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 IV

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(Phase4)

1. Additional Features

- **Forgot Password Flow**: Allow users to reset password via email verification.
- **Remember Me / Stay Logged In**: Use secure cookies or refresh tokens.
- **Profile Page**: After login, users see their profile details.
- **Logout**: Clear session/JWT token securely.
- **Role-Based Access Control**: Different dashboards for Admin / User.

2. UI/UX Improvements

- Add **register/login forms with proper HTML templates** (already started).
- **Responsive design** for mobile, tablet, desktop.
- Add animations & better **form validation (e.g., password strength meter)**.
- Use **toast notifications** for success/failure instead of plain text.
- Add **custom themes/images/icons** to make it professional.

3. API Enhancements

- Add **/forgot-password** and **/reset-password** endpoints.
- Secure endpoints with **JWT authentication** instead of just text sessions.
- Use proper **HTTP status codes** (200, 400, 401, 500).
- Add **rate limiting** to prevent brute-force attacks.
- API documentation with **Swagger or Postman Collection**.

4. Performance & Security Checks

- **Database indexing** on `username` for faster lookups.
- Ensure **passwords are always hashed** with strong algorithms (bcrypt/argon2).
- Enable **HTTPS** (SSL/TLS).
- Prevent **SQL Injection** by using parameterized queries (already applied).

- Add **Content Security Policy (CSP)** headers.
- Optimize static assets (minify CSS/JS/images).

5. Testing of Enhancements

- **Unit Testing** for register/login/logout API endpoints.
- **Integration Testing** for user flows (signup → login → logout).
- **Penetration Testing** for SQL injection, XSS, CSRF.
- **Cross-browser testing** (Chrome, Firefox, Safari, Edge).
- **Load testing** with tools like Apache JMeter or Locust.

6. Deployment

- **Frontend Deployment**:
  - Netlify / Vercel for static HTML login/register pages.
- **Backend Deployment**:
  - Use **Heroku / Render / Railway / AWS / GCP** for Flask API.
  - Use **SQLite for dev**, **PostgreSQL/MySQL** for production.
- **Domain & SSL**:
  - Use a custom domain (e.g., `authsystem.xyz`).
  - Enable **free SSL** via Let's Encrypt.
- **CI/CD Setup**:
  - GitHub → GitHub Actions → Auto deploy to chosen cloud platform.

Code example:

