CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-G4
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App – User Registration & Authentication System

Prepared By: Elijah Thomas N. Rellon

Date of Submission: February 1, 2026

Version: 1.0

Table of Contents

1. Introduction 3
   1.1. Purpose 3
   1.2. Scope 3
   1.3. Definitions, Acronyms, and Abbreviations 3
2. Overall Description 3
   2.1. System Perspective 3
   2.2. User Classes and Characteristics 4
   2.3. Operating Environment 4
   2.4. Assumptions and Dependencies 4
3. System Features and Functional Requirements 4
   3.1. Feature 1: 4
   3.2. Feature 2: 5
4. Non-Functional Requirements 5
5. System Models (Diagrams) 5
   5.1. ERD 6
   5.2. Use Case Diagram 7
   5.3. Activity Diagram 8
   5.4. Class Diagram 9
   5.5. Sequence Diagram 10
6. Appendices 10

1. Introduction

1.1. Purpose

The purpose of this document is to describe the functional and non-functional requirements of the Mini App – User Registration & Authentication System. This document is intended for students, instructors, and developers who will design, analyze, and later implement the system using ReactJS, Spring Boot, and Firebase.

1.2. Scope

The system provides basic user account management features including account registration, login, logout, and access to a protected user dashboard. The system ensures that only authenticated users can access protected pages. This document focuses solely on system documentation and diagrams, not on code implementation.

1.3. Definitions, Acronyms, and Abbreviations

- **FRS** – Functional Requirements Specification
- **UI** – User Interface
- **API** – Application Programming Interface
- **ERD** – Entity Relationship Diagram
- **JWT** – JSON Web Token
- **CRUD** – Create, Read, Update, Delete

2. Overall Description

2.1. System Perspective

The system is a client-server application composed of a React-based frontend, a Spring Boot backend API, and a Firebase database. The frontend handles user interaction, the backend manages authentication logic, and Firebase stores user credentials and profile data.

2.2. User Classes and Characteristics

Guest User

- Has no account or is not logged in
- Can register an account
- Can log in

Authenticated User

- Successfully logged in
- Can view personal dashboard/profile
- Can log out

2.3. Operating Environment

- **Frontend:** ReactJS running on modern web browsers
- **Backend:** Spring Boot REST API
- **Database:** Firebase (Authentication & Firestore/Realtime Database)
- **Tools:** draw.io / diagrams.net for diagrams

2.4. Assumptions and Dependencies

- Users have internet access
- Firebase services are available and properly configured
- Backend API is reachable by the frontend
- Passwords are securely hashed before storage

3. System Features and Functional Requirements

3.1. Feature 1:

Description: Allows a guest user to create a new account by providing valid personal and login information.
Functional Requirements:

- The system shall allow a guest user to register using email and password
- The system shall validate required input fields
- The system shall store user information in the database upon successful registration

3.2. Feature 2:

Description: Allows registered users to log in, access protected pages, and securely log out of the system.

Functional Requirements:

- The system shall authenticate users using valid credentials
- The system shall generate an authentication token upon successful login
- The system shall restrict access to protected pages for unauthenticated users -The system shall invalidate the session or token upon logout

4. Non-Functional Requirements

- **Security:** Passwords must be encrypted and tokens securely handled
- **Performance:** Login and registration responses should occur within acceptable time limits
- **Usability:** The UI must be simple and user-friendly
- **Reliability:** The system should handle invalid inputs and authentication errors gracefully

