catalogue

F:\VJIT\NODE%20JS\LAB\2\catalogue.html

25 Years
SILVER JUBILEE
CELEBRATIONS

Home Registration Login Catalogue Cart Buy

Online Catalogue

| | | |
|--|--|---|
| | | |
| HTML5 Rs:500 Rs:350 Add to Cart | Cooking Rs:2000 Rs:1500 Add to Cart | Guitar Rs:1200 Rs:750 Add to Cart |
| | | |
| Organic Rs:800 Rs:750 Add to Cart | Crafts Rs:2000 Rs:1750 Add to Cart | Fast Foods Rs:2200 Rs:1750 Add to Cart |



Your Cart

| Item | Image | Price | Quantity | Subtotal |
|-----------|-------|---------|----------|----------|
| HTML5 | | Rs.350 | 2 | Rs.700 |
| Cooking | | Rs.1500 | 1 | Rs.1500 |
| Crafts | | Rs.1750 | 1 | Rs.1750 |
| Guitar | | Rs.750 | 1 | Rs.750 |
| FastFoods | | Rs.1750 | 1 | Rs.1750 |

Total: Rs.6450

[Proceed to Checkout](#)

Checkout

Full Name:

Shipping Address:

Payment Method:

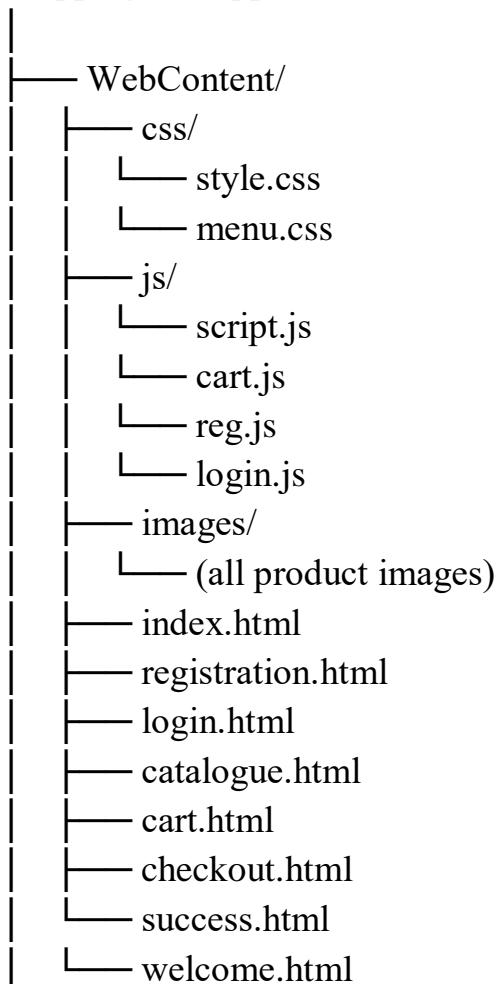
Cash on Delivery Please fill out this field.

[Cancel Order](#) [Place Order](#)

- 3.** Use JavaScript for doing Client – Side Validation of the pages for the above 2 Web Applications.

Project Structure

ShoppingCartApp/



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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href=".//css/style.css">
  <link rel="stylesheet" href=".//css/menu.css">
  <script src=".//js/script.js"></script>
  <title>Registration Form</title>
</head>
  
```

```
<body>

    <div class="container1">
        
        <div class="topnav">
            <a class="active" href="index.html">Home</a>
            <a href="registration.html">Registration</a>
            <a href="login.html">Login</a>
            <a href="catalogue.html">Catalogue</a>
            <a href="cart.html">Cart</a>
            <a href="checkout.html">Buy</a>
        </div>
        <h2>Online Catalogue</h2>
        <div id="items-list" class="container">

            <div class="row__content">
                <div id="grid__column">
                    <!--.card-->
                    <div class="card">
                        
                        <div class="info-card">
                            <h4>HTML5</h4>

                            <p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div>
                    </div> <!--.card-->
                    <div class="card">
                        
                        <div class="info-card">
                            <h4>Cooking</h4>

                            <p class="price">Rs.2000 <span class="u-pull-right">Rs.1500</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div>
                    </div> <!--.card-->
                    <div class="card">
                        
                        <div class="info-card">
                            <h4>Guitar</h4>

                            <p class="price">Rs.1200 <span class="u-pull-right ">Rs.750</span></p>
                            <button class="button button1">Add to Cart</button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```

</div> <!--.card-->
<div class="card">
  
  <div class="info-card">
    <h4>Organic</h4>

    <p class="price">Rs.800 <span class="u-pull-right ">Rs.750</span></p>
    <button class="button button1">Add to Cart</button>
  </div>
</div> <!--.card-->
  <div class="card">
    
    <div class="info-card">
      <h4>Crafts</h4>

      <p class="price">Rs.2000 <span class="u-pull-right ">Rs.1750</span></p>
      <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->
  <div class="card">
    
    <div class="info-card">
      <h4>FastFoods</h4>
      <p class="price">Rs.2200 <span class="u-pull-right ">Rs.1750</span></p>
      <button class="button button1">Add to Cart</button>
    </div>
</div> <!--.card-->

  </div>
</div>
</div>

</body>
</html>

```

menu.css

```

.container1 {
  width: 50%;
  margin: 10px auto;
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

```

```
h2 {  
    text-align: center;  
    margin-top: 20px;  
    margin-left: 60px;  
    margin-bottom: 20px;  
    width: 80%;  
    padding: 10px;  
    background-color: #b91df5;  
    color: white;  
    font-size: 22px;  
    border: none;  
    border-radius: 4px;  
  
}  
img {  
    width: 100%;  
    border: none;  
    border-radius: 4px;  
  
}  
.topnav {  
    overflow: hidden;  
    background-color: #3498db;  
}  
.topnav a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 16px 18px;  
    text-decoration: none;  
    font-size: 19px;  
}  
.topnav a:hover {  
    background-color: #ebdef0;  
    color: black;  
}  
.topnav a.active {  
    background-color: #f7f45d;  
    color: black;  
}  
  
body {  
    background-color: #f8f9fa;  
}
```

```
.form-container {  
    max-width: 500px;  
    margin: 50px auto;  
    background-color: white;  
    padding: 30px;  
    border-radius: 15px;  
    box-shadow: 0 0 20px rgba(0,0,0,0.1);  
}  
.form-title {  
    text-align: center;  
    margin-bottom: 25px;  
    font-weight: bold;  
}
```

registration.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  
        <link rel="stylesheet" href="./css/menu.css">  
    <title>Registration Form</title>  
  
    </head>  
<body>  
  
    <div class="container1">  
          
            <div class="topnav">  
                <a href="index.html">Home</a>  
                <a class="active" href="registration.html">Registration</a>  
                <a href="login.html">Login</a>  
                <a href="catalogue.html">Catalogue</a>  
                <a href="cart.html">Cart</a>  
                <a href="checkout.html">Buy</a>  
            </div>  
        <h2>Registration Form</h2>  
        <form name="reg_form" onsubmit="return validate()" method="post" action="success.html">
```

```
<!-- Username -->
<div class="mb-3">
    <label for="username" class="form-label">Username</label>
    <input type="text" name="uname" class="form-control border border-primary border-2 rounded-2" id="username" placeholder="Enter username" required>
</div>

<!-- Password -->
<div class="mb-3">
    <label for="password" class="form-label">Password</label>
    <input type="password" name="pass" class="form-control border border-primary border-2 rounded-2" id="password" placeholder="Enter password" required>
</div>

<!-- Address -->
<div class="mb-3">
    <label for="address" class="form-label">Address</label>
    <textarea name="address" class="form-control border border-primary border-2 rounded-2" id="address" rows="3" placeholder="Enter your address"></textarea>
</div>

<!-- Gender -->
<div class="mb-3">
<label class="form-label d-block">Gender</label>
<div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="gender" id="male-inline" value="male">
        <label class="form-check-label" for="male-inline">Male</label>
</div>
<div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="gender" id="female-inline" value="female">
        <label class="form-check-label" for="female-inline">Female</label>
</div>
<div class="form-check form-check-inline">
    <input class="form-check-input" type="radio" name="gender" id="other-inline" value="other">
        <label class="form-check-label" for="other-inline">Other</label>
</div>
</div>

<!-- Email -->
<div class="mb-3">
    <label for="email" class="form-label">Email ID</label>
    <input type="email" name="email" class="form-control border border-primary border-2 rounded-2" id="email" placeholder="Enter your email" required>
</div>
```

```

<!-- Mobile Number -->
<div class="mb-3">
    <label for="mobile" class="form-label">Mobile Number</label>
    <input type="tel" maxlength="10" name="mobile" class="form-control border border-primary border-2 rounded-2" id="mobile" placeholder="Enter your mobile number" required>
</div>
<!-- Submit Button -->
<div class="d-flex justify-content-center">
<button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
<button type="submit" class="btn btn-success w-50">Register</button><br>
</div>
<h6>Already Registered ! -><a href="login.html">Login here !</a></h6>

</form>
</div>
<script src=".js/reg.js"></script>
</body>

</html>

```

reg.js

```

//reg.js
function validate() {
var uname = document.reg_form.uname;
var pass = document.reg_form.pass;
var address = document.reg_form.address;
var email = document.reg_form.email;
var mobile = document.reg_form.mobile;
var genderRadios = document.reg_form.gender;

if (uname.value.trim() === "") {
    alert("User Name is required");
    uname.focus();
    return false;
}

if (!pass.value.match(/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[^a-zA-Z0-9])(?!.*\s).{8,15}$/)) {
    alert("Password must be 8-15 characters long and contain a lowercase, uppercase, digit, and special character.");
    pass.focus();
    return false;
}

```

```
if (address.value.trim() === "") {
    alert("Address is required");
    address.focus();
    return false;
}

let genderSelected = false;
for (let i = 0; i < genderRadios.length; i++) {
    if (genderRadios[i].checked) {
        genderSelected = true;
        break;
    }
}
if (!genderSelected) {
    alert("Please select your gender.");
    return false;
}

if (!email.value.match(/^\w+([.-]?\w+)*@\w+([.-]?\w+)*(\.\w{2,3})+$/)) {
    alert("Please enter a valid email!");
    email.focus();
    return false;
}

if (!mobile.value.match(/^1-9][0-9]{9}$/)) {
    alert("Phone number must be 10 digits and cannot start with 0.");
    mobile.focus();
    return false;
}

return true;
}
```

success.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="./css/menu.css">
    <title>Registration Success</title>
</head>
```

```
<body>

    <div class="container">
        
        <div class="topnav">
            <a href="index.html">Home</a>
            <a class="active" href="registration.html">Registration</a>
            <a href="login.html">Login</a>
            <a href="catalogue.html">Catalogue</a>
        </div>

        <p><h3>Registration Successful! <a href="login.html">Login here !</a></h3></p>
    </div>

</body>
</html>
```

login.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="./css/menu.css">
    <title>Login Form</title>
</head>
<body>

    <div class="container1">
        
        <div class="topnav">
            <a href="index.html">Home</a>
            <a href="registration.html">Registration</a>
            <a class="active" href="login.html">Login</a>
            <a href="catalogue.html">Catalogue</a>
            <a href="cart.html">Cart</a>
            <a href="checkout.html">Buy</a>
        </div>
        <h2>Login Form</h2>
    </div>
```

```

<form name="reg_form" onsubmit="return validate()" action="welcome.html"
method="post">
<!-- Email -->
<div class="mb-3">
    <label for="email" class="form-label">Email ID</label>
    <input type="email" name="email" class="form-control border border-primary
border-2 rounded-2" id="email" placeholder="Enter your email" required>
</div>

<!-- Password -->
<div class="mb-3">
    <label for="password" class="form-label">Password</label>
    <input type="password" name="pass" class="form-control border border-primary
border-2 rounded-2" id="password" placeholder="Enter password" required>
</div>

<!-- Submit Button -->
<div class="d-flex justify-content-center">
<button type="reset" class="btn btn-primary w-50 me-4">Reset</button>
<button type="submit" class="btn btn-success w-50">Login</button><br>
</div>
<h6>Not Registered -> <a href="registration.html">Register here !</a></h6>

</form>
</div>
<script src=".js/login.js"></script>
</body>
</html>

```

login.js

//login.js

```

function validate() {

var pass = document.reg_form.pass;
var email = document.reg_form.email;
if (!pass.value.match(/^(?=.*\d)(?=.**[a-z])(?=.*[A-Z])(?=.*[^a-zA-Z0-
9])(?!.*\s).{8,15}$/)) {
    alert("Password must be 8-15 characters long and contain a lowercase, uppercase,
digit, and special character.");
    pass.focus();
    return false;
}

```

```
if (!email.value.match(/^\w+([.-]?\w+)*@\w+([.-]?\w+)*(.\w{2,3})+$)) {  
    alert("Please enter a valid email!");  
    email.focus();  
    return false;  
}  
return true;  
}
```

catalogue.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/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">  
        <link rel="stylesheet" href="./css/style.css">  
        <link rel="stylesheet" href="./css/menu.css">  
        <script src="script.js"></script>  
    <title>Online Catalogue</title>  
  
</head>  
<body>  
  
    <div class="container1">  
          
        <div class="topnav">  
            <a href="index.html">Home</a>  
            <a href="registration.html">Registration</a>  
            <a href="login.html">Login</a>  
            <a class="active" href="catalogue.html">Catalogue</a>  
            <a href="cart.html">Cart</a>  
            <a href="checkout.html">Buy</a>  
        </div>  
        <h2>Online Catalogue</h2>  
        <div id="items-list" class="container">  
  
            <div class="row__content">  
                <div id="grid__column">  
                    <!--.card-->  
                    <div class="card">  
                          
                        <div class="info-card">  
                            <h4>HTML5</h4>
```

```
<p class="price">Rs.500 <span class="u-pull-right">Rs.350</span></p>
<button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">

<div class="info-card">
<h4>Cooking</h4>

<p class="price">Rs.2000 <span class="u-pull-right ">Rs.1500</span></p>
<button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">

<div class="info-card">
<h4>Guitar</h4>

<p class="price">Rs.1200 <span class="u-pull-right ">Rs.750</span></p>
<button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">

<div class="info-card">
<h4>Organic</h4>

<p class="price">Rs.800 <span class="u-pull-right ">Rs.750</span></p>
<button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">

<div class="info-card">
<h4>Crafts</h4>

<p class="price">Rs.2000 <span class="u-pull-right ">Rs.1750</span></p>
<button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->
<div class="card">

<div class="info-card">
<h4>FastFoods</h4>
```

```
<p class="price">Rs.2200 <span class="u-pull-right ">Rs.1750</span></p>
    <button class="button button1">Add to Cart</button>
</div>
</div> <!--.card-->