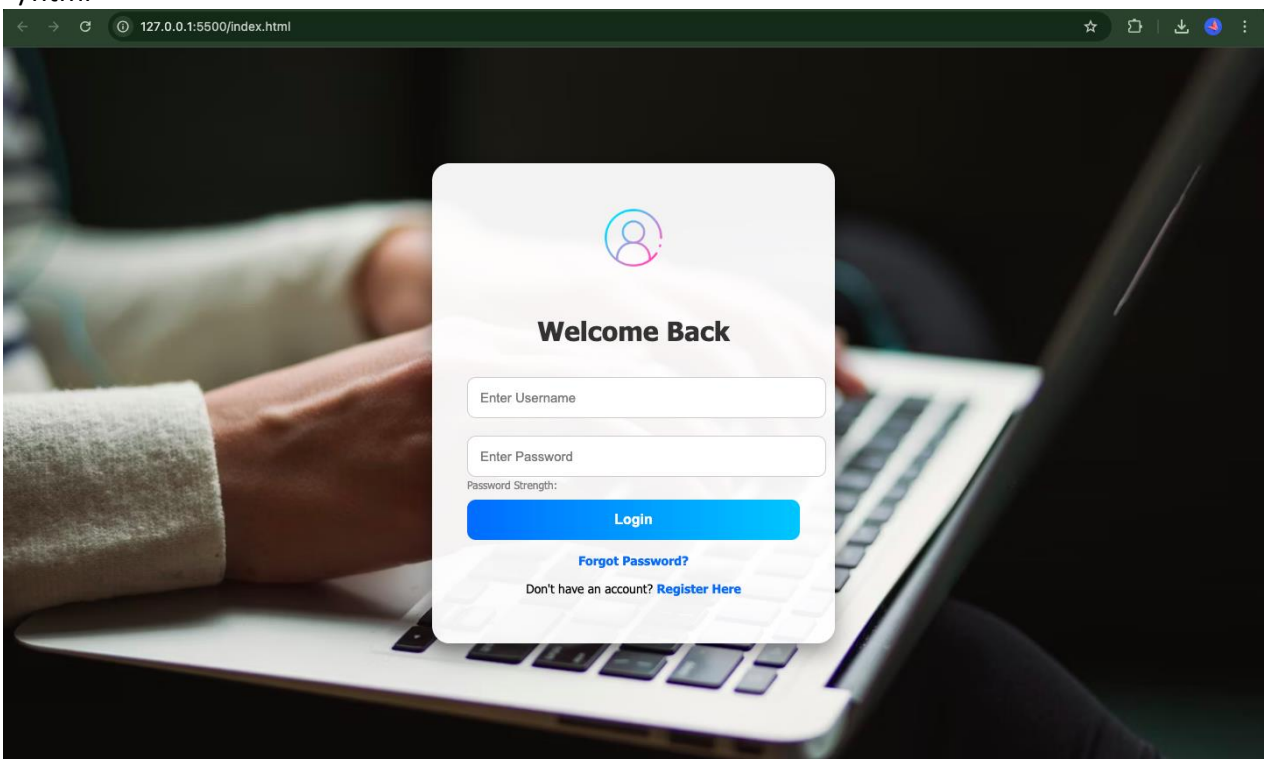
```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Secure Login Page</title>
 <style>
  body {
   margin: 0;
   font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
   background: url('https://images.unsplash.com/photo-1508780709619-
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, 0.95);
  padding: 40px;
  border-radius: 20px;
  box-shadow: 0 10px 25px rgba(0,0,0,0.4);
  width: 380px;
  text-align: center;
  animation: fadeIn 1s ease-in-out;
}

@keyframes fadeIn {
  from { opacity: 0; transform: translateY(-20px); }
  to { opacity: 1; transform: translateY(0); }
}

.login-container img {
  width: 90px;
  margin-bottom: 15px;
}

.login-container h2 {
  margin-bottom: 25px;
  color: #333;
  font-weight: bold;
}

.login-container input[type="text"],
.login-container input[type="password"] {
  width: 100%;
  padding: 14px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 10px;
  font-size: 15px;
  transition: border 0.3s;
}

.login-container input:focus {
  border: 1px solid #007bff;
  outline: none;
}
```

```
.login-container button {
  width: 100%;
  padding: 14px;
  background: linear-gradient(90deg, #007bff, #00c6ff);
  color: white;
  border: none;
  border-radius: 10px;
  font-size: 16px;
  font-weight: bold;
  cursor: pointer;
  transition: transform 0.2s, background 0.3s;
}

.login-container button:hover {
  transform: scale(1.05);
  background: linear-gradient(90deg, #0056b3, #0096c7);
}

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

.login-container a {
  color: #007bff;
  font-weight: bold;
  text-decoration: none;
}
.login-container a:hover {
  text-decoration: underline;
}

.login-container .extra-links {
  margin-top: 15px;
  font-size: 13px;
}

.strength {
  text-align: left;
  font-size: 12px;
  margin-top: -8px;
  margin-bottom: 10px;
  color: gray;
```

```
  }
 </style>
</head>
<body>
 <div class="login-container">
  <img src="https://cdn-icons-png.flaticon.com/512/5087/5087579.png" alt="Login Icon">
  <h2>Welcome Back</h2>

  <!-- Login Form -->
  <form action="/login" method="post">
   <input type="text" name="username" placeholder="Enter Username" required>
   <input type="password" id="password" name="password" placeholder="Enter Password"
required>
   <div class="strength" id="strengthMessage">Password Strength: </div>
   <button type="submit">Login</button>
  </form>

  <div class="extra-links">
   <p><a href="/forgot-password">Forgot Password?</a></p>
  </div>

  <p>Don't have an account? <a href="/register">Register Here</a></p>
 </div>

 <!-- Password Strength Script -->
 <script>
  const passwordInput = document.getElementById('password');
  const strengthMessage = document.getElementById('strengthMessage');

  passwordInput.addEventListener('input', () => {
   const value = passwordInput.value;
   let strength = "Weak";
   let color = "red";

   if (value.length > 6 && /[A-Z]/.test(value) && /\d/.test(value) && /[^A-Za-z0-9]/.test(value)) {
    strength = "Strong";
    color = "green";
   } else if (value.length > 4 && (/[A-Z]/.test(value) || /\d/.test(value))) {
    strength = "Medium";
    color = "orange";
   }
   strengthMessage.textContent = "Password Strength: " + strength;
   strengthMessage.style.color = color;
  });
```

```
 </script>
</body>
</html>
```



Git hub link: https://github.com/hoorulfirthous/Nan-mudhalvan