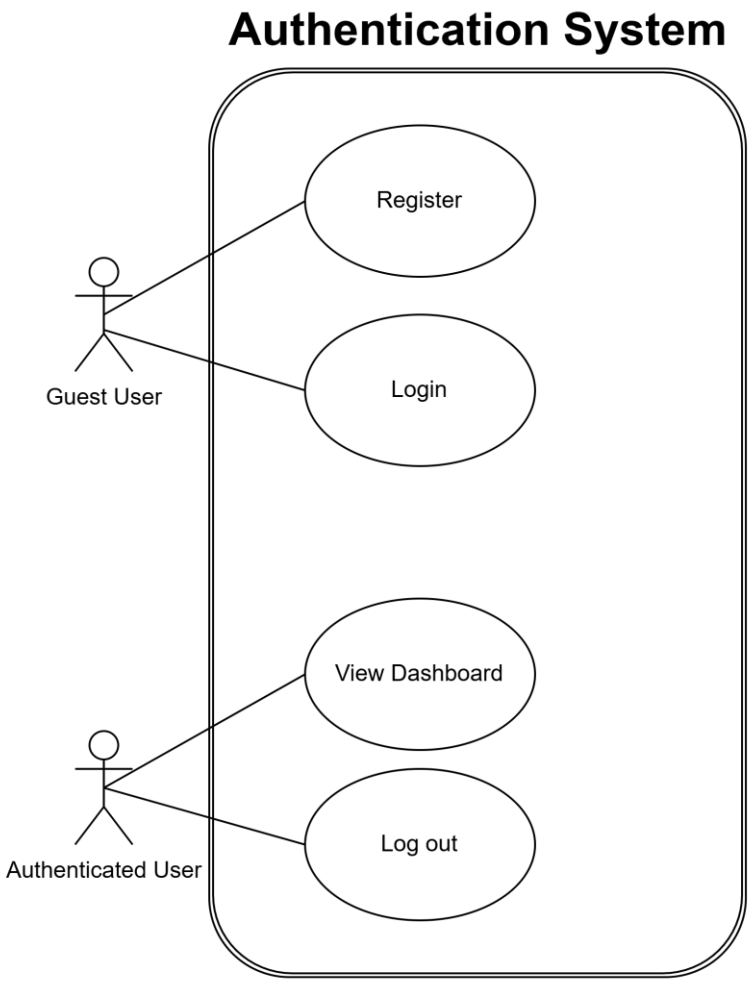
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