</div>
</div>
</div>

</body>
</html>
```

script.js

```
// script.js
document.addEventListener("DOMContentLoaded", () => {
    const buttons = document.querySelectorAll(".button1");

    buttons.forEach((button, index) => {
        button.addEventListener("click", () => {
            const card = button.closest(".card");
            const name = card.querySelector("h4").innerText;
            const price = parseFloat(card.querySelector(".u-pull-right").innerText.replace("Rs.", ""));
            const image = card.querySelector("img").src;

            const item = { name, price, image, quantity: 1 };

            let cart = JSON.parse(localStorage.getItem("cart")) || [];

            const existingItem = cart.find(i => i.name === item.name);
            if (existingItem) {
                existingItem.quantity += 1;
            } else {
                cart.push(item);
            }

            localStorage.setItem("cart", JSON.stringify(cart));
            alert(`${item.name} added to cart.`);
        });
    });
});
```

cart.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Cart</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href=".//css/menu.css">
    <script defer src="cart.js"></script>

</head>
<body>
<div class="container1">
    
        <div class="topnav">
<a href="index.html">Home</a>
<a href="registration.html">Registration</a>
<a href="login.html">Login</a>
<a href="catalogue.html">Catalogue</a>
<a class="active" href="cart.html">Cart</a>
<a href="checkout.html">Buy</a>
</div>
<div class="container mt-5">
    <h2>Your Cart</h2>
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Item</th>
                <th>Image</th>
                <th>Price</th>
                <th>Quantity</th>
                <th>Subtotal</th>
            </tr>
        </thead>
        <tbody id="cart-items"></tbody>
    </table>
    <h4 id="total-price">Total: Rs.0</h4>
    <a href="checkout.html" class="btn btn-success">Proceed to Checkout</a>
</div>
</body>
</html>
```

cart.js

```
// cart.js
document.addEventListener("DOMContentLoaded", () => {
  const cart = JSON.parse(localStorage.getItem("cart")) || [];
  const cartItems = document.getElementById("cart-items");
  const totalPrice = document.getElementById("total-price");

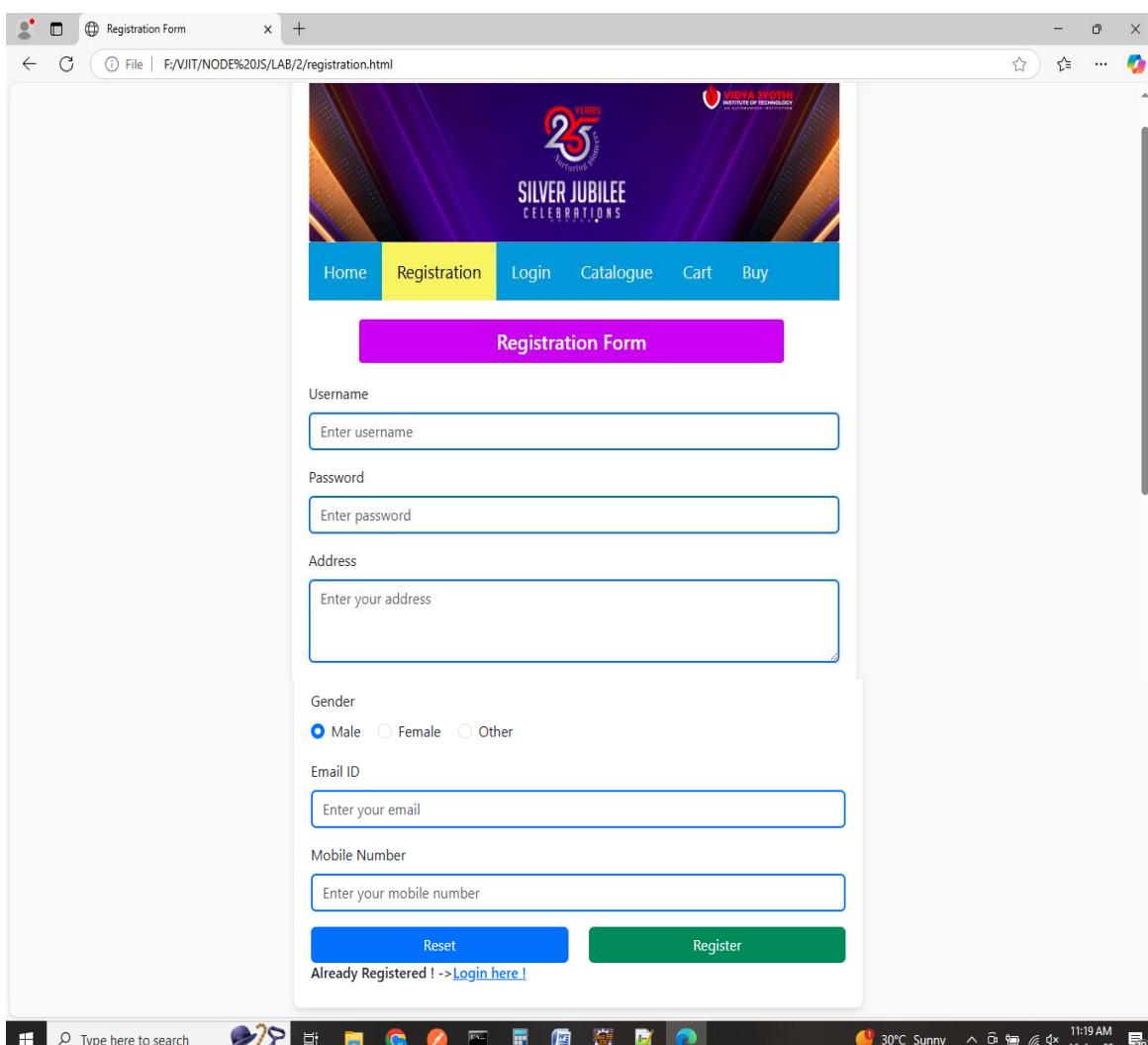
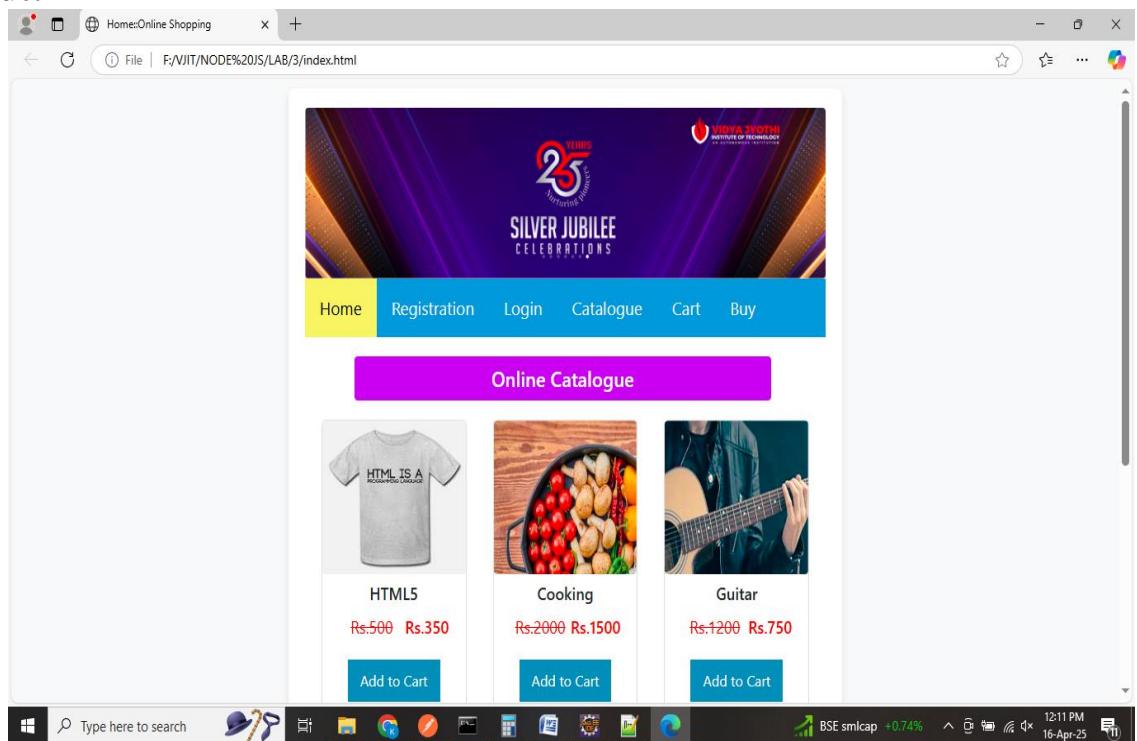
  let total = 0;
  cart.forEach(item => {
    const row = document.createElement("tr");
    const subtotal = item.price * item.quantity;
    total += subtotal;
    row.innerHTML = `
      <td>${item.name}</td>
      <td></td>
      <td>Rs.${item.price}</td>
      <td>${item.quantity}</td>
      <td>Rs.${subtotal}</td>
    `;
    cartItems.appendChild(row);
  });
  totalPrice.innerText = `Total: Rs.${total}`;
});
```

checkout.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Checkout</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="./css/menu.css">
</head>
<body>
  <div class="container1">
    
    <div class="topnav">
      <a href="index.html">Home</a>
      <a href="registration.html">Registration</a>
      <a href="login.html">Login</a>
      <a href="catalogue.html">Catalogue</a>
      <a href="cart.html">Cart</a>
      <a class="active" href="checkout.html">Buy</a>
    </div>
  </div>
```

```
<div class="container mt-5">
  <h2>Checkout</h2>
  <form id="checkout-form">
    <div class="mb-3">
      <label for="name" class="form-label">Full Name:</label>
      <input type="text" id="name" class="form-control border border-primary border-2 rounded-2" required>
    </div>
    <div class="mb-3">
      <label for="address" class="form-label">Shipping Address:</label>
      <textarea id="address" class="form-control border border-primary border-2 rounded-2" rows="3" required></textarea>
    </div>
    <div class="mb-3">
      <label for="payment" class="form-label">Payment Method:</label>
      <select id="payment" class="form-control border border-primary border-2 rounded-2">
        <option value="COD">Cash on Delivery</option>
        <option value="card">Credit/Debit Card</option>
      </select>
    </div>
    <div class="d-flex justify-content-center">
      <button type="reset" class="btn btn-primary w-50 me-4">Cancel Order</button>
      <button type="submit" class="btn btn-success w-50 me-4">Place Order</button>
    </div>
  </form>
  <div id="confirmation" class="mt-4"></div>
</div>

