



**CEBU INSTITUTE OF TECHNOLOGY**  
**UNIVERSITY**

**IT342- G6**

**SYSTEMS INTEGRATION AND  
ARCHITECTURE 1**

---

**FUNCTIONAL REQUIREMENTS  
SPECIFICATION (FRS)**

---

Project Title: Laboratory Activity: Mini App – User Registration & Authentication

Prepared By: Keint Warren J. Poliquit

Date of Submission: February/14/2026

Version: 1

# Table of Contents

1. Introduction .....	3
1.1. Purpose .....	3
1.2. Scope .....	3
1.3. Definitions, Acronyms, and Abbreviations .....	3
2. Overall Description .....	4
2.1. System Perspective .....	4
2.2. User Classes and Characteristics.....	4
2.3. Operating Environment .....	4
2.4. Assumptions and Dependencies .....	4
3. System Features and Functional Requirements .....	6
3.1. Feature 1: .....	<b>Error! Bookmark not defined.</b>
3.2. Feature 2: .....	<b>Error! Bookmark not defined.</b>
3.3. Feature 3: .....	<b>Error! Bookmark not defined.</b>
3.4. Feature 4: .....	<b>Error! Bookmark not defined.</b>
4. Non-Functional Requirements .....	7
5. System Models (Diagrams) .....	8
5.1. ERD .....	8
5.2. Use Case Diagram .....	8
5.3. Activity Diagram .....	9
5.4. Class Diagram.....	10
5.5. Sequence Diagram .....	11
6. Appendices .....	12

## 1. Introduction

### 1.1. Purpose

A system that would allow users to register, login, view profile and logout, with users being divided into 2 actors; Guests and Authenticated. Guest users are users who are not authenticated to the system as such they can only either login or register an account. Authenticated users can login, view profile, and logout.

### **1.2. Scope**

Allow users to register, login, view profile, and logout. Authenticated users cannot edit profile yet and Guest users cannot view profile and logout as they are not recorded unto the system.

### **1.3. Definitions, Acronyms, and Abbreviations**

- **Guest Users** – Users that are not authenticated/recored in the database.
- **Authenticated Users** – Users that are authenticated/recored in the database.
- **AuthController** – The interface between the user and the system
- **AuthService** - Checks on whether the user is either a guest or an authenticated.
- **UserRepository** – Handles the interaction with the database for the user class, by performing CRUD operations on user data.
- **PasswordEncoder** - Ensures that raw passwords are never stored in plain text, enhancing security.

## **2. Overall Description**

### **2.1. System Perspective**

This system would allow users to access the website/application with their accounts thus allowing them to track their digital footprints and personalize themselves unto the system via storing their data into the system database.

### **2.2. User Classes and Characteristics**

- **Guest Users** – Users that are not authenticated/recored in the database.
- **Authenticated Users** – Users that are authenticated/recored in the database.

### **2.3. Operating Environment**

#### **Hardware:**

- Server
- Client
- Storage
- Network

#### **Software:**

- Operating System
- WebServer
- Database Management System
- Programming Language Runtime
- Web Browser

**Tools:**

- IDE / Code Editor
- Version Control (Git)
- API Testing Tool (Postman)
- Build Tools (Maven / Gradle / npm)
- Authentication Libraries (Spring Security, JWT, bcrypt)
- ERD & Diagram Tools (draw.io, Lucidchart, PlantUML)

**2.4. Assumptions and Dependencies****Assumptions:**

- Users have a device with a browser.
- Each user has a unique account.
- Network connection is available.

**External Dependencies:**

- Database system (MySQL/PostgreSQL).
- Authentication libraries (JWT, bcrypt).
- Web server (Apache/Nginx).
- Optional email service for verification.