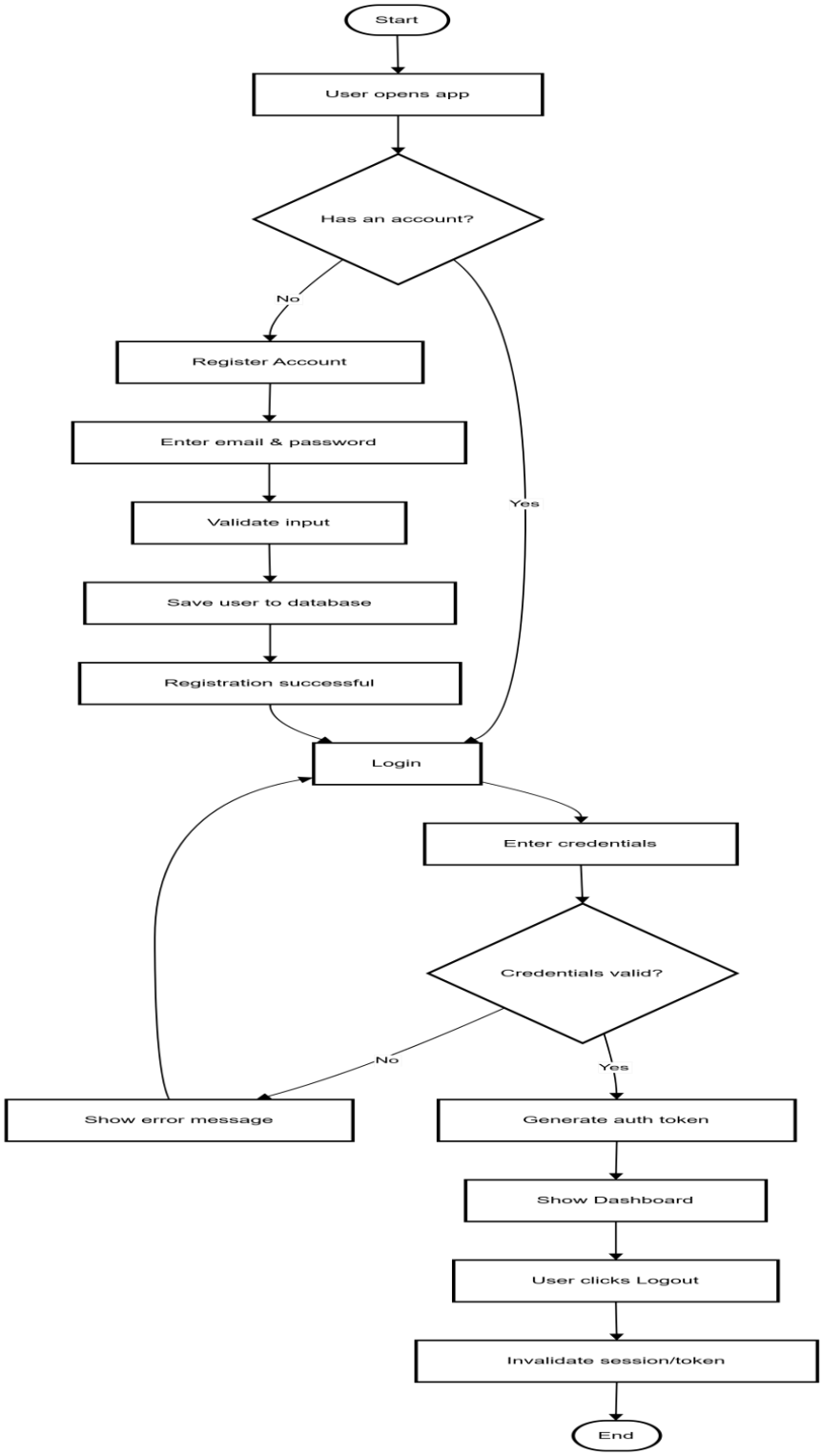
5.1. ERD

User	
Long	**id**
String	fullName
String	email
String	passwordHash
String	createdAt