<script>
  document.getElementById("checkout-form").addEventListener("submit", function(e) {
    e.preventDefault();
    localStorage.removeItem("cart");
    document.getElementById("confirmation").innerHTML = `<div class="alert alert-success">
      Order placed successfully! Thank you for shopping.
    </div>`;
  });
</script>
</body>
</html>
```

Output:

Registration Form

SILVER JUBILEE CELEBRATIONS

Home Registration Login Catalogue Cart Buy

Registration Form

Username
Rahul

Password

Address
Hyderabad

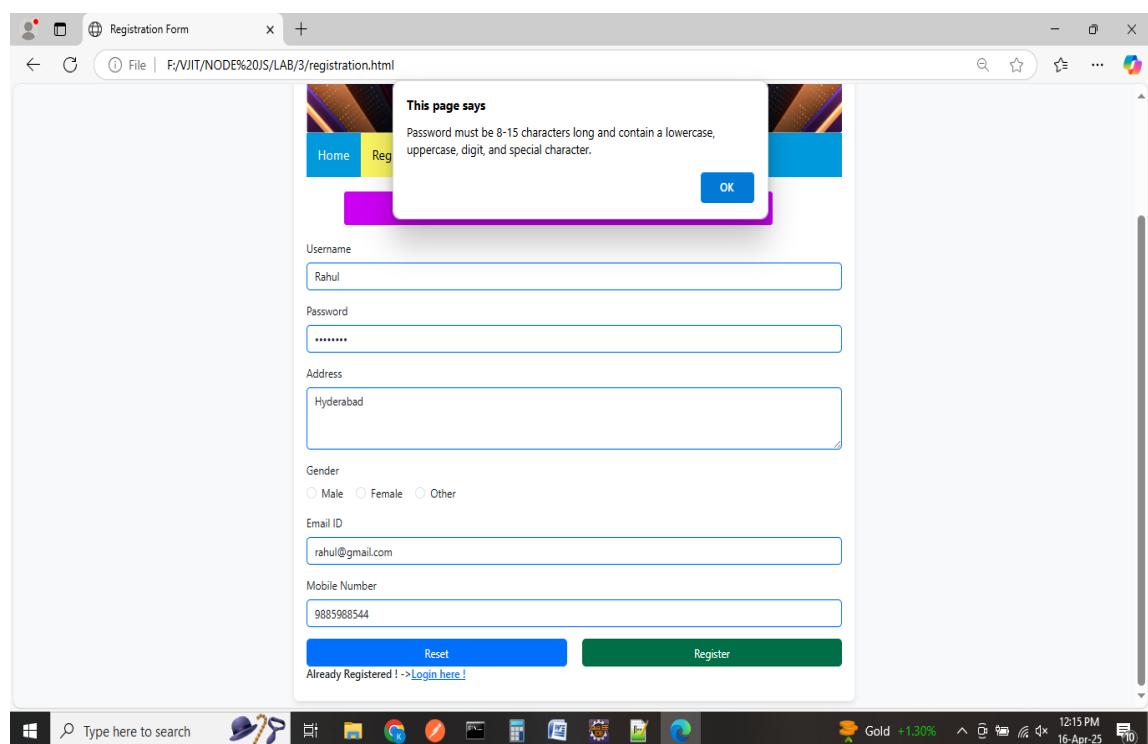
Gender
 Male Female Other

Email ID
rahul@gmail.com

Mobile Number
9885988544

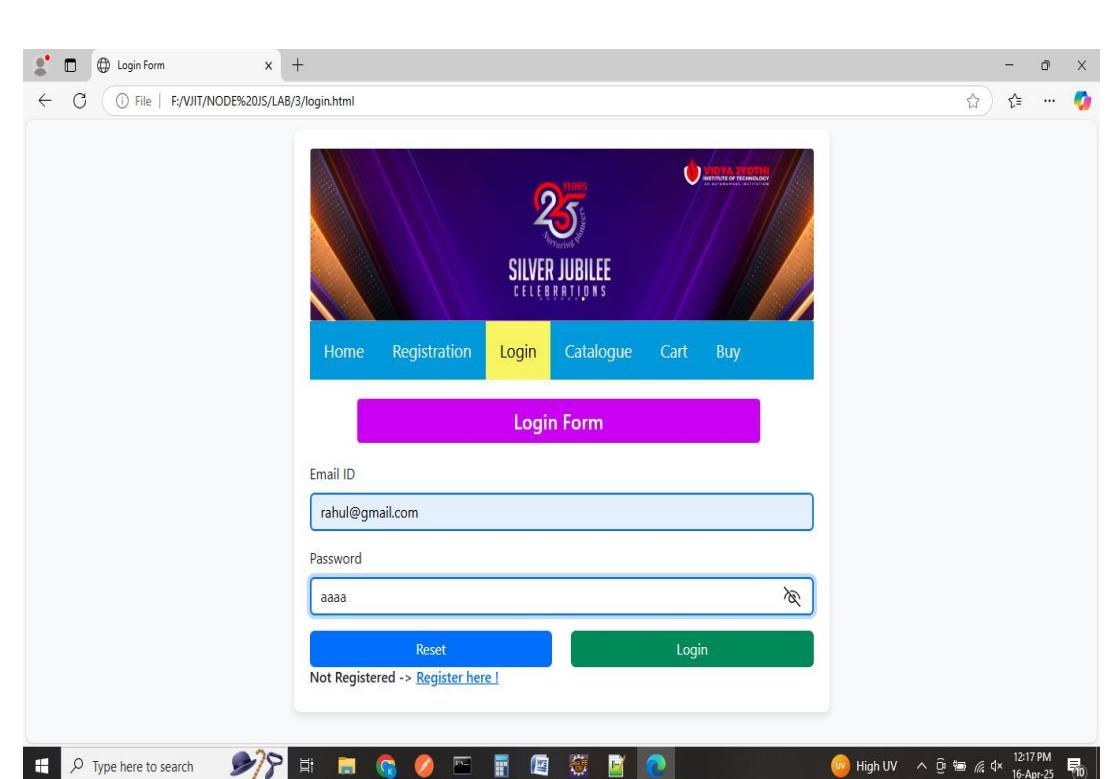
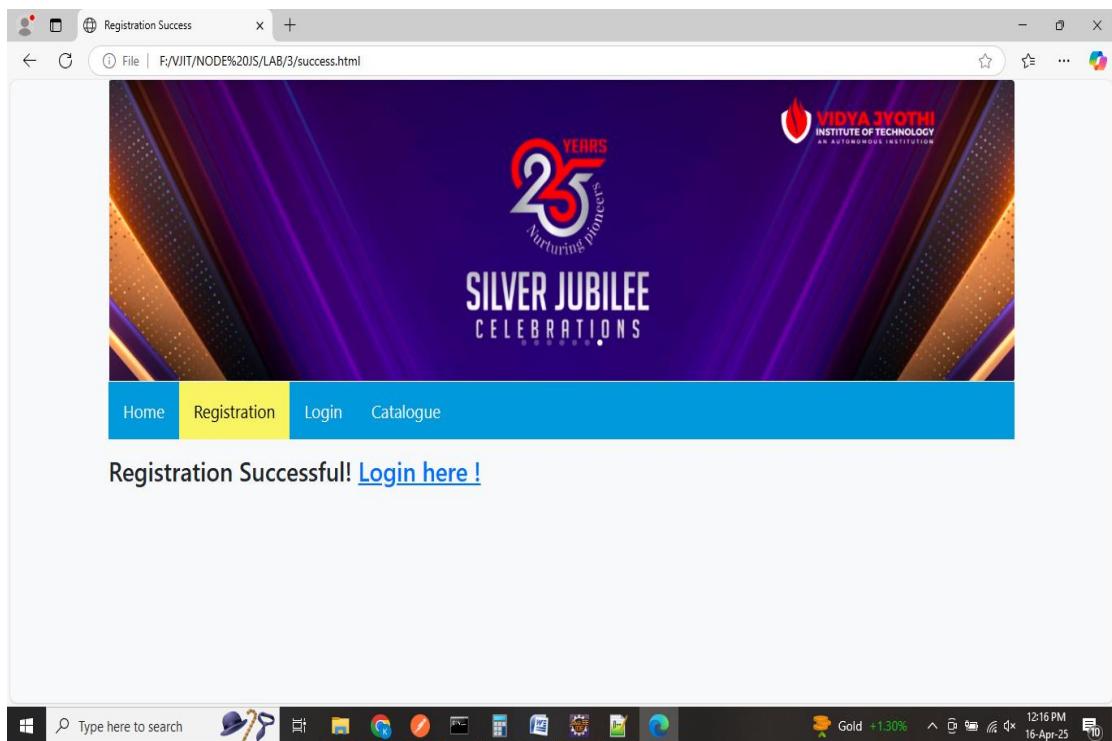
Reset Register

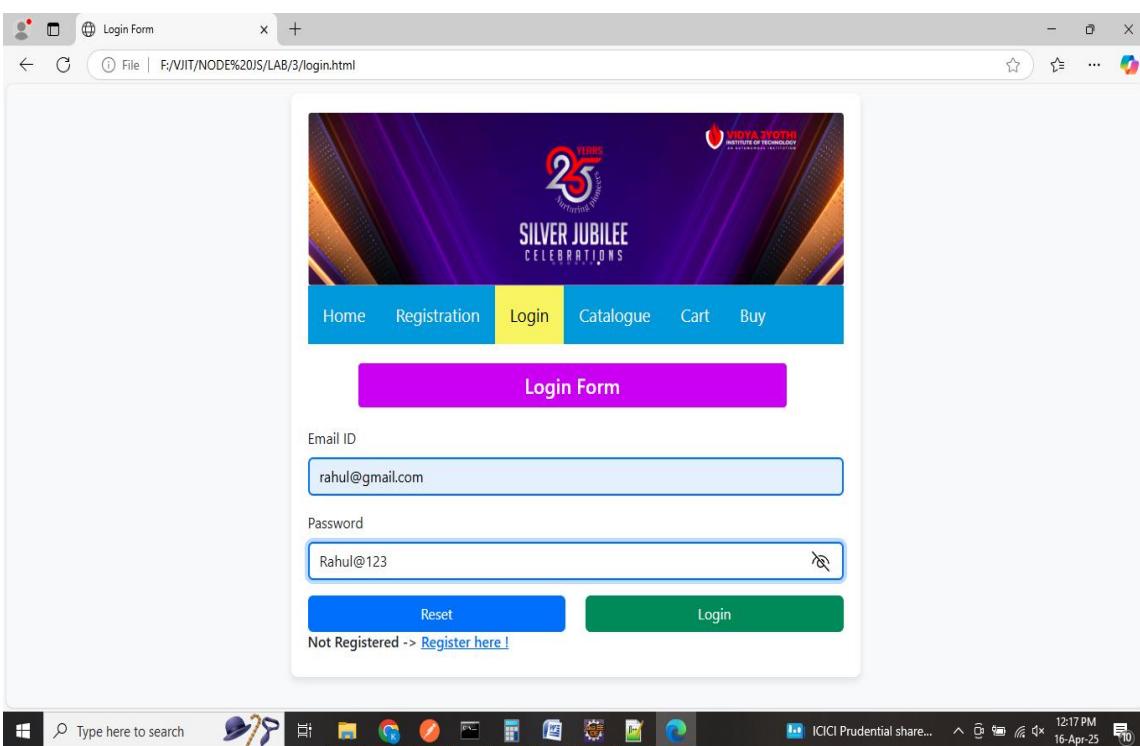
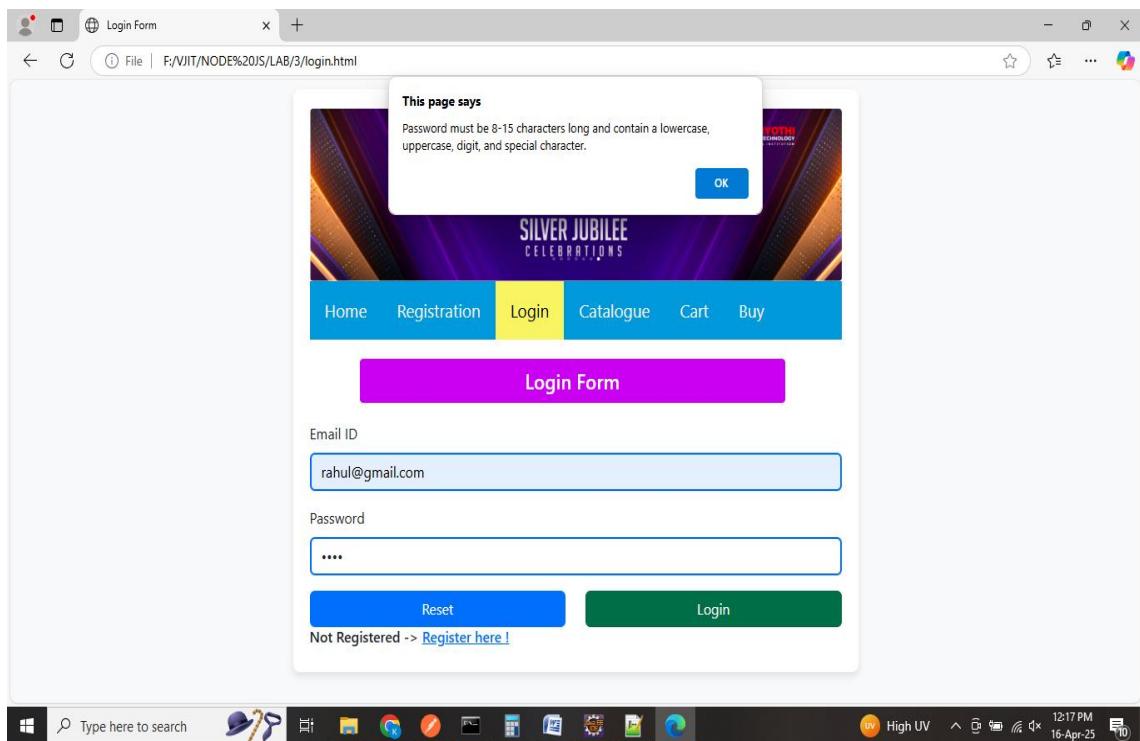
Already Registered ! ->[Login here !](#)



The screenshot shows a web browser window with the title "Registration Form". The URL in the address bar is "F:\VJIT\NODE%20JS\LAB\3\registration.html". The page header features the text "SILVER JUBILEE CELEBRATIONS". A navigation menu at the top includes links for Home, Registration (which is highlighted in yellow), Login, Catalogue, Cart, and Buy. Below the menu, a pink header bar contains the text "Registration Form". The main content area contains fields for Username (Rahul), Password (Rahul@123), Address (Enter your address), Gender (Male selected), Email ID (Enter your email), and Mobile Number (Enter your mobile number). At the bottom are "Reset" and "Register" buttons, and a link for "Already Registered! -> [Login here!](#)". The browser's taskbar at the bottom shows various pinned icons and the system tray indicates the date as 16-Apr-25.

This screenshot is identical to the one above, showing the same registration form for "SILVER JUBILEE CELEBRATIONS". The only difference is the address entered in the "Address" field, which now reads "Hyderabad". All other fields and the overall layout remain the same.





Login Form

File | F:/VJIT/NODE%20JS/LAB/3/login.html

25 YEARS
SILVER JUBILEE CELEBRATIONS

Home Registration Login Catalogue Cart Buy

Login Form

Email ID
rahu@gmail.com

Password
Rahul@123

Reset Login

Not Registered -> [Register here!](#)

Online Catalogue

This page says
HTML5 added to cart.

OK

SILVER JUBILEE CELEBRATIONS

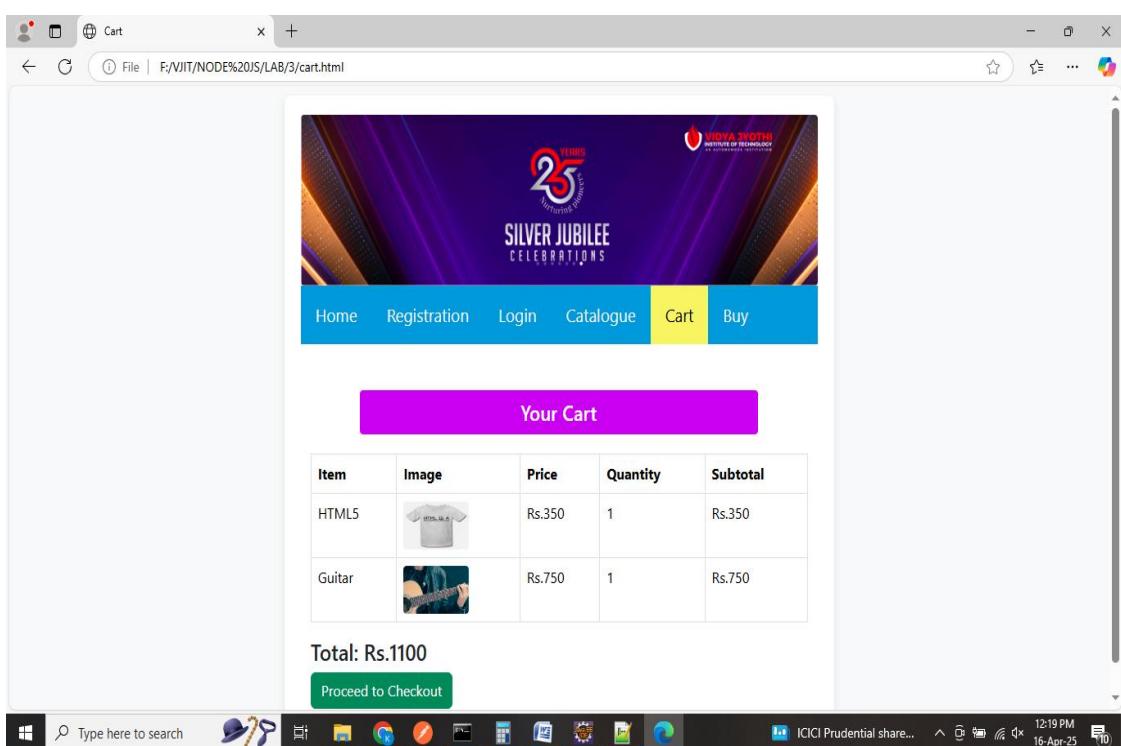
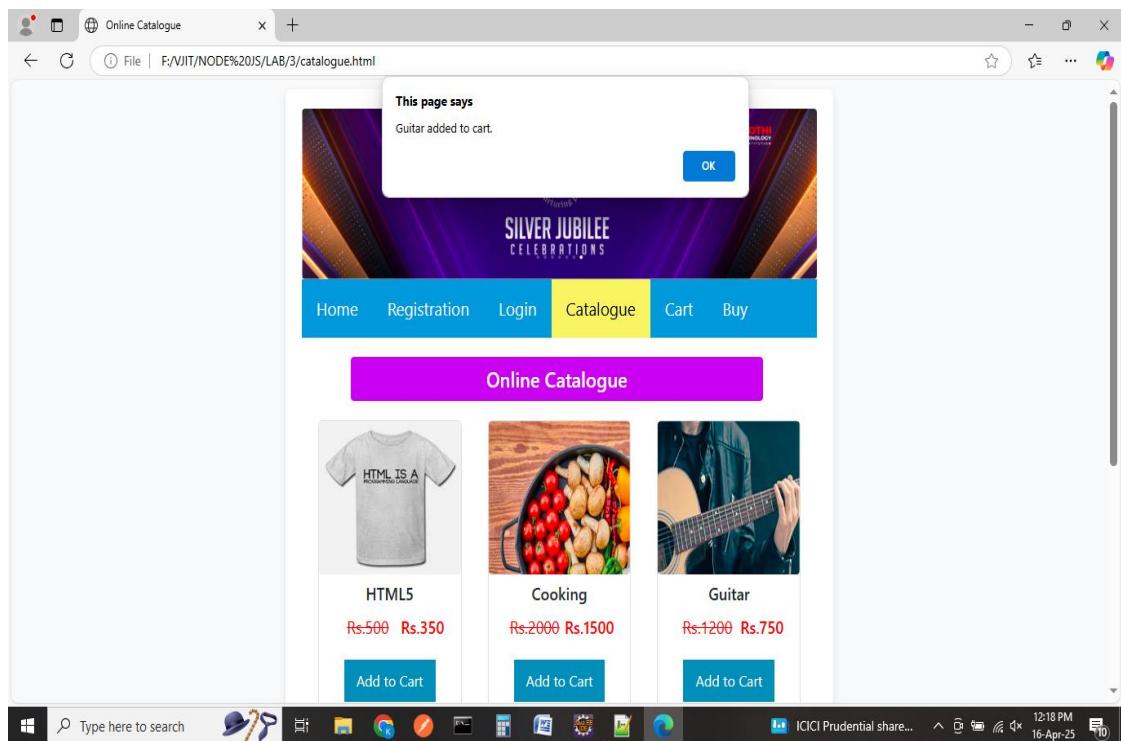
Home Registration Login Catalogue Cart Buy

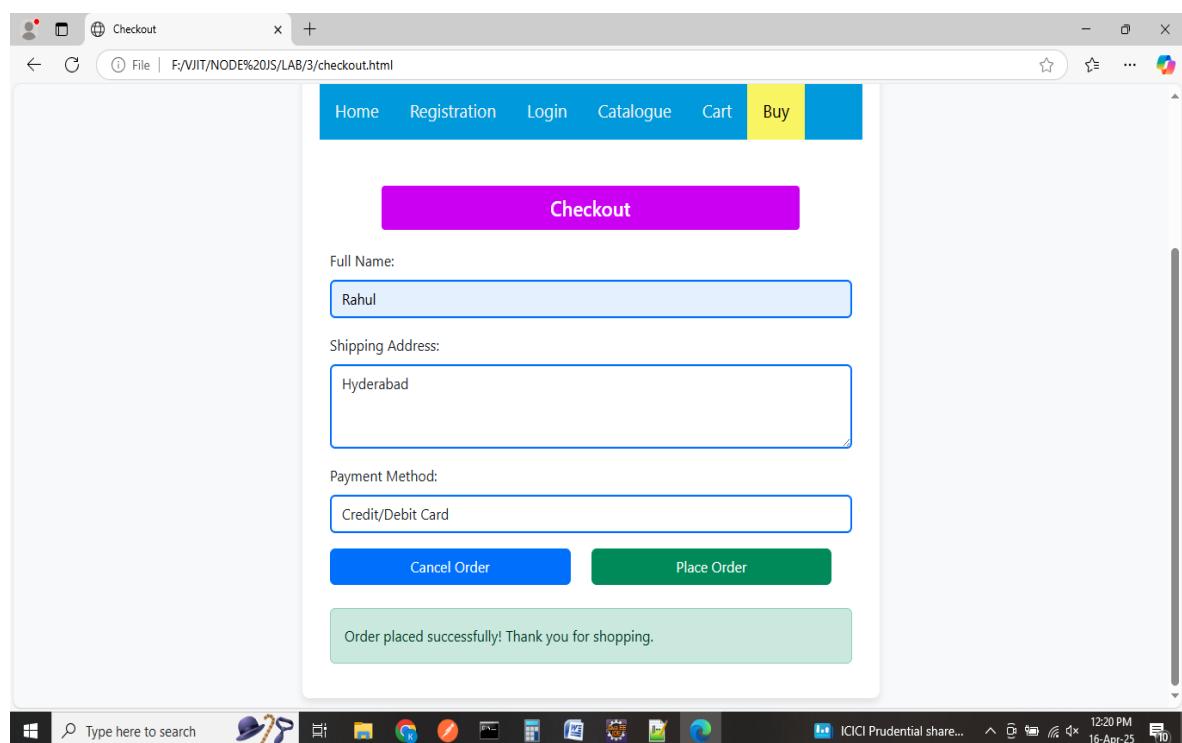
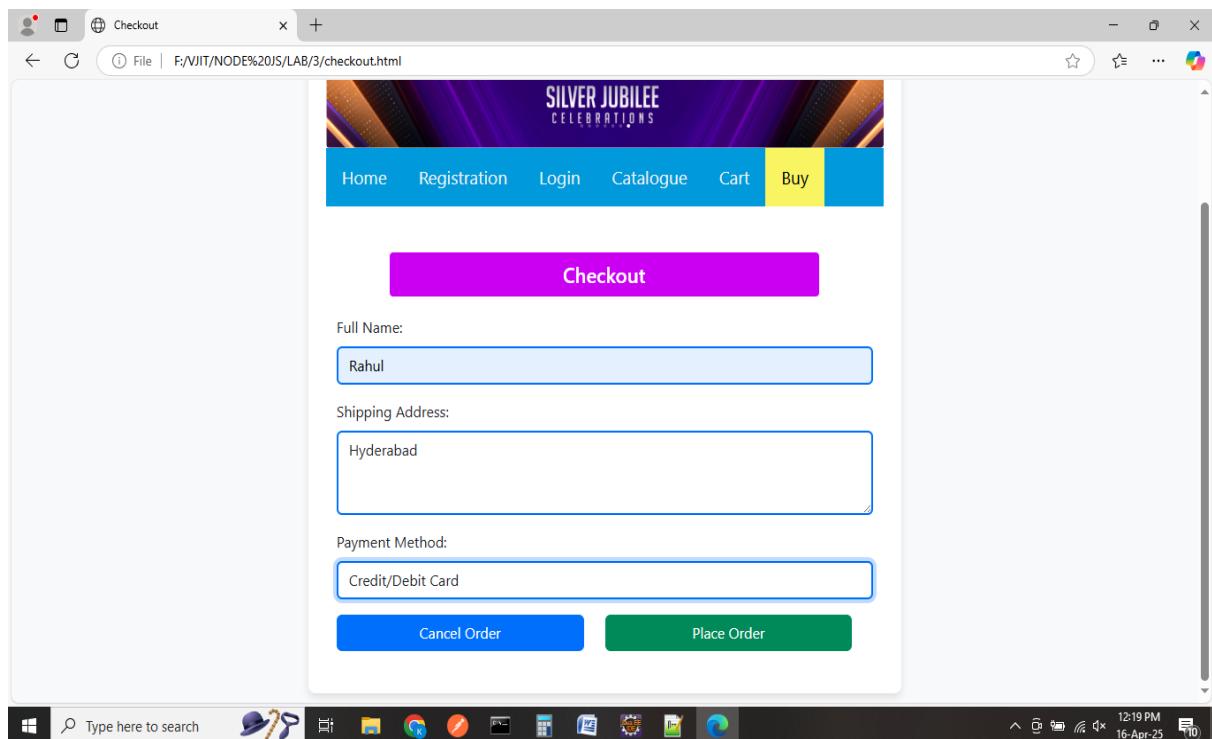
Online Catalogue

HTML5
Rs.500 Rs.350
[Add to Cart](#)

Cooking
Rs.2000 Rs.1500
[Add to Cart](#)

Guitar
Rs.1200 Rs.750
[Add to Cart](#)





4. Develop a Java Stand Alone Application that connects with the Database (Oracle / mySql) and perform the CRUD operation on the database tables.

4.1 Java Program to **CREATE** Database Table in MySQL Server

