Security Measures & Testing

Authentication, Authorization, and Data Protection Mechanisms:

Authentication:

- Users register by providing an email, username, and password.
- Firebase Authentication is used to manage user identities, including registration, login, and password reset.
- Email verification is implemented to confirm the user's identity before allowing full account functionality.

Authorization:

- Only registered users can log in and receive an authentication token (idToken).
- The application uses Firebase Authentication to manage access to user-related data and actions.
- Email verification is enforced before the user's account is fully activated.

Data Protection:

- User credentials are encrypted using AES before being transmitted for verification.
- Firebase Authentication stores passwords securely using industry-standard hashing mechanisms.
- User data is stored in Firebase Realtime Database under unique user IDs to prevent unauthorized access.

Security Best Practices Followed:

The system follows various security best practices to protect user data and prevent vulnerabilities:

- **HTTPS Enforcement**: All API requests and Firebase Authentication transactions occur over HTTPS, preventing data interception.
- **Password Hashing**: Firebase Authentication securely hashes and stores user passwords, reducing the risk of credential exposure.
- **Input Validation**: The system ensures that all required fields (email, username, password) are provided. Email validation is handled by Firebase, ensuring users enter valid email addresses.

- **Email Verification**: A verification email is sent to new users, preventing unauthorized account creation and ensuring only verified users can complete registration.
- **Encryption of Sensitive Data**: AES encryption is used when transmitting sensitive user data via email verification links to protect against interception.
- **Error Handling and Logging**: Proper error handling ensures that detailed error messages are not exposed to attackers. Server-side logging helps in debugging security-related incidents.
- **Rate Limiting and Throttling**: Firebase Authentication provides built-in protections against brute-force attacks, preventing repeated login attempts from malicious actors.

Results of Basic Security Tests:

- **SQL Injection Prevention**: Since the system does not use SQL queries, the risk of SQL injection is minimal. Firebase handles data retrieval securely through structured queries.
- **Cross-Site Scripting (XSS) Prevention**: The frontend does not directly render user-generated content, reducing the risk of stored XSS attacks. The application uses React's built-in escaping to prevent direct script execution. Messages displayed via setMessage(response.message) and setError(response.error) originate from server responses, which are handled without directly inserting raw user input into the DOM thus minimizing XSS risks.
- **Broken Authentication Testing**: Authentication tests confirmed that only registered and verified users could log in. Unauthorized access attempts without a valid authentication token were blocked.
- **Password Reset Security**: The password reset functionality was tested and requires a valid reset code (oobCode), which is received through email, before allowing password changes, preventing unauthorized password resets.