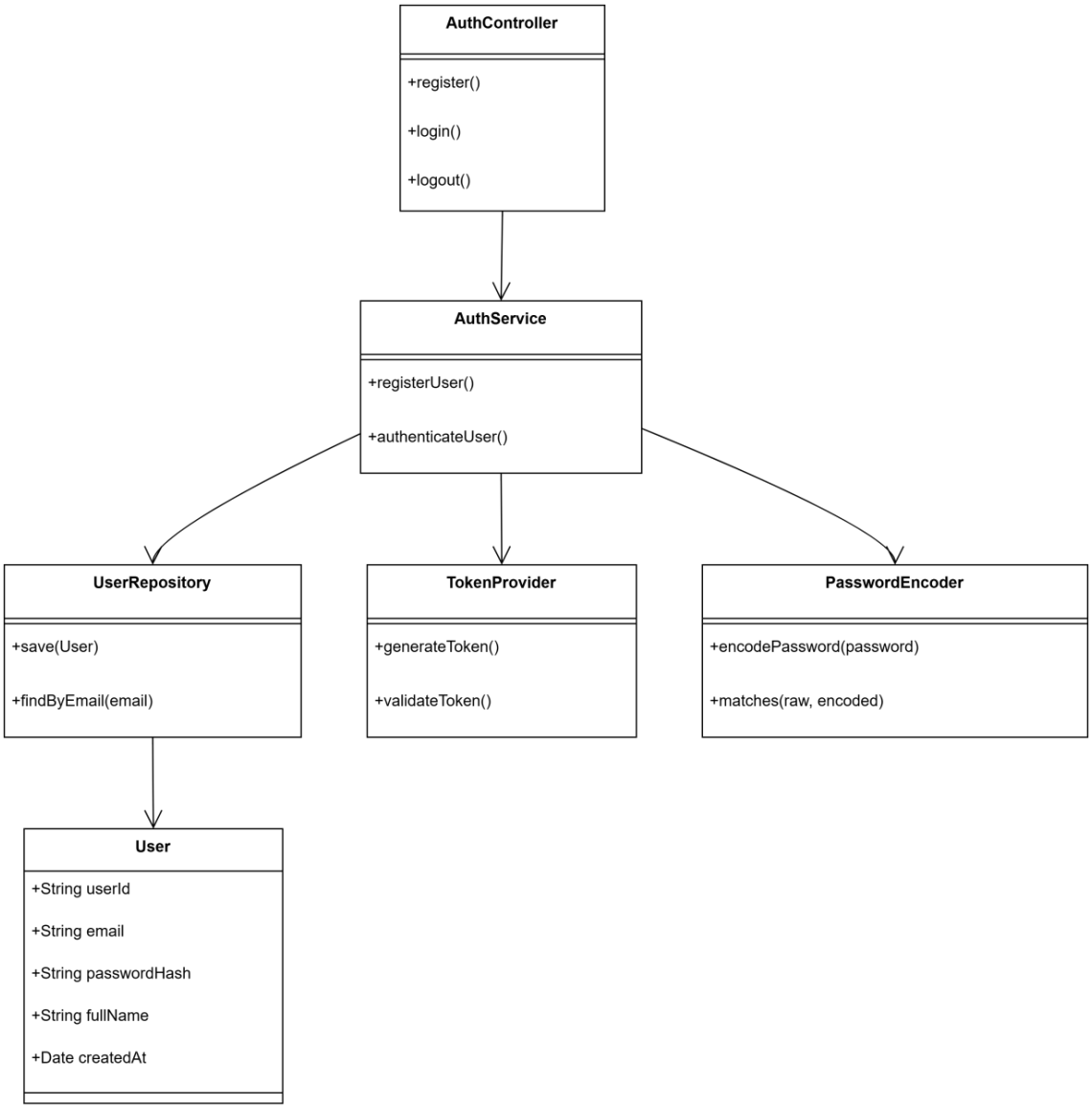
5.2. Use Case Diagram



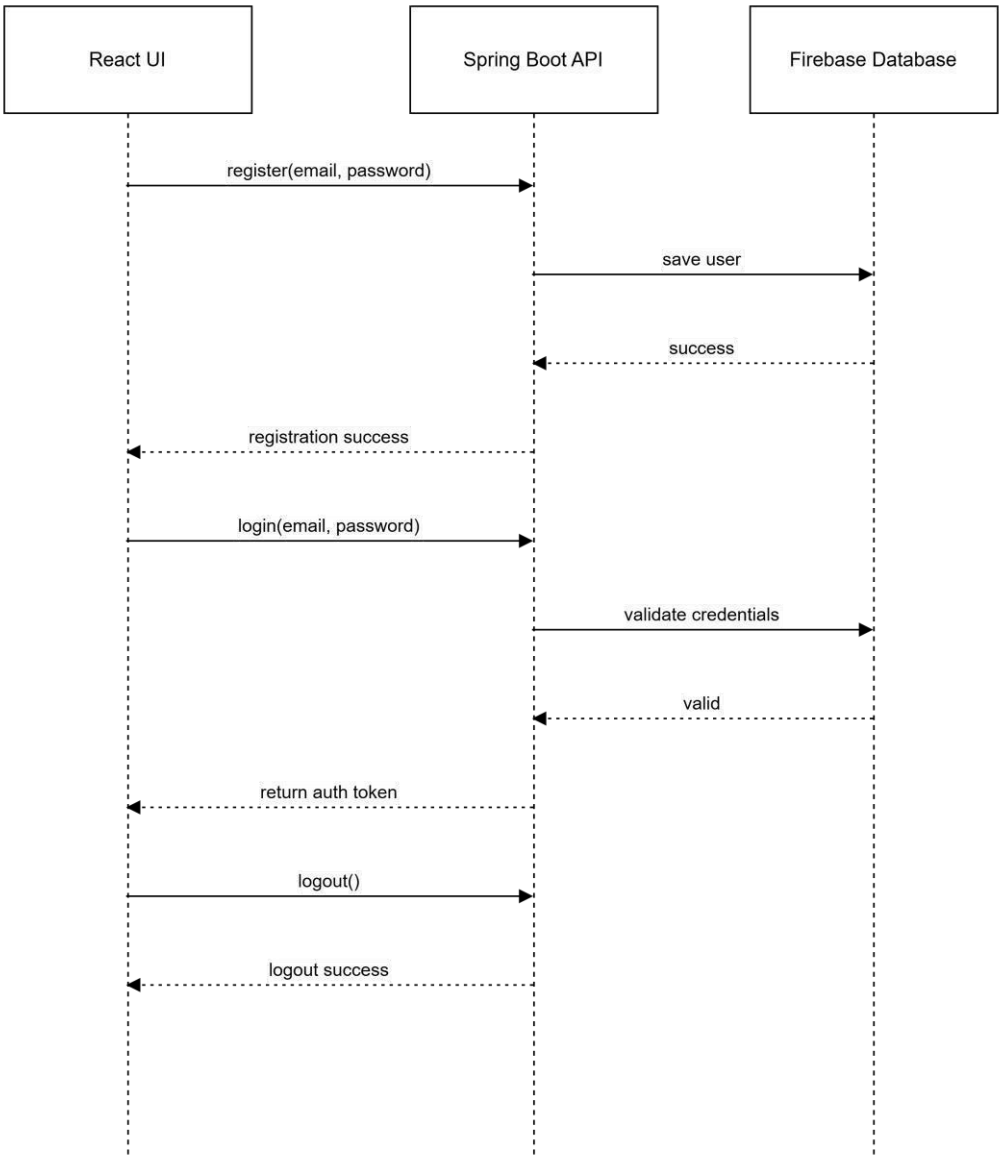
5.3. Activity Dagram