```
// This code is for establishing connection with MySQL database and retrieving data
// from database through JDBC
/*
 *1. import --->java.sql
 *2. load and register the driver ---> com.jdbc.
 *3. create connection
 *4. create a statement
 *5. execute the query
 *6. process the results
 *7. close
 */
import java.io.*;
import java.sql.*;

class create
{
    public static void main(String[] args) throws ClassNotFoundException,SQLException
    {
        String url= "jdbc:mysql://localhost:3306/vjit"; // Database details
        String username = "root"; // MySQL credentials
        String password = "vjit";
        Class.forName("com.mysql.cj.jdbc.Driver"); // Driver name
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("Connection Established successfully");
        Statement st = con.createStatement();
        String sql = "CREATE TABLE STUDENTS(Roll_No INTEGER NOT
NULL,First_Name VARCHAR(255),Last_Name VARCHAR(255),Age
INTEGER,PRIMARY KEY ( Roll_No ))";
        st.execute(sql);
        System.out.println("Created table in given database... ");
        st.close(); // close statement
        con.close(); // close connection
        System.out.println("Connection Closed....");
    }
}
```

Output:

```
MySQL 8.0 Command Line Client

mysql> show databases;
+-----+
| Database |
+-----+
| cseds   |
| customer-sale |
| employeedb |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| vjit     |
| world   |
+-----+
10 rows in set (0.01 sec)

mysql> use vjit;
Database changed
mysql>
```

```
Command Prompt

F:\VJIT\JAVA\LAB\Week13>javac create.java

F:\VJIT\JAVA\LAB\Week13>java create
Connection Established successfully
Created table in given database...
Connection Closed.....

F:\VJIT\JAVA\LAB\Week13>
```

```
MySQL 8.0 Command Line Client
mysql> show tables;
+-----+
| Tables_in_vjit |
+-----+
| allotment
| bill_details_view
| book
| cassette
| daily_issues_view
| daily_sales_view
| employee
| employee_adding
| employee_deletions
| employee_net_salary
| employee_pay_details
| iss_rec_view
| issue_details
| issues_date_wise
| salary_changes
| student_machine_allocations
| students
| thursday_machine_allocations |
+-----+
18 rows in set (0.00 sec)

mysql>
```

```
MySQL 8.0 Command Line Client
mysql> desc students;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Roll_No    | int        | NO   | PRI | NULL    |       |
| First_Name | varchar(255) | YES  |     | NULL    |       |
| Last_Name  | varchar(255) | YES  |     | NULL    |       |
| Age        | int        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

4.2 Java Program to **INSERT** Record into Database Table in MySql Server

```

import java.sql.*;
class StudentInsertionApplication
{
    public static void main(String[] args) throws ClassNotFoundException,SQLException
    {
        String url= "jdbc:mysql://localhost:3306/vjit"; // Database details
        String username = "root"; // MySQL credentials
        String password = "vjit";
        Class.forName("com.mysql.cj.jdbc.Driver"); // Driver name
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("Connection Established successfully");

        Statement st = con.createStatement();
        int c = st.executeUpdate("insert into students values(6703,'Rahul','HYD',20)");
        System.out.println(c + "\t Student Record inserted successfully");
        st.close();
        con.close();
    }
}

```

Output:

```

F:\VJIT\JAVA\LAB\Week13>javac insert.java
F:\VJIT\JAVA\LAB\Week13>java StudentInsertionApplication
Connection Established successfully
1      Student Record inserted successfully
F:\VJIT\JAVA\LAB\Week13>

```

```

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
| 6703 | Rahul | HYD | 20 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

4.3 Java Program to **UPDATE** Record into Database Table in MySql Server

```

import java.sql.*;
class UpdateStudentApplication
{
public static void main (String[]args) throws ClassNotFoundException, SQLException
{
    String url= "jdbc:mysql://localhost:3306/vjit"; // Database details
    String username = "root"; // MySQL credentials
    String password = "vjit";
    Class.forName("com.mysql.cj.jdbc.Driver"); // Driver name
    Connection con = DriverManager.getConnection(url, username, password);
    System.out.println("Connection Established successfully");

    Statement st = con.createStatement();

    int rows = st.executeUpdate("update students set First_Name = 'Rahul Dravid' where
    Roll_No=6703");
    System.out.println(rows + " row modified");
    st.close();
    con.close();
}
}

```

Output:

```

F:\VJIT\JAVA\LAB\Week13>javac update.java
F:\VJIT\JAVA\LAB\Week13>java UpdateStudentApplication
Connection Established successfully
1 row modified

F:\VJIT\JAVA\LAB\Week13>_

```

```

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
| 6703 | Rahul | HYD | 20 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
| 6703 | Rahul Dravid | HYD | 20 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>

```

4.4 Java Program to **DELETE** Record from Database Table in MySql Server

```
import java.sql.*;
import java.util.*;

class StudentDeleteApplication
{
    public static void main (String[]args) throws ClassNotFoundException, SQLException
    {
        String url= "jdbc:mysql://localhost:3306/vjit"; // Database details
        String username = "root"; // MySQL credentials
        String password = "vjit";
        Class.forName("com.mysql.cj.jdbc.Driver"); // Driver name
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("Connection Established successfully");

        Statement st = con.createStatement();
        Scanner sc = new Scanner(System.in);
        System.out.println("ENTER STUDENT NUMBER");
        int rno = sc.nextInt();

        int c = st.executeUpdate("delete from students where Roll_No =" + rno);
        if (c == 0)
            System.out.println("Student data does not exist");
        else
            System.out.println("Student data deleted successfully");
        st.close();
        con.close();
    }
}
```

Output:

```
C:\ Command Prompt
F:\VJIT\JAVA\LAB\Week13>java StudentDeleteApplication
Connection Established successfully
ENTER STUDENT NUMBER
6703
Student data deleted successfully

F:\VJIT\JAVA\LAB\Week13>
```

```
MySQL 8.0 Command Line Client

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
| 6703 | Rahul Dravid | HYD | 20 |
| 6706 | Ajay | TPT | 30 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
| 6706 | Ajay | TPT | 30 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

4.5 Java Program to **SELECT** Records from Database Table in MySql Server

```
import java.sql.*;  
import java.util.*;  
class StudentsDetails  
{  
    public static void main (String[]args) throws ClassNotFoundException, SQLException  
    {  
        String url= "jdbc:mysql://localhost:3306/vjit"; // Database details  
        String username = "root"; // MySQL credentials  
        String password = "vjit";  
        Class.forName("com.mysql.cj.jdbc.Driver"); // Driver name  
        Connection con = DriverManager.getConnection(url, username, password);  
        System.out.println("Connection Established successfully");  
        Statement st = con.createStatement();  
        System.out.println("\n\nStudent Details: ");  
        System.out.println("-----");  
        ResultSet rs = st.executeQuery("select * from students");  
        while(rs.next())  
        {  
            System.out.println("Roll No: " + rs.getInt(1));  
            System.out.println("First Name: " + rs.getString(2));  
            System.out.println("Last Name : " + rs.getString(3));  
            System.out.println("-----");  
        }  
        rs.close();  
        st.close();  
        con.close();  
    }  
}
```

Output:

```
Command Prompt
F:\VJIT\JAVA\LAB\Week13>javac read.java
F:\VJIT\JAVA\LAB\Week13>java StudentsDetails
Connection Established successfully

Student Details:
-----
Roll No:      6706
First Name:   Ajay
Last Name :   TPT
-----
F:\VJIT\JAVA\LAB\Week13>
```

```
MySQL 8.0 Command Line Client

mysql> select * from students;
+-----+-----+-----+-----+
| Roll_No | First_Name | Last_Name | Age |
+-----+-----+-----+-----+
|    6706 | Ajay        | TPT       |   30 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

5. Create an XML for the Bookstore. Validate the same using both DTD and XSD.

BookStore.xml

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

BookStore-noDTD.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Bookstore with no DTD-->
<Bookstore>
  <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
    <Title>A First Course in Database Systems</Title>
    <Authors>
      <Author>
        <First_Name>Jeffrey</First_Name>
        <Last_Name>Ullman</Last_Name>
      </Author>
      <Author>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Author>
    </Authors>
  </Book>
  <Book ISBN="ISBN-0-13-815504-6" Price="100">
    <Remark>
      Buy this book bundled with "A First Course" - a great deal!
    </Remark>
  </Book>
</Bookstore>
```

```

</Remark>
<Title>Database Systems: The Complete Book</Title>
<Authors>
  <Author>
    <First_Name>Hector</First_Name>
    <Last_Name>Garcia-Molina</Last_Name>
  </Author>
  <Author>
    <First_Name>Jeffrey</First_Name>
    <Last_Name>Ullman</Last_Name>
  </Author>
  <Author>
    <First_Name>Jennifer</First_Name>
    <Last_Name>Widom</Last_Name>
  </Author>
</Authors>
</Book>
</Bookstore>

```

BookStore-DTD.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Bookstore with DTD -->
<!-- Try changes: -->
<!-- Make Edition required -->
<!-- Swap order of First_Name, Last_Name -->
<!-- Add empty Remark -->
<!-- Add Magazine, omit closing tag -->
<!DOCTYPE Bookstore [
  <!ELEMENT Bookstore (Book | Magazine)*>
  <!ELEMENT Book (Title, Authors, Remark?)>
  <!ATTLIST Book ISBN CDATA #REQUIRED
    Price CDATA #REQUIRED
    Edition CDATA #IMPLIED>
  <!ELEMENT Magazine (Title)>
  <!ATTLIST Magazine Month CDATA #REQUIRED Year CDATA #REQUIRED>
  <!ELEMENT Title (#PCDATA)>
  <!ELEMENT Authors (Author+)>
  <!ELEMENT Remark (#PCDATA)>
  <!ELEMENT Author (First_Name, Last_Name)>
  <!ELEMENT First_Name (#PCDATA)>
  <!ELEMENT Last_Name (#PCDATA)>
]>

```

```

<Bookstore>
  <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
    <Title>A First Course in Database Systems</Title>
    <Authors>
      <Author>
        <First_Name>Jeffrey</First_Name>
        <Last_Name>Ullman</Last_Name>
      </Author>
      <Author>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Author>
    </Authors>
  </Book>
  <Book ISBN="ISBN-0-13-815504-6" Price="100">
    <Title>Database Systems: The Complete Book</Title>
    <Authors>
      <Author>
        <First_Name>Hector</First_Name>
        <Last_Name>Garcia-Molina</Last_Name>
      </Author>
      <Author>
        <First_Name>Jeffrey</First_Name>
        <Last_Name>Ullman</Last_Name>
      </Author>
      <Author>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Author>
    </Authors>
    <Remark>
      Buy this book bundled with "A First Course" - a great deal!
    </Remark>
  </Book>
</Bookstore>

```

BookStore-.xsd

```

<?xml version="1.0" ?>
<!-- XSD for Bookstore-XSD.xml --&gt;

&lt;xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"&gt;
  &lt;xsd:element name="Bookstore"&gt;
    &lt;xsd:complexType&gt;
      &lt;xsd:sequence&gt;
        &lt;xsd:element name="Book" type="BookType"&gt;
</pre>

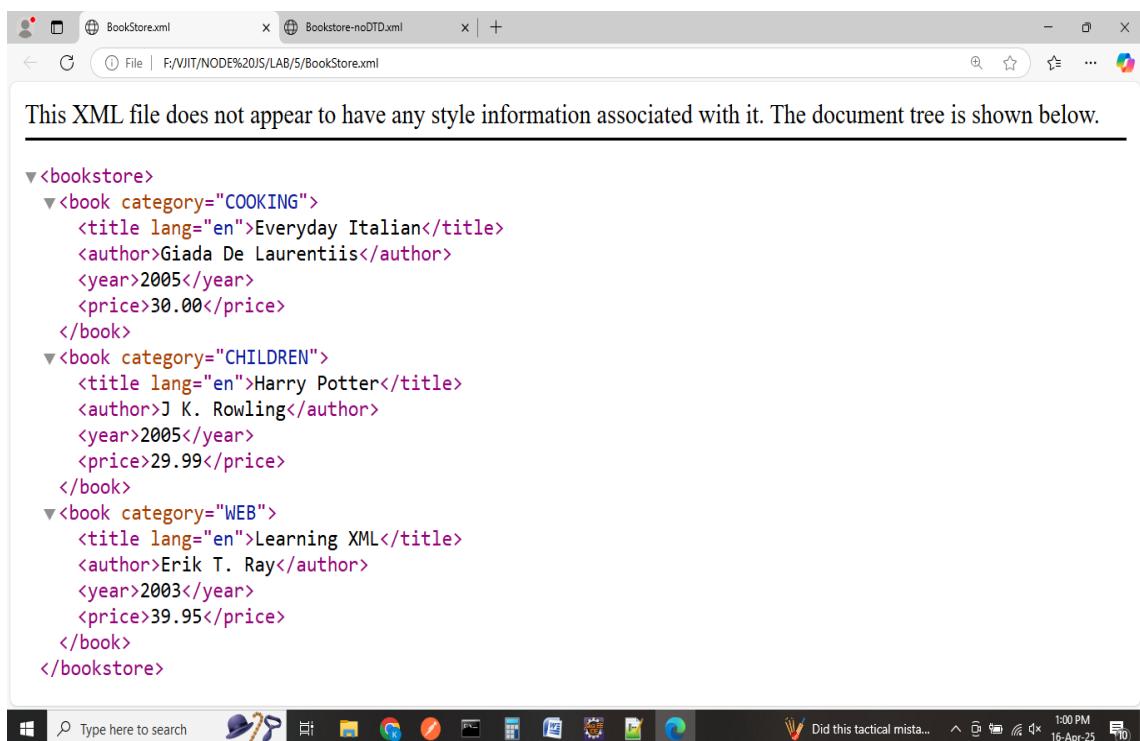
```

```
        minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="Author" type="AuthorType"
        minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:key name="BookKey">
    <xsd:selector xpath="Book" />
    <xsd:field xpath="@ISBN" />
</xsd:key>
<xsd:key name="AuthorKey">
    <xsd:selector xpath="Author" />
    <xsd:field xpath="@Ident" />
</xsd:key>
<xsd:keyref name="AuthorKeyRef" refer="AuthorKey">
    <xsd:selector xpath="Book/Authors/Auth" />
    <xsd:field xpath="@authIdent" />
</xsd:keyref>
<xsd:keyref name="BookKeyRef" refer="BookKey">
    <xsd:selector xpath="Book/Remark/BookRef" />
    <xsd:field xpath="@book" />
</xsd:keyref>
</xsd:element>
<xsd:complexType name="BookType">
    <xsd:sequence>
        <xsd:element name="Title" type="xsd:string" />
        <xsd:element name="Authors">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="Auth" maxOccurs="unbounded">
                        <xsd:complexType>
                            <xsd:attribute name="authIdent" type="xsd:string"
                                use="required" />
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Remark" minOccurs="0">
    <xsd:complexType mixed="true">
        <xsd:sequence>
            <xsd:element name="BookRef" minOccurs="0"
                maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:attribute name="book" type="xsd:string"
                        use="required" />

```

