



**COLLEGE CODE:9111**

**COLLEGE NAME: SRM MADURAI COLLEGE FOR  
ENGINEERING AND TECHNOLOGY**

**DEPARTMENT: B.E COMPUTER SCIENCE AND  
ENGINEERING**

**STUDENT NM-ID: 451186A06133385A11F419FE498D4137**

**A8884C7CC5278F7B7394345D64BD84CB  
1FDBD7B5DCA253284674E6DBD128FDFD  
857E9787939AE7BBA024C9CE6C315408  
DE8BAC54E8107FF98F34AABCF0F38D6C**

**ROLL NO.:911123104018**

**911123104020**

**911123104051**

**911123104053**

**911123104702**

**DATE:**

**Completed the project named as Phase III**

**TECHNOLOGY PROJECT NAME: LOGIN  
AUTHENTICATION SYSTEM**

**SUBMITTED BY, NAME:**

**HOORUL FIRTHOUS.A**

**JAYASAKTHI K R**

**SRI LAKSHMI.S**

**TAAKSHINI DEVI.R**

**RIZWANA BARVEEN M**



---

# Project: Login Authentication System(Phase3)

---

## 1. Project Setup

- Choose tech stack (e.g., **Frontend:** React / HTML-CSS-JS, **Backend:** Node.js/Express, **Database:** MongoDB/MySQL, **Auth:** JWT & bcrypt).
- Initialize repository on **GitHub**.
- Setup development environment (install dependencies, frameworks, database drivers).
- Define folder structure for frontend, backend, and configuration files.

## 2. Core Features Implementation

- **User Registration (Signup):**
  - Input fields (username, email, password).
  - Password hashing before storage.
  - Basic validation (email format, password strength).
- **User Login:**
  - Verify credentials by comparing hashed password.
  - Generate and return **JWT token**.
- **Password Reset (Forgot/Reset):**
  - User requests reset link → sent via email.
  - Secure token-based reset mechanism.
- **Logout:**
  - Invalidate/expire token.
  - Redirect user to login page.

## 3. Data Storage (Local State / Database)

- **Database Tables/Collections:**
  - `users` (id, username, email, hashed\_password, created\_at).
  - `sessions/tokens` (optional, if using refresh tokens).
- **Local State (Frontend):**
  - Store login state (isAuthenticated).
  - Store JWT token (in memory / secure cookies / localStorage with caution).

## 4. Testing Core Features

- **Unit Testing:**
  - Test user registration (duplicate emails, invalid input).
  - Test login (wrong password, correct password).
  - Test JWT verification.
- **Integration Testing:**
  - Test API endpoints with Postman/Insomnia.
  - Verify DB storage & retrieval.
- **UI Testing:**
  - Validate input forms.
  - Ensure error messages display correctly.

## 5. Version Control (GitHub)

- Create GitHub repository for collaboration.
- Use branches (`feature/auth`, `feature/ui`, `bugfix/...`) for modular development.
- Commit frequently with meaningful messages.
- Use **GitHub Issues/Projects** for task tracking.
- Optional: Setup **CI/CD** for auto-deployment.

Minimal code example

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>
</style>
  body {
    margin: 0;
    font-family: Arial, sans-serif;
    background: url('https://images.unsplash.com/photo-1508780709619-79562169bc64?auto=format&fit=crop&w=1350&q=80') no-repeat center center/cover;
    height: 100vh;
    display: flex;
    justify-content: center;
```

```
    align-items: center;
}

.login-container {
    background: rgba(255, 255, 255, 0.9);
    padding: 40px;
    border-radius: 15px;
    box-shadow: 0 8px 20px rgba(0,0,0,0.3);
    width: 350px;
    text-align: center;
}

.login-container img {
    width: 80px;
    margin-bottom: 20px;
}

.login-container h2 {
    margin-bottom: 20px;
    color: #333;
}

.login-container input[type="text"],
.login-container input[type="password"] {
    width: 100%;
    padding: 12px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 8px;
}

.login-container button {
    width: 100%;
    padding: 12px;
    background: #007bff;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 16px;
    cursor: pointer;
}
```

```

    transition: background 0.3s;
}

.login-container button:hover {
    background: #0056b3;
}

.login-container p {
    margin-top: 15px;
    font-size: 14px;
}

.login-container a {
    color: #007bff;
    text-decoration: none;
}
.login-container a:hover {
    text-decoration: underline;
}
</style>
</head>
<body>
    <div class="login-container">
        
        <h2>Login</h2>
        <form action="/login" method="post">
            <input type="text" name="username" placeholder="Enter Username"
required>
            <input type="password" name="password" placeholder="Enter Password"
required>
            <button type="submit">Login</button>
        </form>
        <p>Don't have an account? <a href="/register">Register</a></p>
    </div>
</body>
</html>

```