5.4. Class Diagram



5.5. Sequence Diagram

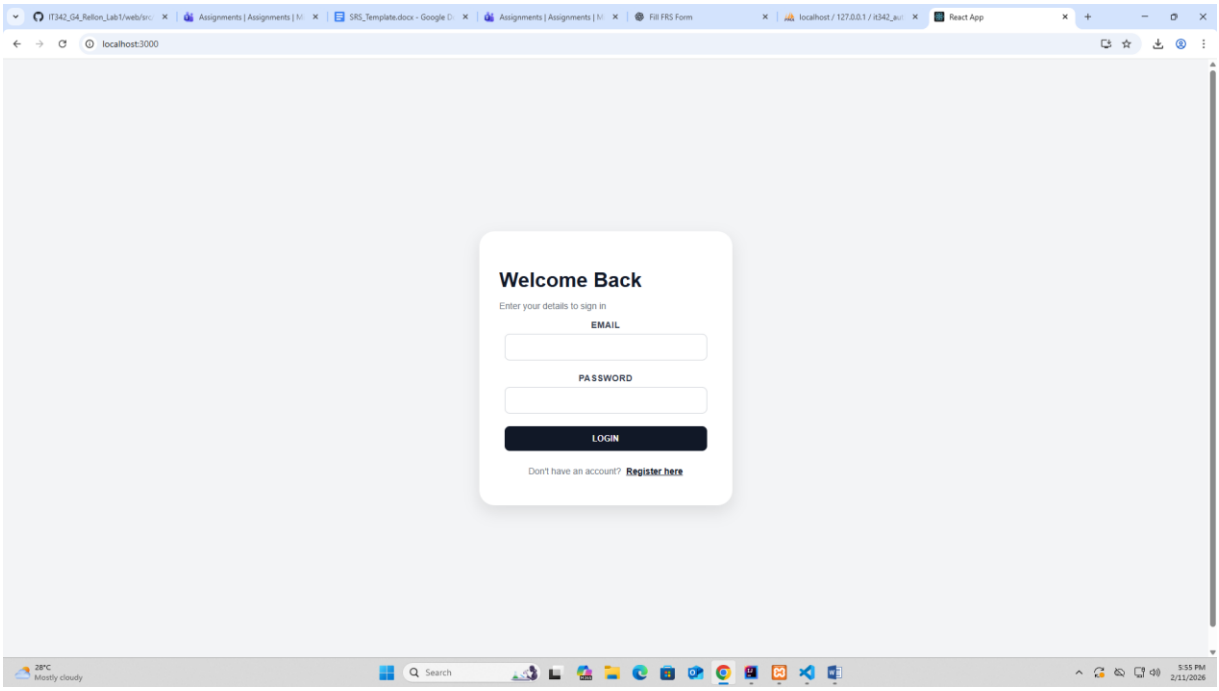


6. Appendices