```
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="ISBN" type="xsd:string" use="required" />
<xsd:attribute name="Price" type="xsd:integer" use="required" />
</xsd:complexType>
<xsd:complexType name="AuthorType">
  <xsd:sequence>
    <xsd:element name="First_Name" type="xsd:string" />
    <xsd:element name="Last_Name" type="xsd:string" />
  </xsd:sequence>
  <xsd:attribute name="Ident" type="xsd:string" use="required" />
</xsd:complexType>
</xsd:schema>
```

Output:



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<bookstore>
  ▼<book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  ▼<book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ▼<book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!-- Bookstore with no DTD -->
<!DOCTYPE Bookstore>
<Bookstore>
  <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
    <Title>A First Course in Database Systems</Title>
    <Authors>
      <Author>
        <First_Name>Jeffrey</First_Name>
        <Last_Name>Ullman</Last_Name>
      </Author>
      <Author>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Author>
    </Authors>
  </Book>
  <Book ISBN="ISBN-0-13-815504-6" Price="100">
    <Remark> Buy this book bundled with "A First Course" - a great deal! </Remark>
    <Title>Database Systems: The Complete Book</Title>
    <Authors>
      <Author>
        <First_Name>Hector</First_Name>
      </Author>
    </Authors>
  </Book>
</Bookstore>

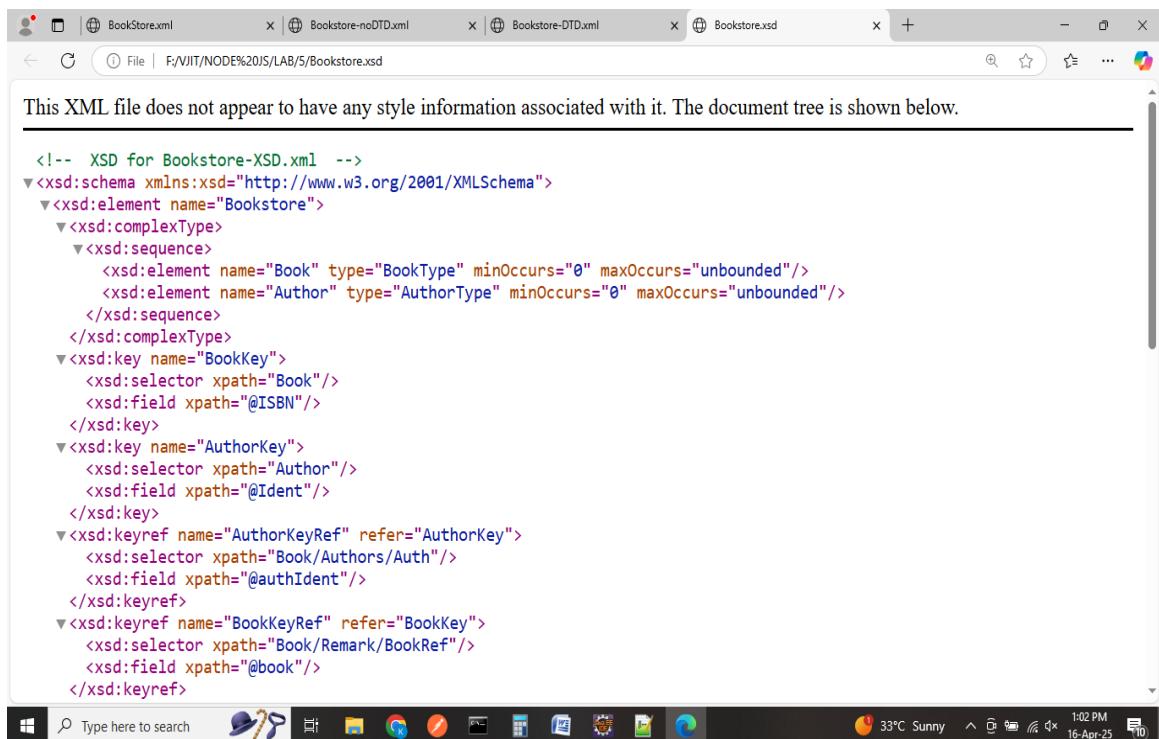
```

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!-- Bookstore with DTD -->
<!-- Try changes: -->
<!-- Make Edition required -->
<!-- Swap order of First_Name, Last_Name -->
<!-- Add empty Remark -->
<!-- Add Magazine, omit closing tag -->
<!DOCTYPE Bookstore>
<Bookstore>
  <Book ISBN="ISBN-0-13-713526-2" Price="85" Edition="3rd">
    <Title>A First Course in Database Systems</Title>
    <Authors>
      <Author>
        <First_Name>Jeffrey</First_Name>
        <Last_Name>Ullman</Last_Name>
      </Author>
      <Author>
        <First_Name>Jennifer</First_Name>
        <Last_Name>Widom</Last_Name>
      </Author>
    </Authors>
  </Book>
  <Book ISBN="ISBN-0-13-815504-6" Price="100">
    <Title>Database Systems: The Complete Book</Title>
    <Authors>
      <Author>
        <First_Name>Hector</First_Name>
      </Author>
    </Authors>
  </Book>
</Bookstore>

```

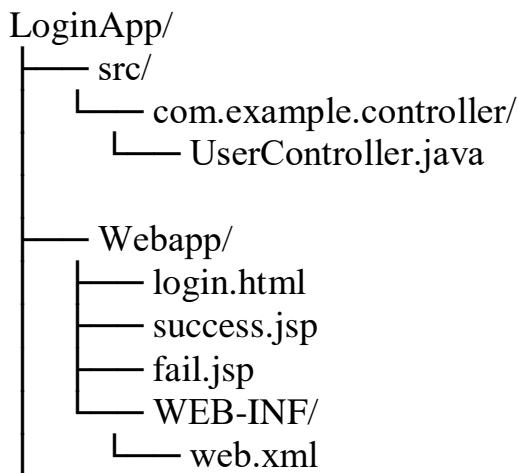


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- XSD for Bookstore-XSD.xml -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Bookstore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Book" type="BookType" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element name="Author" type="AuthorType" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:key name="BookKey">
      <xsd:selector xpath="Book"/>
      <xsd:field xpath="@ISBN"/>
    </xsd:key>
    <xsd:key name="AuthorKey">
      <xsd:selector xpath="Author"/>
      <xsd:field xpath="@Ident"/>
    </xsd:key>
    <xsd:keyref name="AuthorKeyRef" refer="AuthorKey">
      <xsd:selector xpath="Book/Authors/Auth"/>
      <xsd:field xpath="@authIdent"/>
    </xsd:keyref>
    <xsd:keyref name="BookKeyRef" refer="BookKey">
      <xsd:selector xpath="Book/Remark/BookRef"/>
      <xsd:field xpath="@book"/>
    </xsd:keyref>
```

6. Design a Controller with Servlet that provides the interaction with Web Application

Project Structure



1. Create a Dynamic Web Project

In Eclipse:

- File > New > Dynamic Web Project
- Name it (e.g., LoginApp)
- Target Runtime: Apache Tomcat
- Click Finish

2. Create Servlet Controller

UserController.java

```
package com.example.controller;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class UserController extends HttpServlet
{
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        // Get form data
    }
}
```

```

String username = request.getParameter("uname");
String password = request.getParameter("pass");

// Simple validation
if ("admin".equals(username) && "admin123".equals(password)) {
    request.setAttribute("message", "Login Successful!");
    RequestDispatcher rd = request.getRequestDispatcher("success.jsp");
    rd.forward(request, response);
} else {
    request.setAttribute("error", "Invalid credentials!");
    RequestDispatcher rd = request.getRequestDispatcher("fail.jsp");
    rd.forward(request, response);
}
}
}
}

```

3. Configure Servlet in web.xml

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
  version="4.0">

    <display-name>LoginApp</display-name>
    <welcome-file-list>
        <welcome-file>login.html</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.jsp</welcome-file>
        <welcome-file>default.htm</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>UserController</servlet-name>
        <servlet-class>com.example.controller.UserController</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>UserController</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
</web-app>

```

4. HTML Form to Post to Servlet

login.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <h2>Login Page</h2>
    <form action="login" method="post">
        Username: <input type="text" name="uname" required><br><br>
        Password: <input type="password" name="pass" required><br><br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

5. Success Page

success.jsp

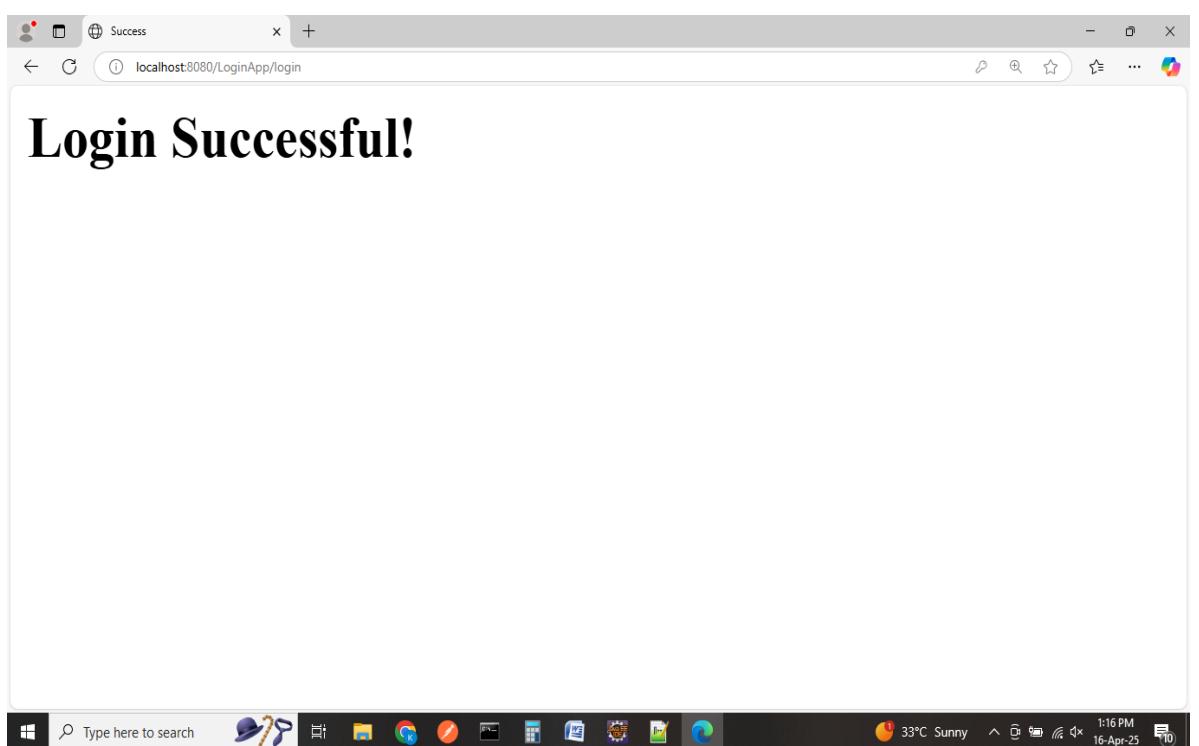
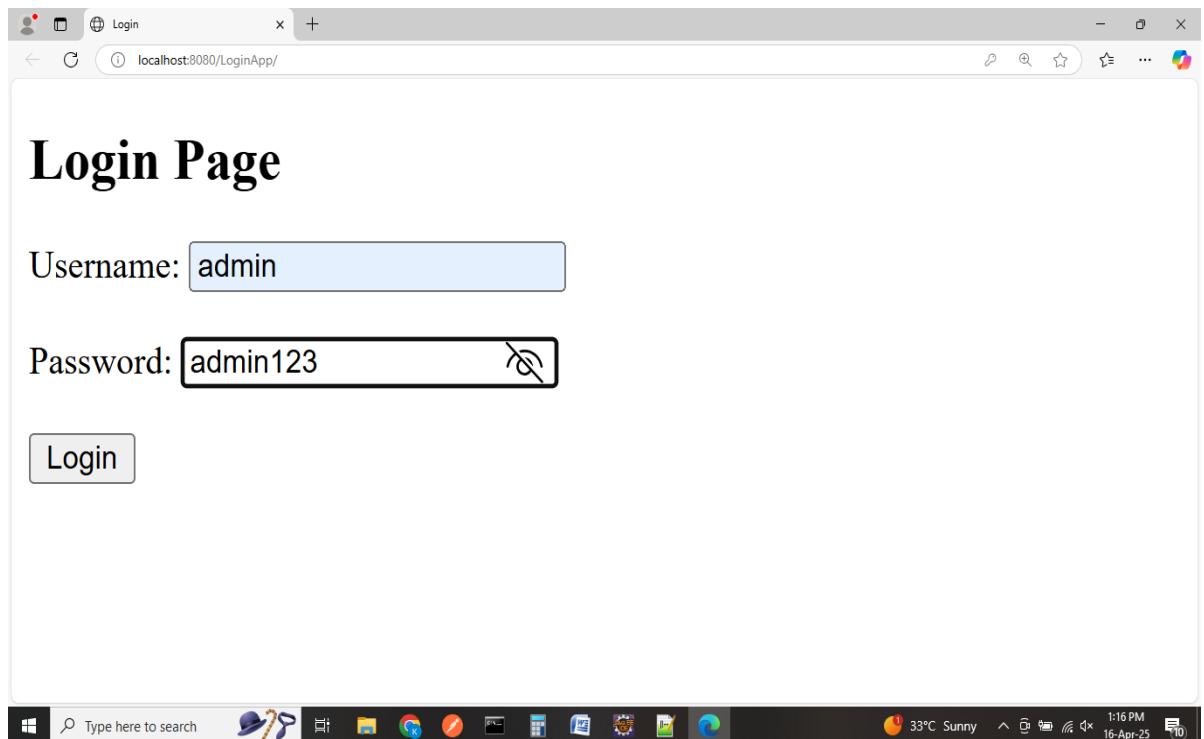
```
<%@ page language='java' contentType='text/html; charset=UTF-8' %>
<html>
<head><title>Success</title></head>
<body>
    <h2>${message}</h2>

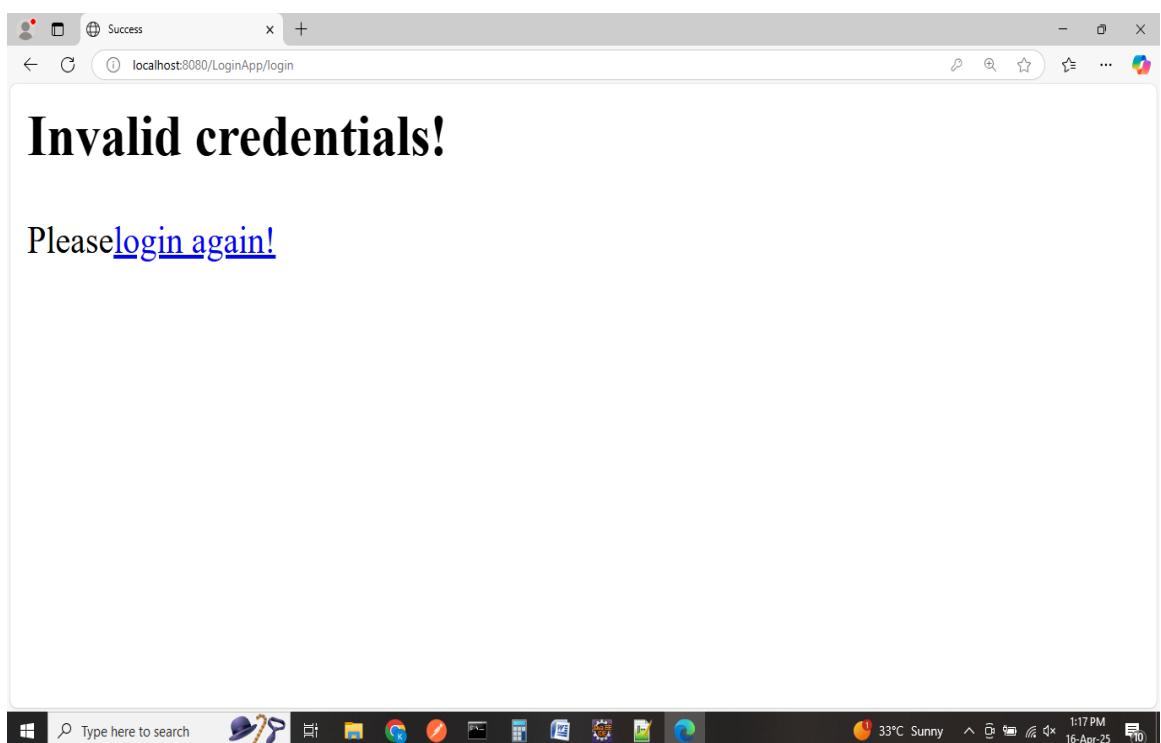
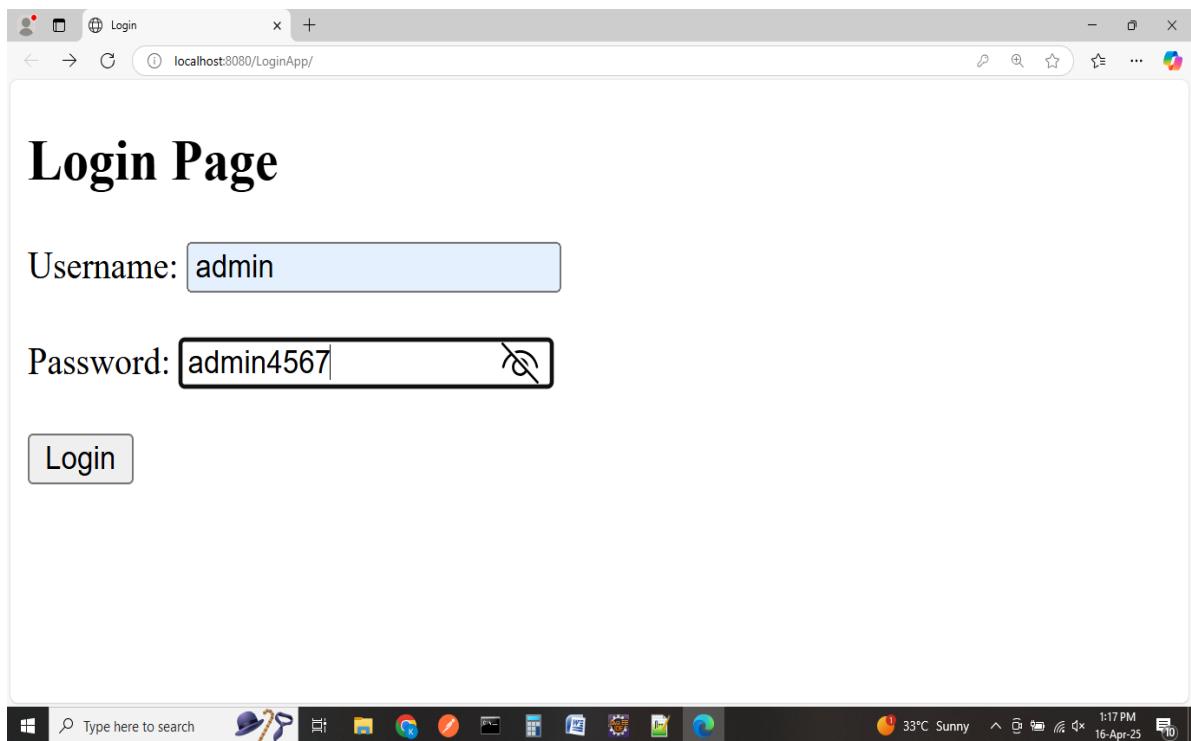
</body>
</html>
```

6. Failure Page

fail.jsp

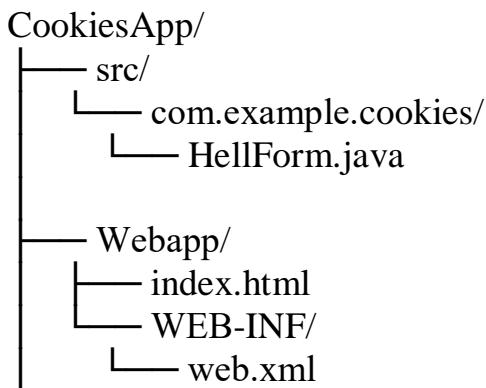
```
<%@ page language='java' contentType='text/html; charset=UTF-8' %>
<html>
<head><title>Success</title></head>
<body>
    <h2>${error}</h2>
    Please<a href="login.html">login again!</a>
</body>
</html>
```

Output:



7. Maintaining the transactional history of any user is very important. Explore the various Session Tracking mechanism using **Cookies**

Project Structure



1. Create a Dynamic Web Project

In Eclipse:

- File > New > Dynamic Web Project
 - Name it (e.g., CookiesApp)
 - Target Runtime: Apache Tomcat
 - Click Finish
-

2. Create Servlet Controller

HellForm.java