### **3. System Features and Functional Requirements**

#### **1. User Registration**

**Description:** Allows a new user to create an account in the system. **Functional Requirements:**

- Users can enter required details (username, email, password).
  - System validates input (e.g., email format, password strength).
  - Passwords are securely hashed before storing in the database.
  - System prevents duplicate accounts with the same email/username.
- 

#### **2. User Login**

**Description:** Allows a registered user to access their account. **Functional Requirements:**

- Users provide email/username and password.
  - System verifies credentials against the database.
  - System generates a session or token for authenticated access.
  - Invalid credentials show an error message.
- 

#### **3. View Profile / Dashboard**

**Description:** Lets logged-in users access their personal information or dashboard. **Functional Requirements:**

- Only authenticated users can access this feature.
  - Displays user-specific data such as username, email, or account settings.
  - Protected pages are not accessible by guests.
- 

#### **4. User Logout**

**Description:** Ends the user's session and prevents further access without login. **Functional Requirements:**

- User can trigger logout manually.

- System invalidates the session or token.
- After logout, protected pages are blocked until login.

#### 4. Non-Functional Requirements

- **Performance:** The system should handle multiple simultaneous users without noticeable delays.
- **Security:** Passwords must be hashed; sessions or tokens must be protected; sensitive data must be encrypted.
- **Usability:** The interface should be user-friendly, intuitive, and accessible via web browsers.
- **Reliability:** The system should operate 24/7 with minimal downtime.
- **Scalability:** The system should support growth in the number of users without major performance loss.
- **Maintainability:** Code should be modular and well-documented for easy updates.
- **Availability:** Protected pages and authentication services should be consistently accessible when required.

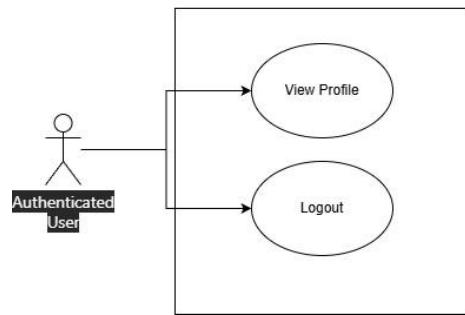
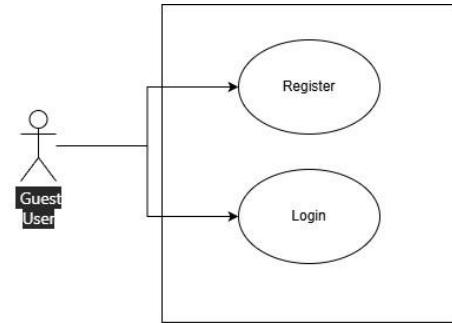
## 5. System Models (Diagrams)

Insert the necessary diagrams for the system:

