#Authentication project

Functional Requirements

* User Registration
* **Form Submission:** Users should be able to fill out a registration form with fields like username, email, and password.
* **Validation:** Validate user inputs for proper format and check for existing accounts with the same email/username.
* **Password Encryption:** Use bcrypt or another library to hash passwords before storing them in the database.
* **Email Verification:** Optionally, send a verification email to confirm the user’s email address.
* User Login
* **Form Submission:** Users should be able to log in with their email/username and password.
* **Authentication:** Compare the hashed password stored in the database with the user’s input.
* **Session Management:** Use JWT (JSON Web Token) or another method to maintain user sessions securely.
* Password Management
  + **Reset Password:** Provide a mechanism for users to reset their passwords via email verification.
  + **Change Password:** Allow authenticated users to change their password from their profile.
* Profile Management
  + **View Profile:** Authenticated users should be able to view their profile information.
  + **Edit Profile:** Allow users to update their profile information like name, email, and other optional details.
* Access Control
* **Role-Based Access Control:** Implement roles (e.g., user, admin) and restrict access to certain features based on roles.
* **Protected Routes:** Ensure that certain routes are accessible only to authenticated users.
* Security
* **Rate Limiting:** Prevent brute-force attacks by limiting the number of login attempts.
* **Input Validation:** Protect against common vulnerabilities like SQL injection and XSS by validating and sanitizing user inputs.
* **HTTPS:** Use HTTPS to secure data transmission between the client and the server.
* User Feedback
* **Notifications:** Inform users about successful registration, login, and any errors during these processes.
* Logging and Monitoring
* **Activity Logs:** Keep logs of user activities, such as login attempts, password changes, and other critical actions.
* **Error Handling:** Provide meaningful error messages and handle errors gracefully.