```
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
/**  
 * Servlet implementation class HelloForm  
 */  
@WebServlet("/HelloForm")
```

```
public class HelloForm extends HttpServlet
{
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // Create cookies for first and last names.
        Cookie firstName = new Cookie("first_name", request.getParameter("first_name"));
        Cookie lastName = new Cookie("last_name", request.getParameter("last_name"));

        // Set expiry date after 24 Hrs for both the cookies.
        firstName.setMaxAge(60*60*24);
        lastName.setMaxAge(60*60*24);

        // Add both the cookies in the response header.
        response.addCookie( firstName );
        response.addCookie( lastName );
        Cookie cookie = null;
        Cookie[] cookies = null;

        // Get an array of Cookies associated with this domain
        cookies = request.getCookies();

        // Set response content type
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String title = "Reading Cookies Example";
        String docType =
            "<!doctype html public "-//w3c//dtd html 4.0 " +
            "transitional//en\\>\\n";
        out.println(docType +
                    "<html>\\n" +
                    "<head><title>" + title + "</title></head>\\n" +
                    "<body bgcolor = \"#f0f0f0\">\\n" );

        if( cookies != null )
        {
            out.println("<h2> Found Cookies Name and Value</h2>");
            out.println("<table border=\"1\">\\r\\n" + "<tr><th>Cookie Name</th><th>Cookie
Value</th></tr>");
            for (int i = 0; i < cookies.length; i++)
            {
                cookie = cookies[i];
                out.print("<tr><td> " + cookie.getName() + "</td>");
                out.print("<td> " + cookie.getValue() + " </td><br/>");
            }
        }
    }
}
```

```

        out.println("</table>");
    } else {
        out.println("<h2>No cookies founds</h2>");
    }
    out.println("</body>");
    out.println("</html>");

}
}

```

3. Configure Servlet in web.xml

web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xmlns='http://xmlns.jcp.org/xml/ns/javaee'
  xsi:schemaLocation='http://xmlns.jcp.org/xml/ns/javaee
  http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd' id='WebApp_ID'
  version='3.1'>
    <display-name>Cookies</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.jsp</welcome-file>
        <welcome-file>default.htm</welcome-file>
    </welcome-file-list>

</web-app>

```

4. HTML Form to Post to Servlet

index.html

```

<html>
  <body>
    <form action = "HelloForm" method = "GET">
      First Name: <input type = "text" name = "first_name">
      <br />
      Last Name: <input type = "text" name = "last_name" />
      <input type = "submit" value = "Submit" />
    </form>
  </body>
</html>

```

Output:

A screenshot of a web browser window. The address bar shows "localhost:8080/Cookies/". The page contains a form with two input fields and a submit button. The first field is labeled "First Name:" and contains the value "Rahul". The second field is labeled "Last Name:" and contains the value "Dravid". To the right of these fields is a "Submit" button.

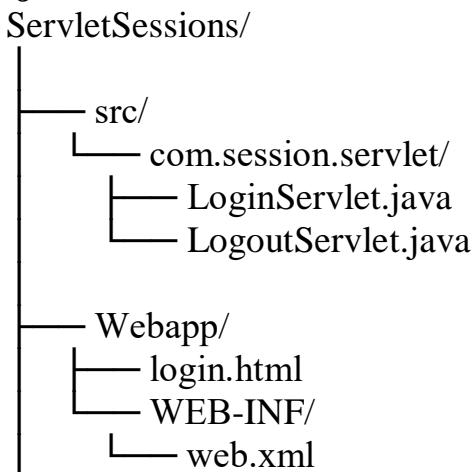
A screenshot of a web browser window. The address bar shows "localhost:8080/Cookies/HelloForm?first_name=Rahul&last_name=Dravid". The page displays the text "No cookies founds" in a large, bold font.

A screenshot of a web browser window. The address bar shows "localhost:8080/Cookies/HelloForm?first_name=Rahul&last_name=Dravid". The page displays the text "Found Cookies Name and Value" in a large, bold font. Below this, there is a table showing the cookie names and their corresponding values.

| Cookie Name | Cookie Value |
|-------------|--------------|
| first_name | Rahul |
| last_name | Dravid |

8. Maintaining the transactional history of any user is very important. Explore the various Session Tracking mechanism using **Sessions**.

Project Structure



LoginServlet.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.time.LocalTime;

public class LoginServlet extends HttpServlet
{
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // Get the name from the request
        String name = request.getParameter("name");

        // Create a new session or get the existing one
        HttpSession session = request.getSession();

        // Store the name and current time in the session
        session.setAttribute("name", name);
        session.setAttribute("startTime", System.currentTimeMillis()); // Current time in
millisecods

        // Set content type and response writer
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
    }
}
  
```

```

// Generate the welcome page
out.println("<html>");
out.println("<head><title>Welcome Page</title></head>");
out.println("<body>");
out.println("<h1>Hello " + name + "</h1>");

// Display the start time in the top-right corner
out.println("<div style='position: absolute; top: 10px; right: 10px;'>Start Time: "
+ LocalTime.now() + "</div>");

// Button for logging out
out.println("<form action='LogoutServlet' method='POST'>");
out.println("<input type='submit' value='Logout'>");
out.println("</form>");

out.println("</body>");
out.println("</html>");
}
}

```

LogoutServlet.java

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LogoutServlet extends HttpServlet
{
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
    {
        // Get the session
        HttpSession session = request.getSession(false);

        if (session != null)
        {
            // Get the user's name and session start time from the session
            String name = (String) session.getAttribute("name");
            long startTime = (Long) session.getAttribute("startTime");

            // Calculate the session duration in seconds
            long duration = (System.currentTimeMillis() - startTime) / 1000; // in seconds

            // Invalidate the session (log out)
            session.invalidate();
        }
    }
}

```

```

// Set content type and response writer
response.setContentType("text/html");
PrintWriter out = response.getWriter();

// Display the Thank You message with the session duration
out.println("<html>");
out.println("<head><title>Logout Page</title></head>");
out.println("<body>");
out.println("<h1>Thank You " + name + "</h1>");
out.println("<p>Your session lasted for " + duration + " seconds.</p>");
out.println("</body>");
out.println("</html>");
}
}
}

```

login.html

```

<!DOCTYPE html>
<html>
<head>
<title>Login Page</title>
<style>
body
{
    font-family: Arial, sans-serif;
}
.form-container
{
    text-align: center; margin-top: 50px;
}
input[type="text"]
{
    padding: 8px; font-size: 16px;
}
input[type="submit"]
{
    padding: 10px 20px; font-size: 16px; background-color: #4CAF50;
    color: white; border: none; cursor: pointer;
}
</style>
</head>
<body>
<div class="form-container">
<h2>Please Enter Your Name</h2>
<form action="LoginServlet" method="POST">

```

```

<input type="text" name="name" required placeholder="Enter your
name">
<input type="submit" value="Submit">
</form>
</div>
</body>
</html>

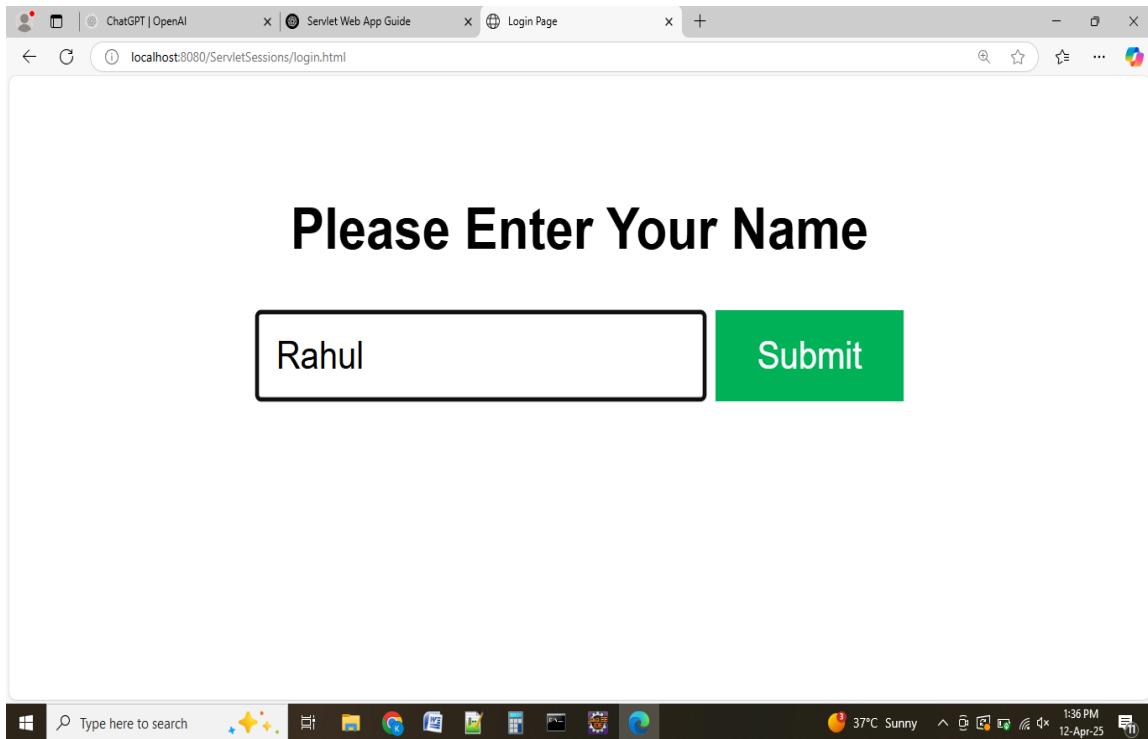
```

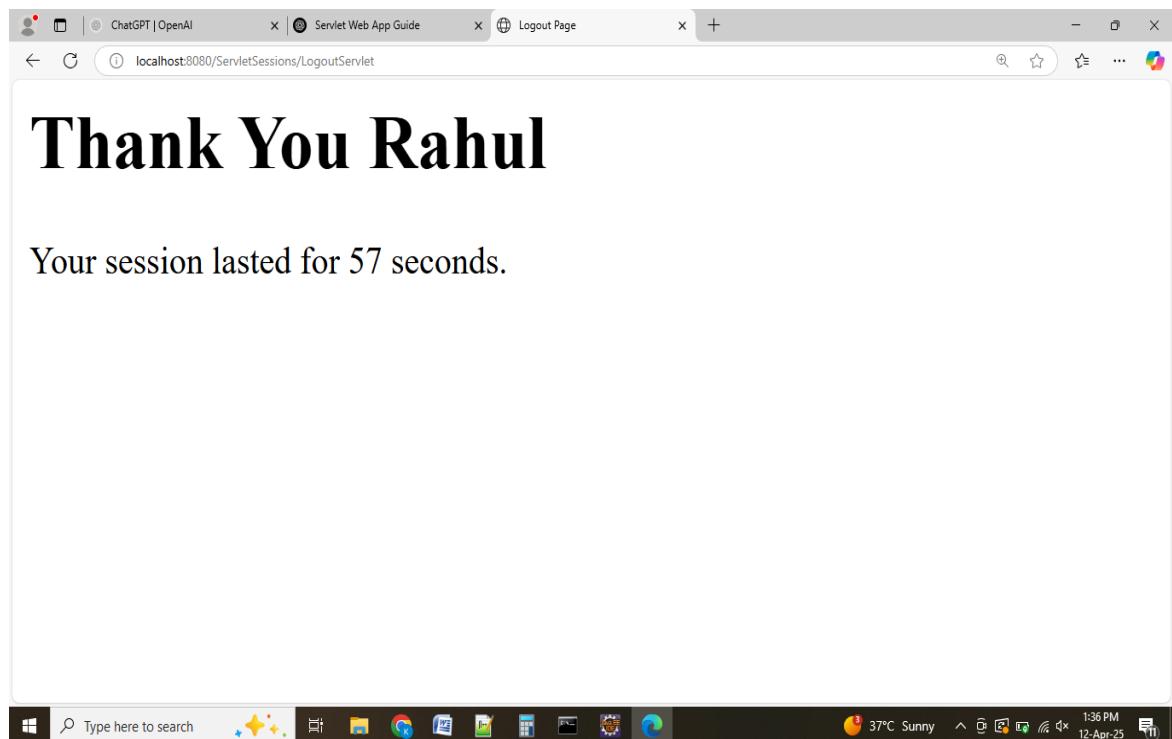
web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
version="4.0">
<display-name>ServletSessions</display-name>
<welcome-file-list>
<welcome-file>login.html</welcome-file>
<welcome-file>index.jsp</welcome-file>
<welcome-file>index.htm</welcome-file>
<welcome-file>default.html</welcome-file>
<welcome-file>default.jsp</welcome-file>
<welcome-file>default.htm</welcome-file>
</welcome-file-list>
<!-- Login Servlet Configuration -->
<servlet>
<servlet-name>LoginServlet</servlet-name>
<servlet-class>LoginServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LoginServlet</servlet-name>
<url-pattern>/LoginServlet</url-pattern>
</servlet-mapping>
<!-- Logout Servlet Configuration -->
<servlet>
<servlet-name>LogoutServlet</servlet-name>
<servlet-class>LogoutServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>LogoutServlet</servlet-name>
<url-pattern>/LogoutServlet</url-pattern>
</servlet-mapping>
</web-app>

```

Output:



9. Develop an **Express Web Application** that can interact with **REST API** to perform **CRUD** operations on Student Data

Step 1: Initialize the Express App

```
mkdir student-crud-app  
cd student-crud-app  
npm init -y  
npm install express
```

Step 2: Create In-Memory Student Data

```
let students = [  
  { id: 1, name: 'Rahul', age: 40 },  
  { id: 2, name: 'Rohith', age: 36 }  
];
```

Step 3: Define REST API Routes (CRUD)

1. GET All Students (Read All)

```
app.get('/students', (req, res) => {  
  res.json(students);  
});
```

2. GET a Student by ID (Read One)

```
app.get('/students/:id', (req, res) => {  
  const student = students.find(s => s.id === req.params.id);  
  if (!student) return res.status(404).json({ error: 'Student not found' });  
  res.json(student);  
});
```

3. POST a New Student (Create)

```
app.post('/students', (req, res) => {  
  const { name, age } = req.body;  
  const newStudent = {  
    id: students.length + 1,  
    name,  
    age
```

```

    };
    students.push(newStudent);
    res.status(201).json(newStudent);
});

```

4. PUT to Update Student (Update)

```

app.put('/students/:id', (req, res) => {
  const student = students.find(s => s.id === req.params.id);
  if (!student) return res.status(404).json({ error: 'Student not found' });

  const { name, age } = req.body;
  student.name = name || student.name;
  student.age = age || student.age;
  res.json(student);
});

```

5. DELETE a Student (Delete)

```

app.delete('/students/:id', (req, res) => {
  students = students.filter(s => s.id !== req.params.id);
  res.json({ message: 'Student deleted' });
});

```

Step 4: index.js

```

const express = require('express');
const app = express();
const PORT = 3000;

app.use(express.json());

let students = [
  { id: 1, name: 'Alice', age: 20 },
  { id: 2, name: 'Bob', age: 22 }
];

app.get('/students', (req, res) => res.json(students));

app.get('/students/:id', (req, res) => {
  const student = students.find(s => s.id === req.params.id);
  if (!student) return res.status(404).json({ error: 'Student not found' });
  res.json(student);
});

```

```
app.post('/students', (req, res) => {
  const { name, age } = req.body;
  const newStudent = { id: students.length + 1, name, age };
  students.push(newStudent);
  res.status(201).json(newStudent);
});

app.put('/students/:id', (req, res) => {
  const student = students.find(s => s.id === req.params.id);
  if (!student) return res.status(404).json({ error: 'Student not found' });

  student.name = req.body.name || student.name;
  student.age = req.body.age || student.age;
  res.json(student);
});

app.delete('/students/:id', (req, res) => {
  students = students.filter(s => s.id !== req.params.id);
  res.json({ message: 'Student deleted' });
});

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

Step 5: How to Test

You can use tools like:

- [Postman](#) or
- curl from terminal

What Is Postman?