### 5.1. ERD

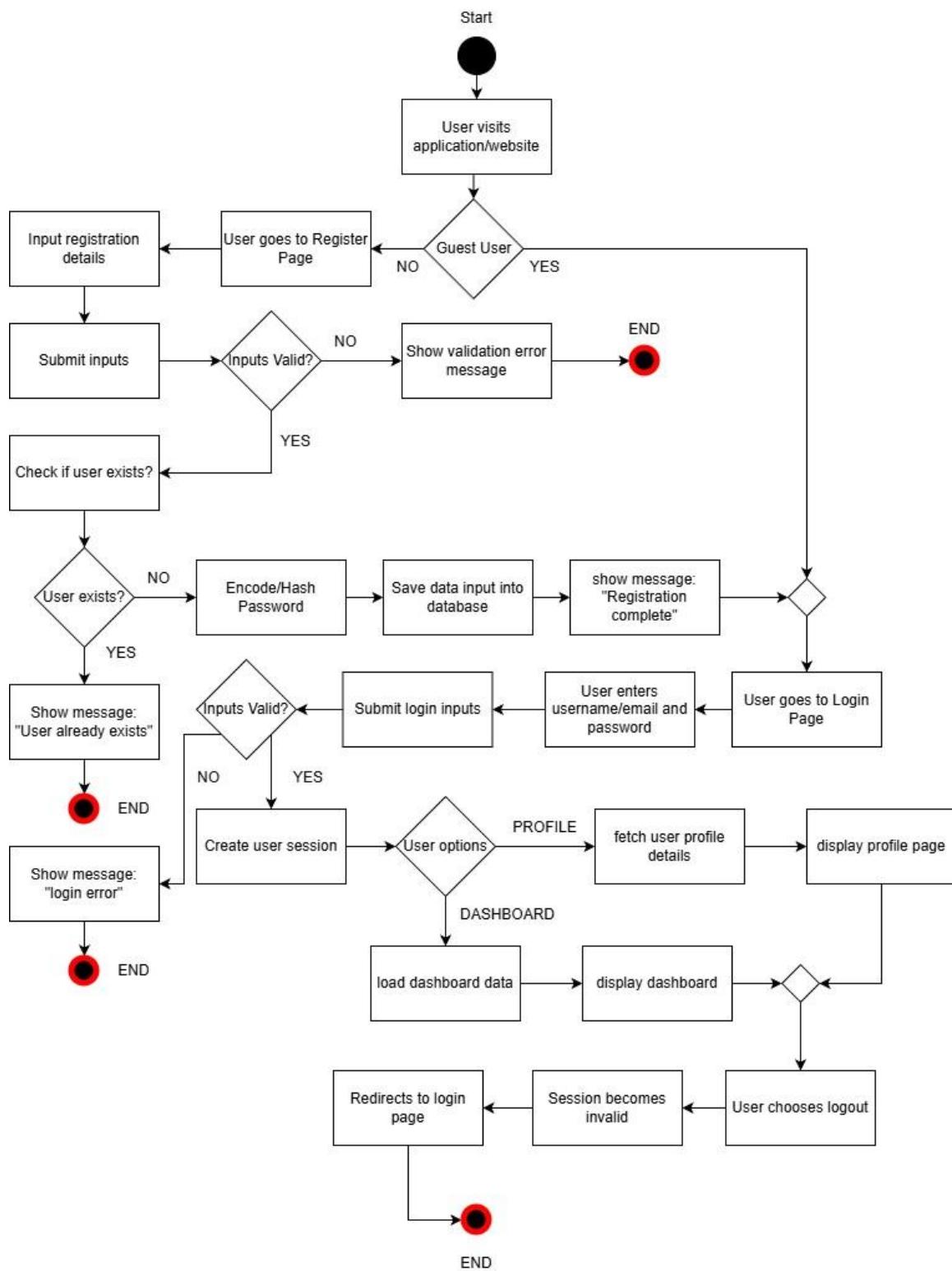
USER	
userID (PK)	varchar(11)
username	varchar(25)
email	varchar(50)
password	varchar(25)
profilePicture	image
profileBio	varchar(100)
address	varchar(100)
gender	varchar(25)
createdDate	date

### 5.2. Use Case Diagram



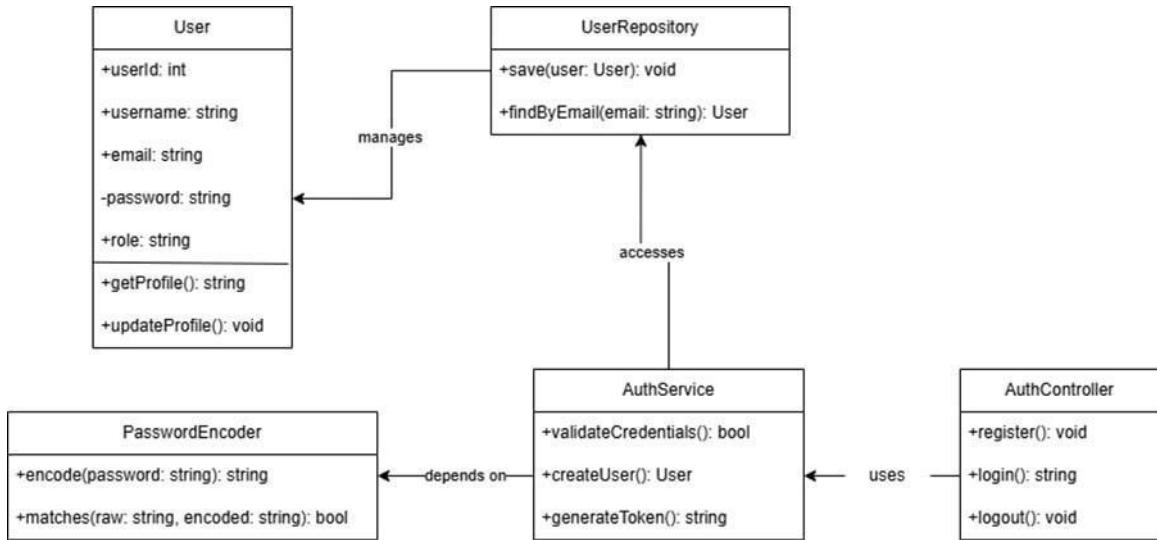
### 5.3.

