***EXPENSE TRACKER PROJECT***

**Release Notes & Feature Descriptions**

**Version : 2025.1.1**

**Features :**

* Add work role in signup window. (Monu-FE&BE) **(DONE)**
* Merge add expense and table window. (Monu-FE) **(WORKING)**
* Login Page Sign-In Attempts feature (Arnav-BE & Monu-BE) \* Feature Description Below. **(WORKING)**
* Password decrypted form save in database. (Arnav-BE)
* Subscription model feature implementation. (To be decided- **NEXT Release**)

**Bugs.**

* Back button to login page. (Arnav) **(DONE)**
* Refresh only same page don’t go back to login page. (Monu)
* If we enter wrong creds and then write correct creds and enter in the application and again enter wrong creds the login attempts should again update to 3 but its coming 2 login attempts left. (Monu)
* Password and Confirm password copy prevention. (Arnav)
* Email already Taken Bug. (Arnav) **(DONE)**

**Login Page Sign-In Attempts Feature**

**Feature Overview:**

* Implement a feature to limit user login attempts for enhanced security.
* Users will have a maximum of **3 login attempts** to enter correct credentials.
* After **3 consecutive failed attempts**, the user's account will be temporarily **blocked for 30 minutes**.
* Clear error messages and validation feedback will be shown to users.

**Front-End (FE) Requirements - Assigned to Monu:**

1. **Track Login Attempts:**
   * Maintain a counter for failed login attempts locally (if needed) until synced with the backend.
   * Display appropriate error messages after each failed attempt.
2. **Display Account Block Notification:**
   * Inform the user when their account is blocked after the third failed attempt.
   * Show a countdown timer indicating the remaining block time (optional but improves UX).
3. **Validation Handling:**
   * Ensure proper validation for incorrect credentials.
   * Display the following messages dynamically:
     + After 1st and 2nd failed attempts: "Incorrect credentials. X attempts remaining."
     + After 3rd failed attempt: "Your account is temporarily blocked for 30 minutes."
4. **Integration with Back-End APIs:**
   * Sync the front-end with back-end endpoints to:
     + Retrieve current login attempt count and block status.
     + Reset login attempts on successful login.
     + Display the block time if the account is blocked.

**Back-End (BE) Requirements - Assigned to Arnav:**

1. **Database Updates:**
   * Add fields in the database table to track login attempts and block status.
2. **Logic Implementation:**
   * **On Login Attempt:**
     + Validate user credentials.
     + If incorrect:
       - Increment the attempt count.
       - After 3 failed attempts, set the block\_end\_time to the current time + 30 minutes.
     + If correct:
       - Reset the attempt\_count and clear block\_end\_time.
   * **On Block Check:**
     + If block\_end\_time is not NULL and greater than the current time, deny login and provide the block time.
3. **API Endpoints:**
   * **POST /login**: Handle login requests, validate credentials, and manage login attempt logic.
   * **GET /login-status**: Return the current login attempt count and block status for a user.
4. **Error Handling:**
   * Ensure appropriate error messages and HTTP status codes are returned.
   * Handle race conditions for concurrent login attempts.

**Dev To-Do List:**

**Front-End (Monu):**

1. Design the front-end logic to track and display login attempts.
2. Create dynamic validation messages for each login attempt.
3. Implement API integration for retrieving and updating login attempt data.
4. Add a timer as well in the validation box.

**Back-End (Arnav):**

1. Update the database schema to include attempt\_count and block\_end\_time.
2. Implement the login attempt and blocking logic in the authentication service.
3. Develop and test the required API endpoints.
4. Ensure proper unit and integration tests for the feature.