Postman is a free and powerful tool for testing REST APIs. It allows you to:

- Send requests (GET, POST, PUT, DELETE, etc.)
- Add headers and request bodies
- See responses from the server
- Save and organize test collections

Step 6: Setup

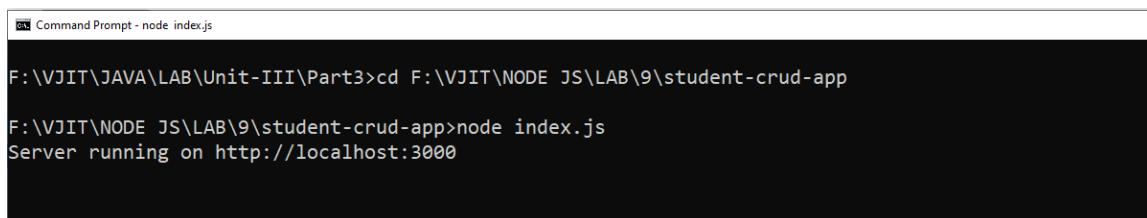
- Your Express.js server should be running.

For example:

node index.js

- Your API endpoint might look like this:

http://localhost:3000/students



```
Command Prompt - node index.js
F:\VJIT\JAVA\LAB\Unit-III\Part3>cd F:\VJIT\NODE JS\LAB\9\student-crud-app
F:\VJIT\NODE JS\LAB\9\student-crud-app>node index.js
Server running on http://localhost:3000
```

- You've installed Postman on your computer.

Step 7: Testing CRUD Operations with Postman

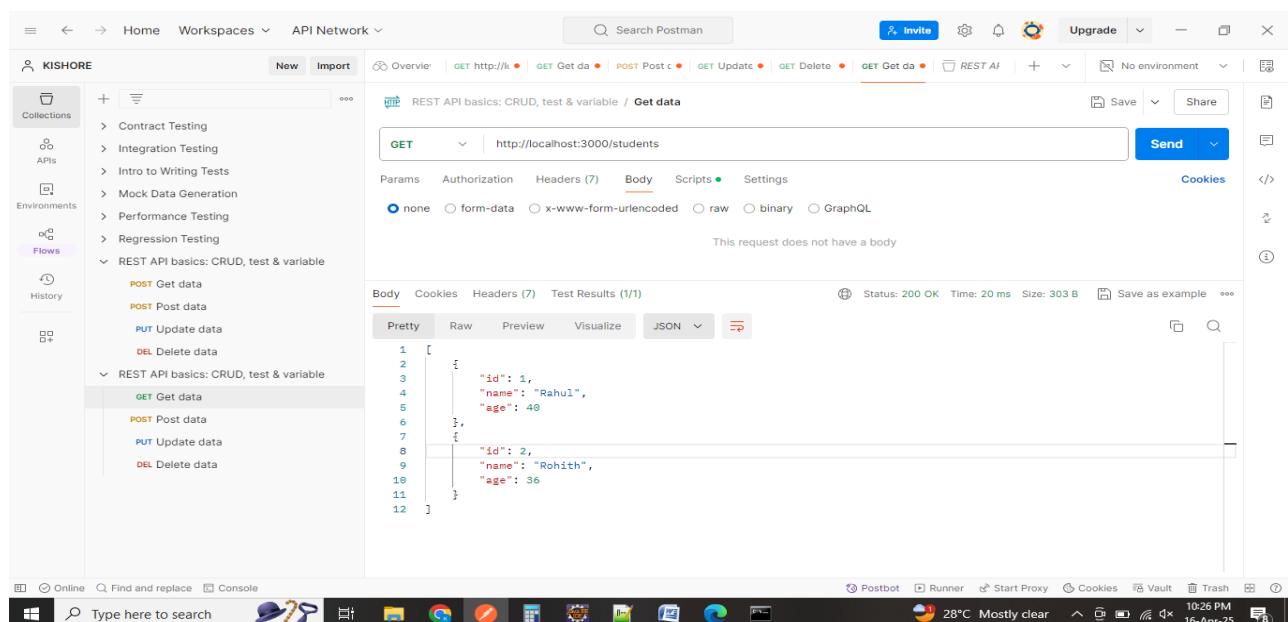
1. GET All Students

Retrieve the full list of students.

- Method:** GET
- URL:** <http://localhost:3000/students>

Steps:

- Open Postman.
- Create a new request tab.
- Set **method** to GET.
- Enter the **URL**: <http://localhost:3000/students>
- Click **Send**.



The screenshot shows the Postman interface with a GET request to <http://localhost:3000/students>. The response body is displayed as follows:

```
[{"id": 1, "name": "Rahul", "age": 40}, {"id": 2, "name": "Rohith", "age": 36}]
```

2. GET a Student by ID

Get details of a single student (e.g., ID = 1)

- **Method:** GET
- **URL:** <http://localhost:3000/students/1>

The screenshot shows the Postman interface with a successful API call. The URL is `http://localhost:3000/students/1`. The response body is:

```

1 {
2   "id": 1,
3   "name": "Rahul",
4   "age": 40
5 }

```

If the ID doesn't exist, you'll get:

The screenshot shows the Postman interface with an unsuccessful API call. The URL is `http://localhost:3000/students/5`. The response body is:

```

1 {
2   "error": "Student not found"
3 }

```

3. POST a New Student (Create)

Add a new student record.

- **Method:** POST
- **URL:** <http://localhost:3000/students>

Steps:

1. Set method to POST
2. Go to the **Body** tab
3. Select **raw** and choose **JSON** from dropdown
4. Add this JSON:

```
{
  "name": "Kohli",
  "age": 35
}
```

5. Click **Send**

The screenshot shows the Postman application interface. On the left, there's a sidebar with collections like 'Contract Testing', 'Integration Testing', etc., and environments. The main workspace shows a 'POST' request to 'http://localhost:3000/students'. The 'Body' tab is selected, showing raw JSON input:

```

1 {
2   "name": "Kohli",
3   "age": 35
4 }
```

The response section shows a status of '201 Created' with a response time of '574 ms' and a size of '272 B'. The response body is also displayed in JSON format:

```

1 {
2   "id": 3,
3   "name": "Kohli",
4   "age": 35
5 }
```

4. PUT to Update a Student

Update an existing student (e.g., ID = 2)

- **Method:** PUT
- **URL:** <http://localhost:3000/students/2>

Steps:

1. Set method to PUT
2. Go to the **Body** tab > raw > JSON
3. Example:

```
{
  "name": "Ramesh",
  "age": 23
}
```

4. Click **Send**

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, APIs, environments, flows, and history. Under the 'REST API basics: CRUD, test & variable' collection, there are five items: POST Get data, POST Post data, PUT Update data, DEL Delete data, and GET Get data. The 'PUT Update data' item is currently selected. The main workspace shows a PUT request to `http://localhost:3000/students/2`. The 'Body' tab is selected, displaying the JSON payload:

```

1 {
2   "name": "Ramesh",
3   "age": 23
4 }
5
6

```

Below the request, the response status is 200 OK, time is 5 ms, size is 268 B. The response body is:

```

1 {
2   "id": 2,
3   "name": "Ramesh",
4   "age": 23
5 }

```

5. DELETE a Student

Delete a student by ID

- Method: DELETE**
- URL: `http://localhost:3000/students/2`**

The screenshot shows the Postman application interface, identical to the previous one but with a different request method. The 'DELETE' method is selected for the request to `http://localhost:3000/students/2`. The 'Body' tab is selected, displaying the same JSON payload as before:

```

1 {
2   "name": "Ramesh",
3   "age": 23
4 }
5
6

```

Below the request, the response status is 200 OK, time is 14 ms, size is 264 B. The response body is:

```

1 {
2   "message": "Student deleted"
3 }

```

ID not found:

```
{ "error": "Student not found" }
```

10. Create a **REACT** Application for the Student Management System having Registration, Login, Contact, About pages and implement ROUTING to navigate through these pages.

Node.js Installation

1. Check if Node.js and npm are Installed

First, verify whether Node.js and npm are installed on your system by running the following commands:

```
node -v  
npm -v
```

If these commands return the version of Node.js and npm, it means they are installed. If not, follow the next steps.

2. Install Node.js and npm

To install Node.js (which includes npm), follow these steps:

1. Download and Install Node.js:

- Go to the official [Node.js website](#).
- Download the **LTS (Long Term Support)** version, as it is the most stable version.
- Follow the installation prompts to install Node.js and npm.

2. Verify Installation:

After installation, verify again by running:

```
node -v  
npm -v
```

These commands should now output the versions of Node.js and npm, confirming the installation.

3. Check npm Path Configuration

If you have installed Node.js and npm but are still facing the issue, it could be because the **npm path is not correctly set** in your system environment variables.

To fix this:

- Open **System Properties**.
- Click on **Environment Variables**.
- Under **System Variables**, find the Path variable and ensure the following paths are added:

- For Windows, it should include the following (based on your installation location):

```
C:\Program Files\nodejs\
```

Once added, restart your terminal to make sure the changes take effect.

Steps:

1. Set Up React Application
2. Install React Router for Navigation
3. Create Pages: Registration, Login, Contact, About
4. Set Up Routing to Navigate Between Pages

1. Set Up the React Application

If you haven't created a React app yet, use **Create React App** to set up a basic project.

```
npx create-react-app myapp  
cd myapp  
npm start
```

2. Install React Router

You need **React Router** to implement routing and navigation between different pages. Install it using npm:

```
npm install react-router-dom
```

3. Create Pages

Now, create the following pages:

- **Registration Page:** A form to register a student.
- **Login Page:** A form to login.
- **Contact Page:** A page to display contact information.
- **About Page:** A page to display information about the system.

4. Implement Routing

Update App.js to set up routing and implement the navigation between pages.

Folder Structure:

```

src/
|-- components/
    |-- About.js
    |-- Contact.js
    |-- Login.js
    |-- Registration.js
|-- App.js
|-- App.css
|-- index.js

```

App.js

```

import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
import Registration from './components/Registration';
import Login from './components/Login';
import Contact from './components/Contact';
import About from './components/About';
import './App.css';

const App = () => {
  return (
    <Router>
      <div className="app">
        <header>
          
          <nav>
            <ul>
              <li><Link to="/">Home</Link></li>
                <li><Link to="/about">About</Link></li>
              <li><Link to="/registration">Registration</Link></li>
              <li><Link to="/login">Login</Link></li>
              <li><Link to="/contact">Contact</Link></li>
            </ul>
          </nav>
        </header>
        <main>
          <Routes>
            <Route path="/registration" element={<Registration />} />
            <Route path="/login" element={<Login />} />
            <Route path="/contact" element={<Contact />} />
            <Route path="/about" element={<About />} />
          </Routes>
        </main>
      </div>
    </Router>
  );
}

export default App;

```

```
<Route path="/" element={<h2>Welcome to the Student Management  
System</h2>} />  
      </Routes>  
    </main>  
  </div>  
  </Router>  
</>  
};  
};  
  
export default App;
```

App.css

```
/* App.css */  
body {  
  font-family: Arial, sans-serif;  
  margin: 0;  
  padding: 0;  
  background-color: #f4f4f4;  
}  
  
.app {  
  max-width: 800px;  
  margin: 20px auto;  
  padding: 20px;  
  background-color: white;  
  border-radius: 8px;  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
}  
header nav ul {  
  display: flex;  
  justify-content: space-around;  
  padding: 10px;  
  background-color: #6c3483;  
  margin: 0;  
}  
  
header nav ul li {  
  list-style: none;  
}  
  
header nav ul li a {  
  color: white;  
  text-decoration: none;  
  padding: 10px;  
}
```

```
header nav ul li a:hover {  
    background-color: #f1c40f;  
}  
  
h2 {  
    color: #333;  
}  
  
form input {  
    margin: 5px;  
    padding: 8px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}  
  
form button {  
    padding: 10px 15px;  
    background-color: #4CAF50;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}  
  
form button:hover {  
    background-color: #45a049;  
}
```

index.js

```
import React from 'react';  
import ReactDOM from 'react-dom/client';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
    <React.StrictMode>  
        <App />  
    </React.StrictMode>  
);
```

```
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

About.js

```
import React from 'react';

const About = () => {
  return (
    <div>
      <h2>About Student Management System</h2>
      <p>This system is designed to manage student records, including registration and login functionality.</p>
    </div>
  );
};

export default About;
```

Registration.js

```
import React, { useState } from 'react';

const Registration = () => {
  const [formData, setFormData] = useState({
    name: '',
    email: '',
    password: '',
  });

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({ ...prevData, [name]: value }));
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    console.log('Registered', formData);
  };

  return (
    <div>
      <h2>Student Registration</h2>
```

```
<form onSubmit={handleSubmit}>
  <label>Name: </label>
  <input
    type="text"
    name="name"
    value={ formData.name }
    onChange={ handleChange }
    required
  />
  <br />
  <label>Email: </label>
  <input
    type="email"
    name="email"
    value={ formData.email }
    onChange={ handleChange }
    required
  />
  <br />
  <label>Password: </label>
  <input
    type="password"
    name="password"
    value={ formData.password }
    onChange={ handleChange }
    required
  />
  <br />
  <button type="submit">Register</button>
</form>
</div>
);
};

export default Registration;
```

Login.js

```
import React, { useState } from 'react';

const Login = () => {
  const [credentials, setCredentials] = useState({
    email: '',
    password: '',
  });
}
```