#### Activity Diagram



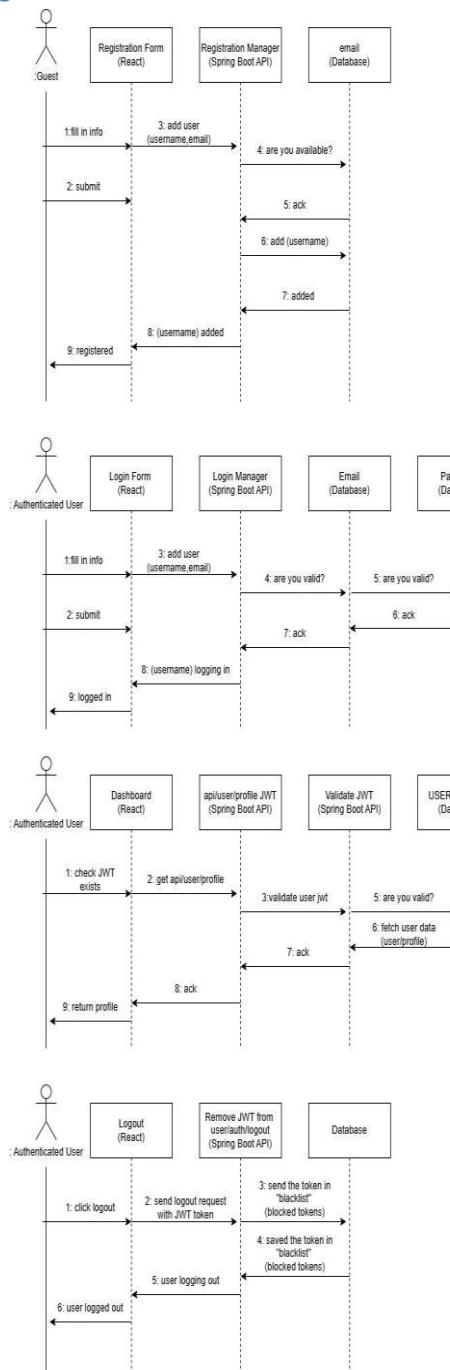
## 5.4.

