Software Requirements Specification (SRS)

For

User Registration and Login System

(A PHP–MySQL Based Authentication Module)

Prepared by: *Priyamay Mondal, Paromita Saha*
Course: BCA – 5th Semester
Subject: PHP with MySQL Lab (BCAC591)

---

1. Core Features & System Work Flow

1.1 User Features (Primary flows)

1. Registration (Sign-Up)

- Users can create an account by providing a unique username or email, a secure password, and optional profile details such as their name.
- Both client-side and server-side validation ensure all required fields are completed and password strength meets defined criteria.
- Passwords are securely hashed using PHP's password_hash() function before being stored in the database.
- Upon successful registration, users are redirected to the login page or optionally logged in automatically.

2. Login (Authentication)

- Users log in using their registered username or email along with their password.
- Passwords are verified using PHP's password_verify() function to ensure secure authentication.
- A secure session is initiated upon successful login, and users are redirected to their protected profile page.
- Failed login attempts trigger a generic error message to prevent disclosure of which credential was incorrect.

3. Profile Page (Protected Access)

- Displays personalized user information including username, registration date, and an editable display name.
- Accessible only when a valid session is active, ensuring authenticated access.
- Includes a logout option that securely terminates the session.

4. Change Password

- Authenticated users can update their password by providing the current password and a new one.
- The current password is verified before allowing the update; the new password is securely hashed before storage.

5. Logout

- Securely ends the user session using PHP's session management functions.
- Implements complete session termination using *session_unset(), session_destroy(),* and cookie invalidation to prevent unauthorized access post-logout.

1.2 Optional / Administrative Features

1. **Admin Dashboard**
   - Provides a secure interface for administrators to view and manage registered users. Access is restricted via admin credentials.

2. **Account Deactivation/Deletion**
   - Allows users or administrators to deactivate or permanently delete user accounts, ensuring flexibility and control over account lifecycle.

3. **Email Verification *(Planned Enhancement)***
   - Introduces a verification step during registration to confirm the user's email address, enhancing account authenticity and security.

4. **Rate Limiting & Account Lockout *(Security Enhancement)***
   - Implements protection against brute-force attacks by limiting login attempts and temporarily locking accounts after repeated failures.

1.3. Basic Security Features

1. **Password Hashing:**
   - Store passwords securely using PHP's password_hash() with the default bcrypt algorithm.
   - Authenticate users with password_verify() to ensure safe password comparison.

2. **Input Validation & Output Escaping:**
   - Use MySQLi prepared statements for all database interactions.
   - Avoid direct string concatenation in SQL queries to prevent injection vulnerabilities.

3. **Error Handling & Messaging:**
    - Regenerate session IDs on login using session_regenerate_id(true) to prevent session fixation.
    - Configure secure cookie parameters (httponly, secure, samesite) to protect session data.
    - Terminate sessions properly during logout using session_unset(), session_destroy(), and cookie invalidation.
    - These measures ensure sessions are securely managed and fully invalidated to prevent unauthorized reuse.

4. **Rate Limiting / Brute-Force Mitigation (recommended):**
    - Apply server-side validation for all user inputs, checking for required fields, length limits, and allowed characters.
    - Escape user-generated content using htmlspecialchars() when rendering output to prevent cross-site scripting (XSS).

5. **HTTPS Requirement (deployment):**
    - Display generic error messages for login failures to avoid revealing whether the username or password was incorrect.
    - Log detailed server-side errors to a secure location for debugging purposes, without exposing them to users.

6. **Rate Limiting & Brute-Force Protection** *(Recommended)*
    - Monitor failed login attempts and implement throttling or temporary account lockouts after configurable thresholds.
    - Optionally integrate CAPTCHA challenges after repeated login failures to deter automated attacks.

6. **HTTPS Enforcement** *(Deployment Requirement)*
    - Use HTTPS in production environments to encrypt data in transit and protect sensitive credentials from interception.