```
const handleChange = (e) => {
  const { name, value } = e.target;
  setCredentials((prev) => ({ ...prev, [name]: value }));
};

const handleSubmit = (e) => {
  e.preventDefault();
  console.log('Logged in', credentials);
};

return (
  <div>
    <h2>Student Login</h2>
    <form onSubmit={handleSubmit}>
      <label>Email:</label>
      <input
        type="email"
        name="email"
        value={credentials.email}
        onChange={handleChange}
        required
      />
      <br />
      <label>Password:</label>
      <input
        type="password"
        name="password"
        value={credentials.password}
        onChange={handleChange}
        required
      />
      <br />
      <button type="submit">Login</button>
    </form>
  </div>
);
};

export default Login;
```

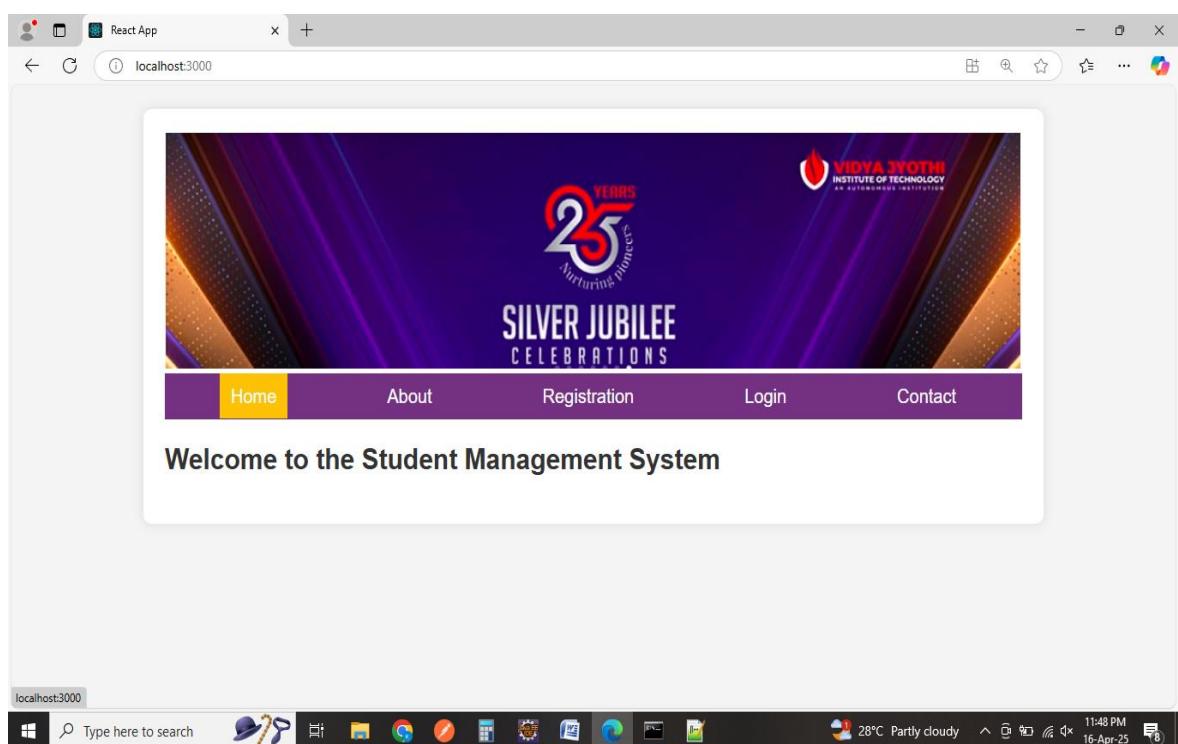
Contact.js

```
import React from 'react';

const Contact = () => {
  return (
    <div>
      <h2>Contact Us</h2>
      <p>Email: support@studentsystem.com</p>
      <p>Phone: 123-456-7890</p>
    </div>
  );
};

export default Contact;
```

Output:



localhost:3000/about

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

25 YEARS
Nurturing Pioneers

SILVER JUBILEE
CELEBRATIONS

Home About Registration Login Contact

About Student Management System

This system is designed to manage student records, including registration and login functionality.



localhost:3000/registration

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

25 YEARS
Nurturing Pioneers

SILVER JUBILEE
CELEBRATIONS

Home About Registration Login Contact

Student Registration

Name:

Email:

Password:

Register

localhost:3000/registration

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

25 YEARS
Nurturing Pioneers

SILVER JUBILEE
CELEBRATIONS

Home About Registration Login Contact

Student Registration

Name:

Email:

Password:

Register

localhost:3000/registration

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

25 YEARS
Nurturing Pioneers

SILVER JUBILEE
CELEBRATIONS

Home About Registration Login Contact

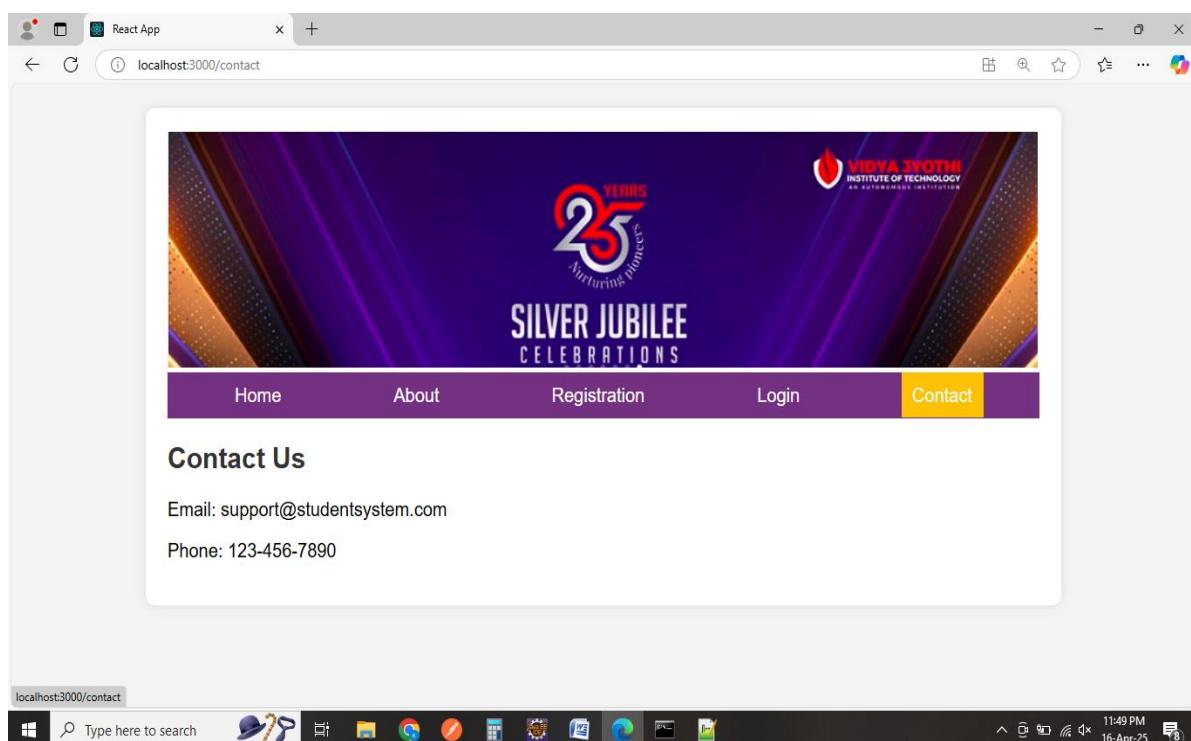
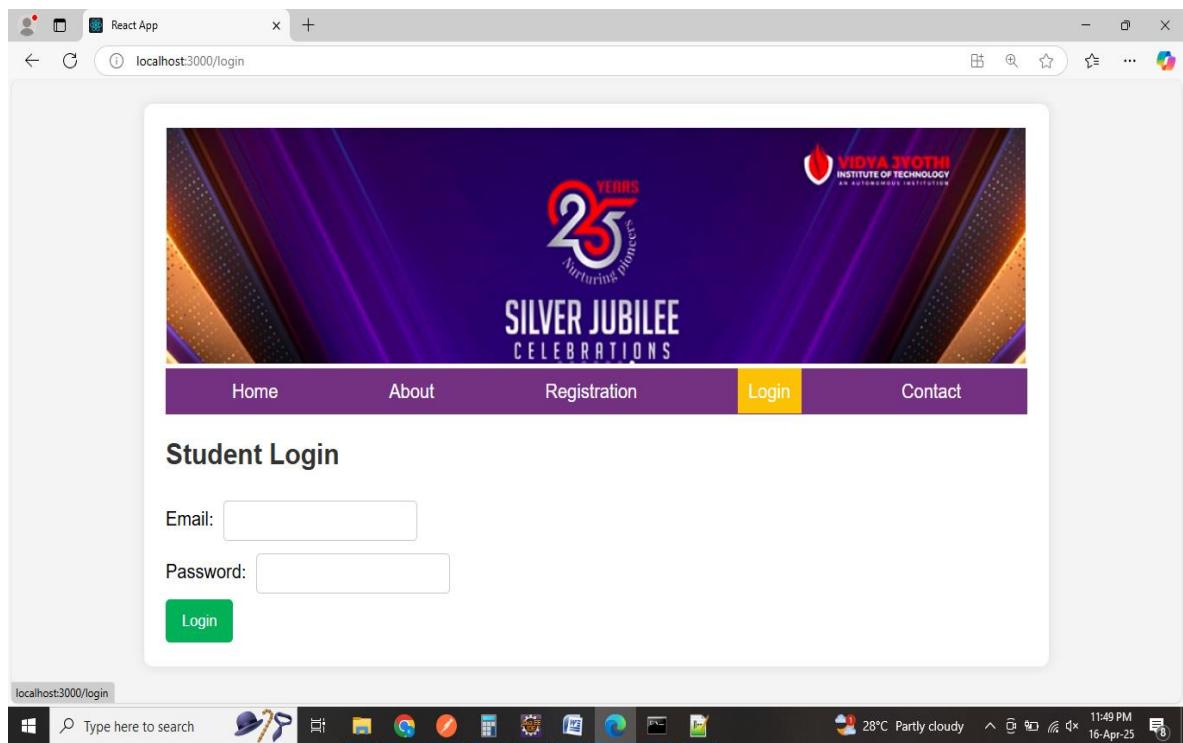
Student Registration

Name:

Email:

Password:

Register



11. Create a **Custom Server** using **HTTP Module** and explore the other modules of Node JS like OS, PATH, EVENT.

server.js

```
// Import required modules
const http = require('http');
const os = require('os');
const path = require('path');
const EventEmitter = require('events');

// Create the HTTP server
const server = http.createServer((req, res) => {
    // Set response headers to return HTML content
    res.setHeader('Content-Type', 'text/html');

    // Trigger the custom event when the server starts
    triggerCustomEvent();

    // Respond with system information and file path information
    res.statusCode = 200;
    res.end(`

        <html>
            <head>

                </head>
                <body>
                    ${getSystemInfo()}
                    ${getFilePathInfo()}
                </body>
            </html>
    `);
});

// Set the server to listen on port 3000
const PORT = 3000;
server.listen(PORT, () => {
    console.log(`Server is running at http://localhost:${PORT}`);
});
```

```
// Create a custom event emitter
const eventEmitter = new EventEmitter();
// Function to trigger a custom event using the 'events' module
function triggerCustomEvent()
{
    eventEmitter.emit('systemEvent'); // Emit a custom event named 'systemEvent'
}

// Event listener for 'systemEvent'
eventEmitter.on('systemEvent', () => {
    console.log("Custom Event Triggered: The system has been started!");
});

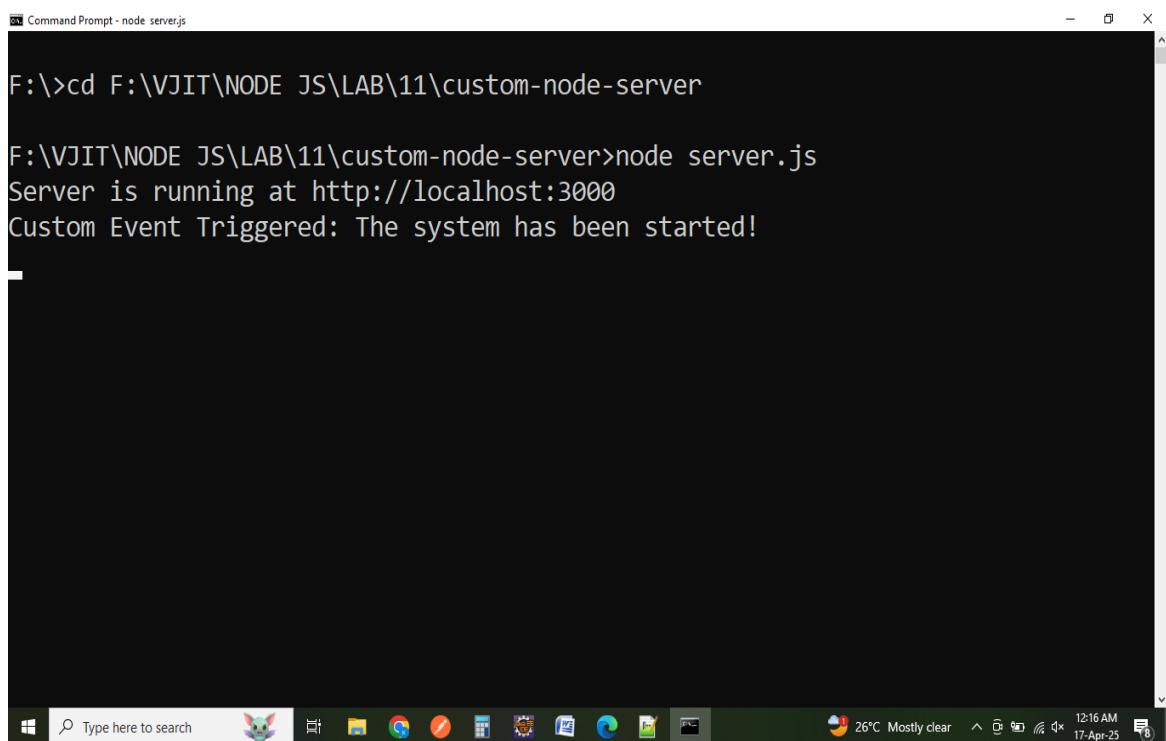
// Function to display system information using the 'os' module
function getSystemInfo()
{
    return `

        <h2>System Information:</h2>
        <ul>
            <li><strong>OS Type:</strong> ${os.type()}</li>
            <li><strong>Platform:</strong> ${os.platform()}</li>
            <li><strong>Architecture:</strong> ${os.arch()}</li>
                <li><strong>Release:</strong> ${os.release()}</li>
            <li><strong>Uptime:</strong> ${os.uptime()} seconds</li>

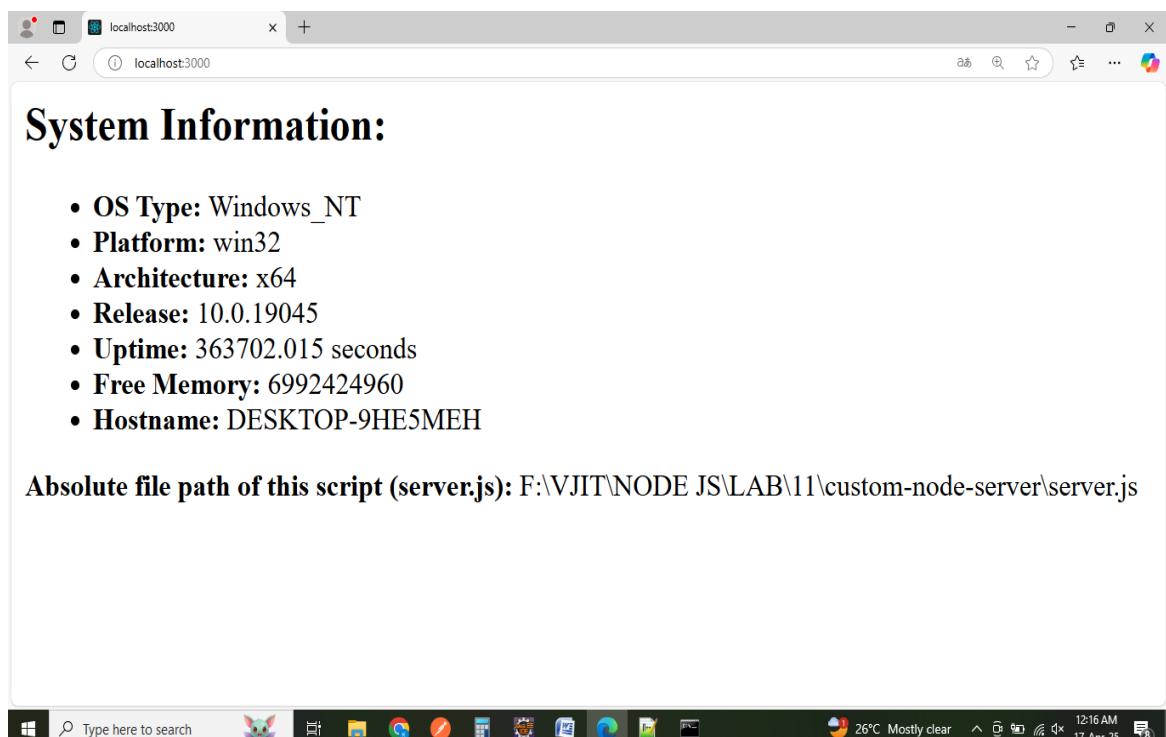
            <li><strong>Free Memory:</strong> ${os.freemem()}</li>
            <li><strong>Hostname:</strong> ${os.hostname()}</li>
        </ul>
    `;
}

// Function to display file path information using the 'path' module
function getFilePathInfo()
{
    // Get the absolute path of the current script
    const filePath = path.join(__dirname, 'server.js');    return `<p><strong>Absolute
file path of this script (server.js):</strong> ${filePath}</p>`;
}
```

Output:



```
F:\>cd F:\VJIT\NODE JS\LAB\11\custom-node-server
F:\VJIT\NODE JS\LAB\11\custom-node-server>node server.js
Server is running at http://localhost:3000
Custom Event Triggered: The system has been started!
```



Absolute file path of this script (server.js): F:\VJIT\NODE JS\LAB\11\custom-node-server\server.js