**Class Diagram**



## 5.5.

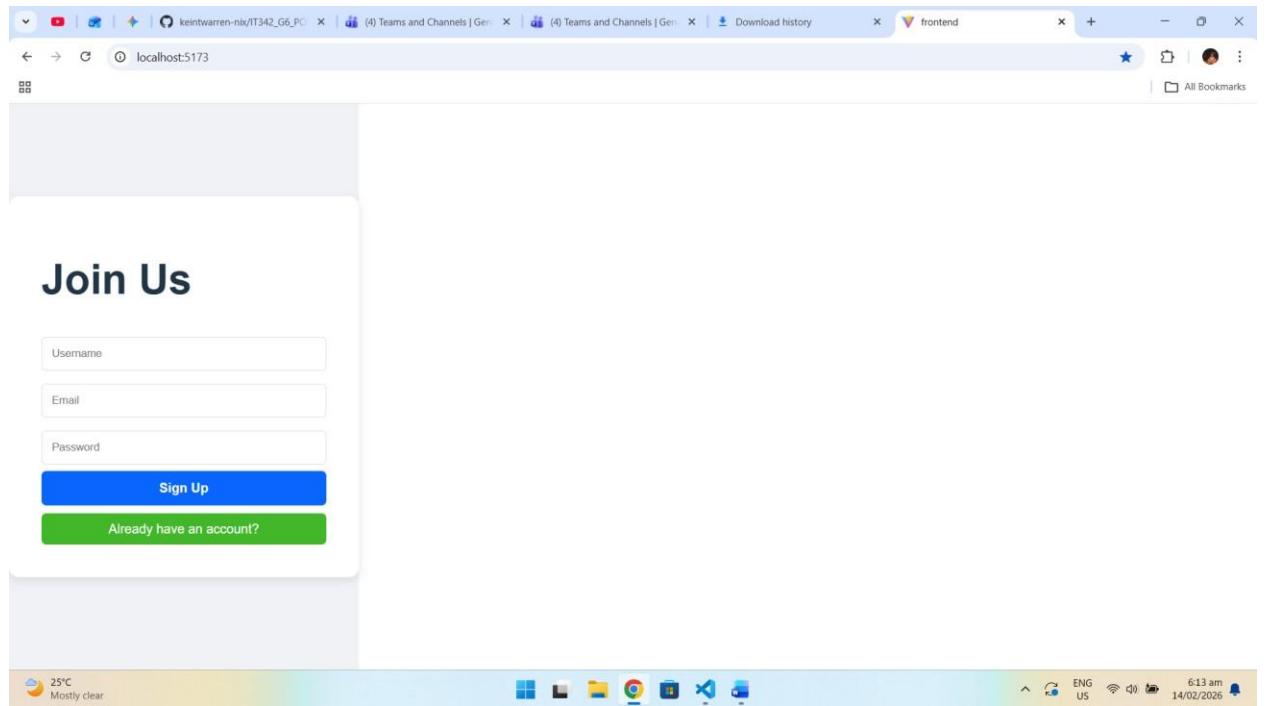
### Sequence Diagram



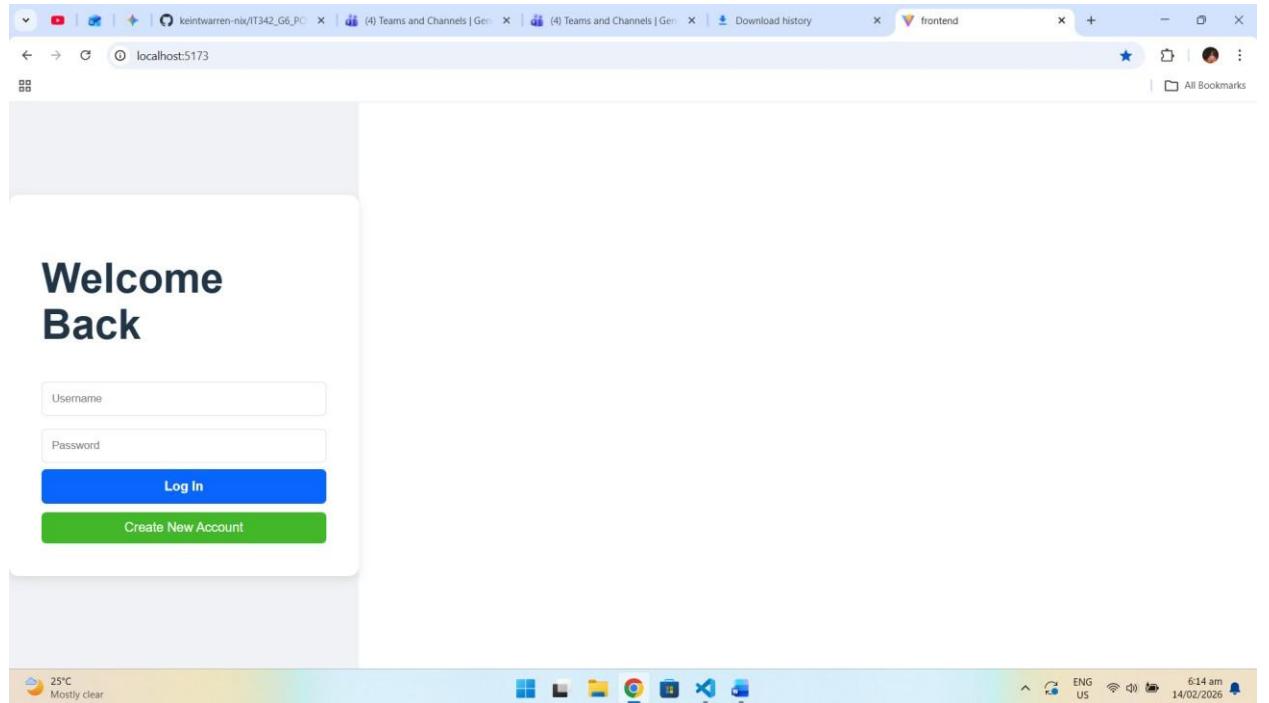
## **6. Appendices**

- <https://www.youtube.com/watch?v=H8ThscWsQV8&list=PLA9oyBlTfuCguMi9PTwvA2ewTPJ7701EI>
- [https://www.reddit.com/r/webdev/comments/13bp9gy/confused\\_about\\_how\\_user\\_registration\\_login\\_system/](https://www.reddit.com/r/webdev/comments/13bp9gy/confused_about_how_user_registration_login_system/)