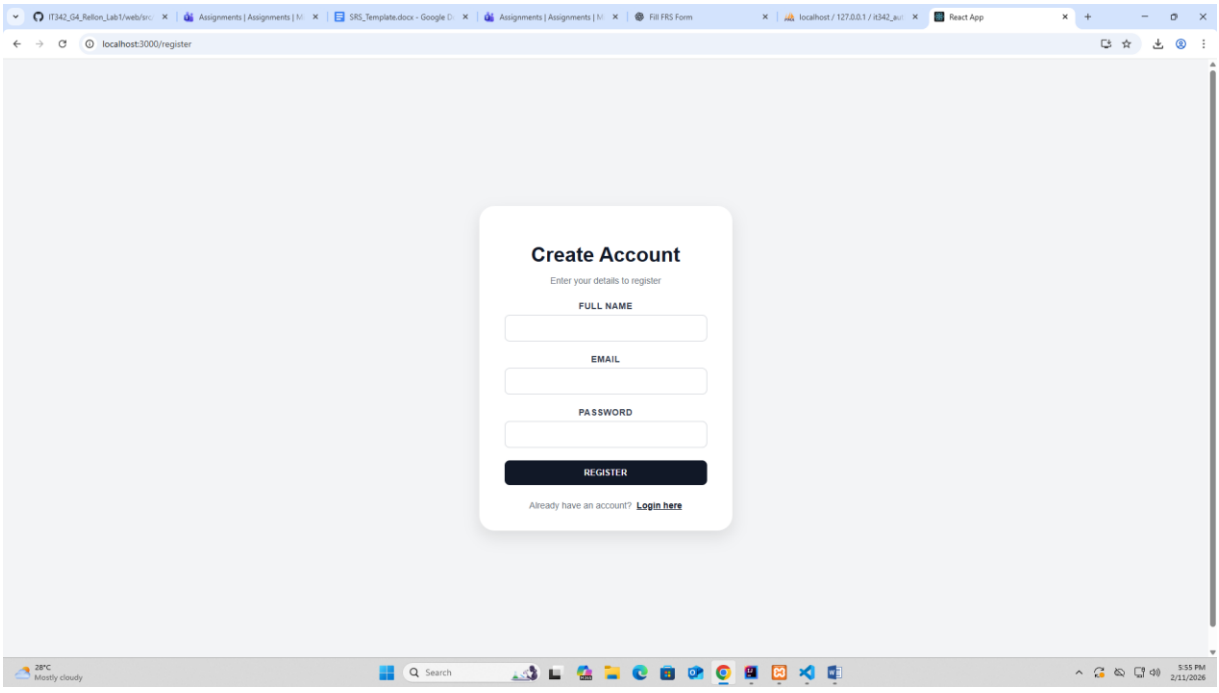
This document serves as the basis for system implementation in the succeeding laboratory activity.

