Individual Report – Task Management System

# Description of Subsystem

The components I was tasked with in this project focused on the User subsystem and part of the frontend development. These include the user registration and login pages, the user profile interface, and access control through authentication. On the backend, I was responsible for setting up user management using Java with Spring Boot, while on the frontend, I used React.js to build the interfaces that interact with the backend via API.  
  
For backend development, I made use of JWT (JSON Web Tokens) for secure authentication, and implemented validation for user registration inputs such as name, email, and password strength. I also contributed to the design of role-based access control (Admin vs Team Member) to restrict what users can see or do based on their assigned role.  
  
On the frontend side, I built the login and registration forms, styled them using CSS and React components, and linked them to the backend APIs. I also helped in developing the dashboard layout that displays task summaries and notifications relevant to each user after login.

# Business Rules for the User Pages

* ➢ A user must be registered and logged in to access any protected pages (e.g., dashboard, task lists).
* ➢ Login will only be successful if the email and password match an existing user in the database.
* ➢ A valid email format must be entered during registration.
* ➢ Passwords must be a minimum of 8 characters and must be encrypted before being stored.
* ➢ Users are assigned roles (Admin or Team Member) that determine their access levels.
* ➢ Only Admins can view and manage all users in the system.
* ➢ Users can only edit their own profiles unless they are Admins.
* ➢ Error messages must be shown on the frontend to guide the user when incorrect or invalid information is entered.
* ➢ If a user is inactive or provides invalid login details three times, their session is blocked temporarily (still under development).

# What Still Needs to Be Done

I have completed the main functionality for user registration, login, and role management. However, a few items still need to be finalized:

* ➢ I still need to improve profile update validation, especially for names (must not contain numbers or special characters).
* ➢ I need to finish the user role toggling UI for the Admin interface (changing a user's role from Team Member to Admin).
* ➢ I’m also working on implementing the "forgot password" feature on the login page.
* ➢ On the frontend, user feedback messages (like “Login successful” or “Invalid credentials”) still need refinement to align with the backend response codes.

# Data Used

The user-related data collected includes:  
- Full Name  
- Email Address  
- Password (stored securely)  
- Role (Admin or Team Member)  
- Profile Picture (optional, still in progress)  
  
This data is stored in a MySQL database, and access to it is managed securely using encrypted tokens and Spring Security. On the frontend, sensitive data is protected using secure local storage and logout mechanisms.  
  
The user data fits into the bigger system by acting as the core access point for all other components—only logged-in users can create, view, assign, or comment on tasks. Admins use this data to manage system access and oversee task activities by different users.

# Communication Challenges

One of the challenges I faced this term was that group communication was not consistent. Some group members were not responding on time and did not always complete their tasks as planned. This delayed some integration steps and made it difficult to test frontend-backend connections fully. Despite this, I focused on completing my own sections and kept progressing independently where possible.