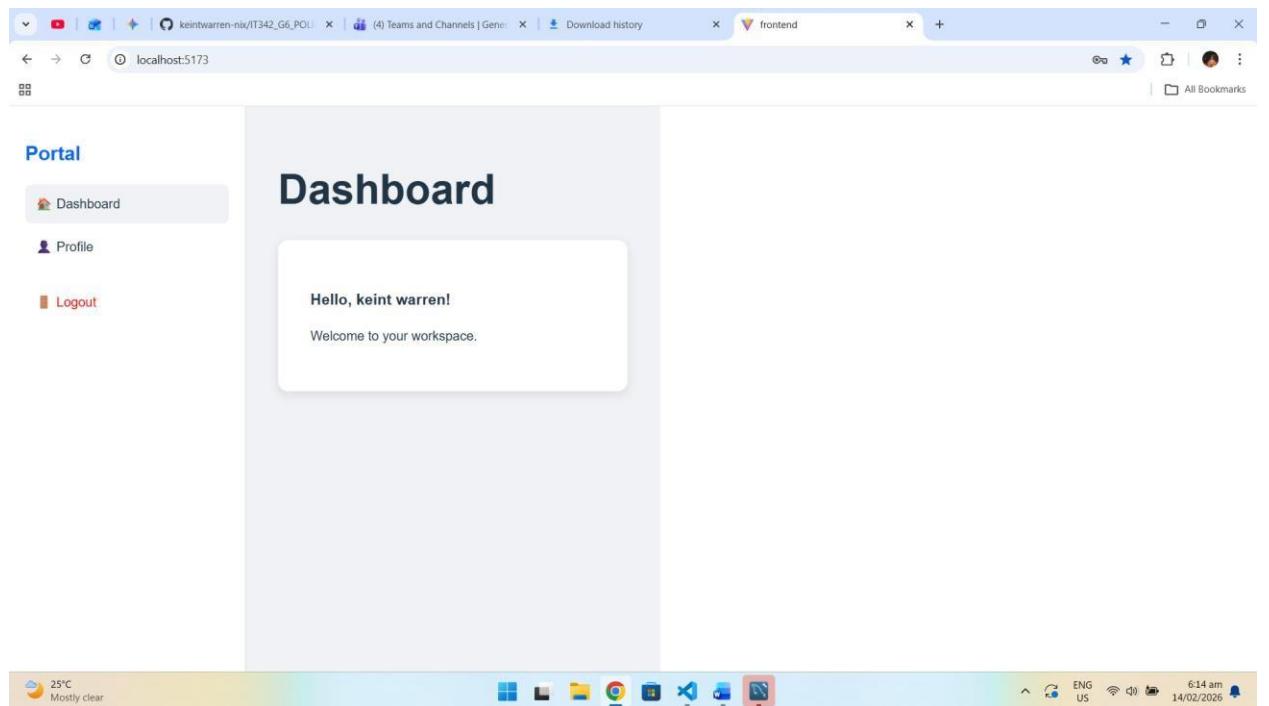
## REGISTER



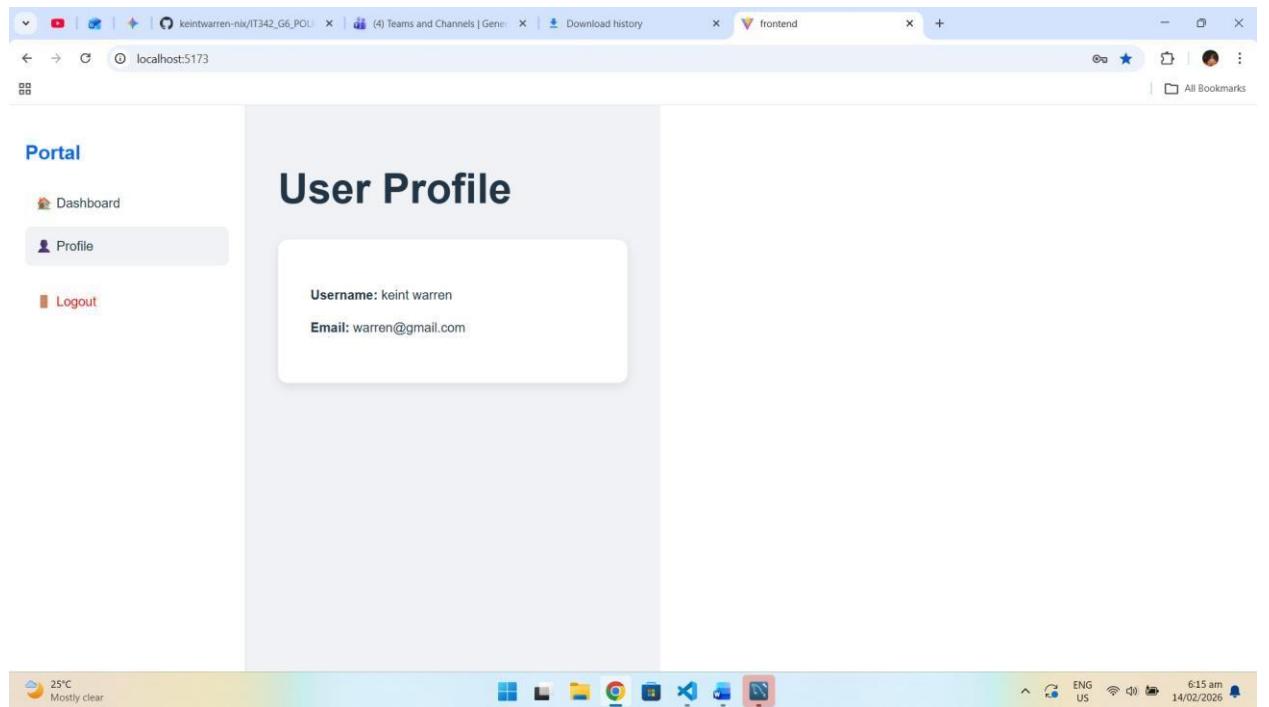
## LOGIN



## DASHBOARD



## PROFILE



LOGOUT

## Portal

Dashboard

Profile

Logout

MOBILE LOGIN