7. Final UI Screenshots

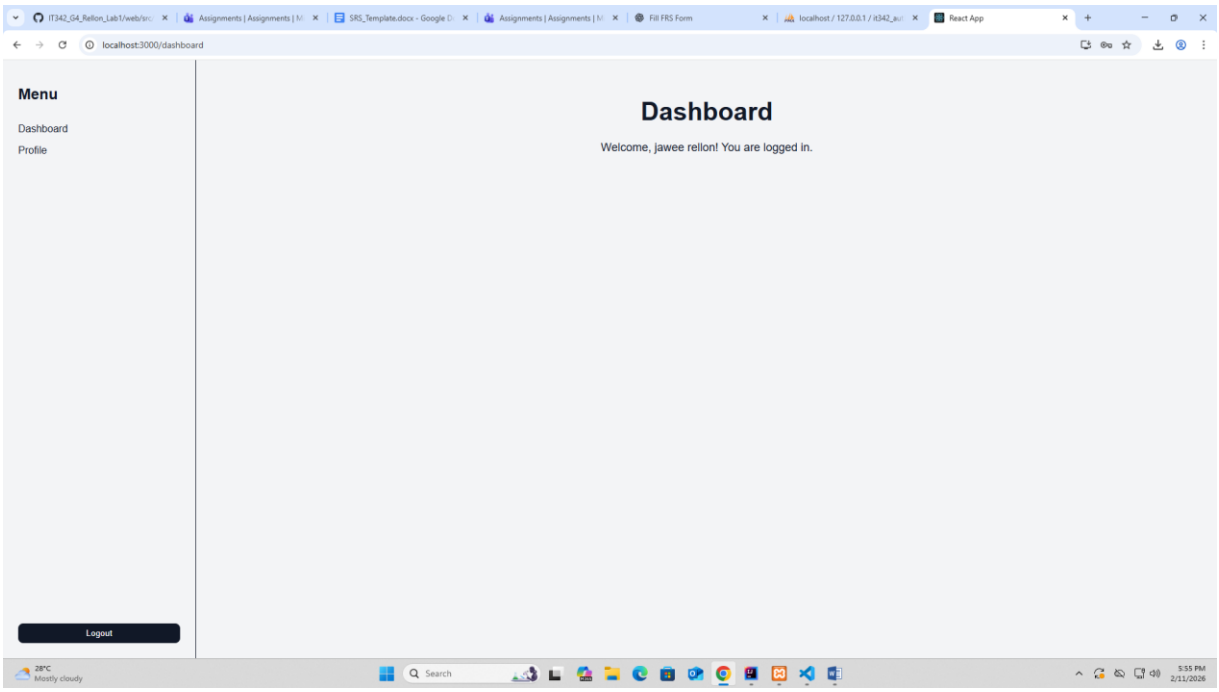
Login



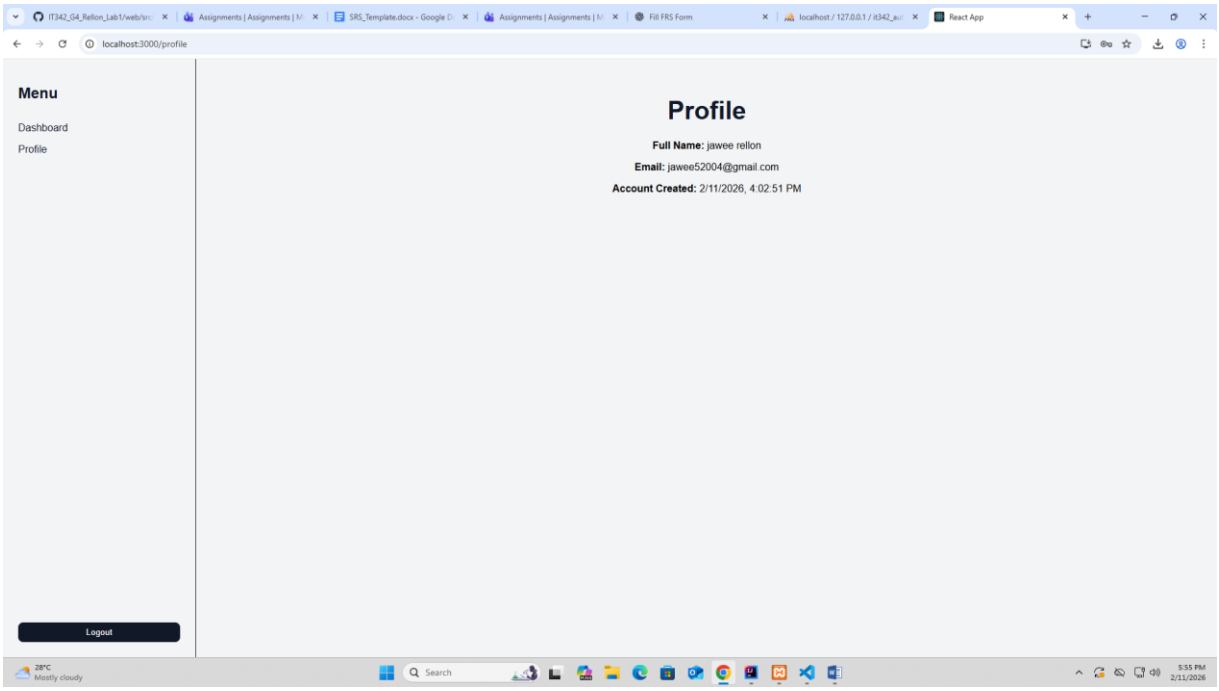
Register



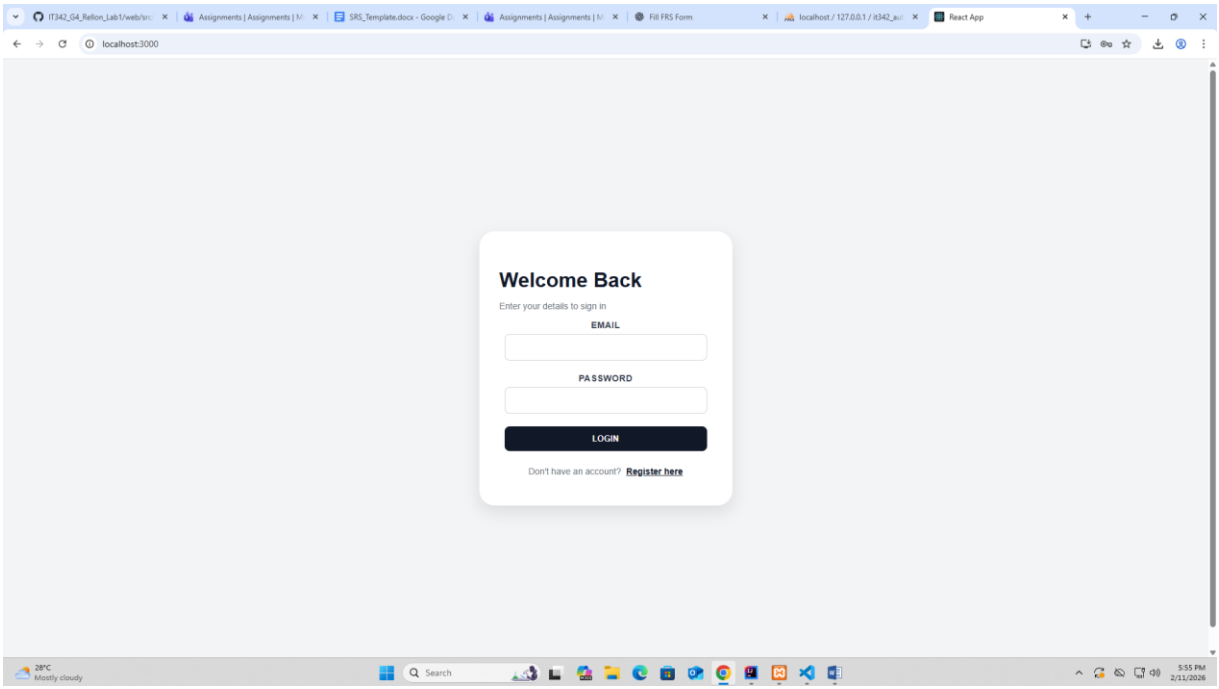
Dashboard



Profile



Logout


localhost:3000
Welcome Back
Enter your details to sign in
EMAIL
PASSWORD
LOGIN
Don't have an account? Register here