CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY

IT342-Section
SYSTEMS INTEGRATION AND ARCHITECTURE 1

FUNCTIONAL REQUIREMENTS SPECIFICATION (FRS)

Project Title: Mini App – User Registration & Authentication

Prepared By: Rhyz Nhicco C. Liebtario

Date of Submission: 2/2/2026

Version:

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to describe the requirements and overall design of a user authentication system that supports user registration, login, logout, and access to protected pages. This document is intended for students, instructors, and developers who need to understand the system's functionality and structure before implementation.

1.2. Scope

The system allows users to create an account, log in using their credentials, view their profile or dashboard when authenticated, and log out securely. The system restricts access to protected pages when the user is not logged in. This document focuses only on the system design and documentation. The actual coding and implementation are outside the scope of this activity.

1.3. Definitions, Acronyms, and Abbreviations

1.3.1. **User** – A person who registers and uses the system
1.3.2. **Authentication** – The process of verifying a user's identity
1.3.3. **Authorization** – The process of granting access to protected resources
1.3.4. **ERD** – Entity Relationship Diagram
1.3.5. **UI** – User Interface
1.3.6. **API** – Application Programming Interface
1.3.7. **PK** – Primary Key

2. Overall Description

2.1. System Perspective

The system is a web-based authentication module that works as part of a larger application. It consists of a React-based frontend, a Spring Boot backend API, and a database for storing user information. The authentication system manages user access and ensures that only logged-in users can access protected pages.

2.2. User Classes and Characteristics

- Guest User
  - Has not logged in
  - Can register an account
  - Can access the login page
  - Cannot access protected pages

- Authenticated User
  - Has successfully logged in
  - Can view their profile or dashboard
  - Can log out of the system
  - Can access protected pages.

2.3. Operating Environment

Hardware

- Desktop or laptop computer
- Internet connection

Software

- Web browser (Chrome, Edge, Firefox)
- React for frontend UI
- Spring Boot for backend API
- Relational Database (e.g., MySQL)

Tools

- draw.io / diagrams.net for system diagrams
- IDE (e.g., IntelliJ, VS Code) for development

2.4. Assumptions and Dependencies

2.4.1. Users have access to a stable internet connection
2.4.2. The system depends on a working database connection
2.4.3. Passwords are stored securely using hashing
2.4.4. The backend API is available and running
2.4.5. Authentication state is maintained using session or token-based mechanisms

3. System Features and Functional Requirements

Describe each major feature of the system and its functional requirements.

3.1. Feature 1: User Registration

Description: Allows a guest user to create a new account by providing required personal and login information.
Functional Requirements:
- The system shall allow users to register using an email and password
- The system shall validate input fields before saving data
- The system shall prevent duplicate email registrations

3.2. Feature 2: User Login and Logout

Description: Allows registered users to log in to the system, access protected pages, and log out securely.
Functional Requirements:
- The system shall authenticate users using email and password
- The system shall allow access to protected pages only when authenticated
- The system shall invalidate the session or token upon logout

4. Non-Functional Requirements

Security

- Passwords must be stored in hashed form
- Unauthorized users cannot access protected pages

Usability

- The system should be easy to use and understand
- Error messages should be clear and informative

Performance

- Login and registration responses should be processed within an acceptable time

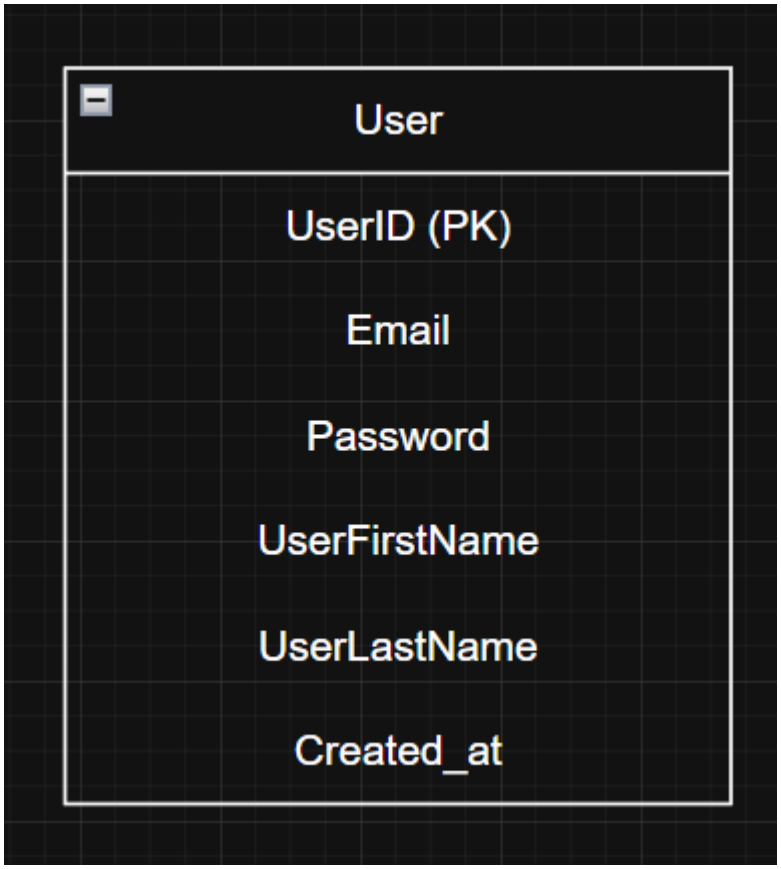
Reliability

- The system should handle invalid login attempts properly
- User sessions should be managed consistently

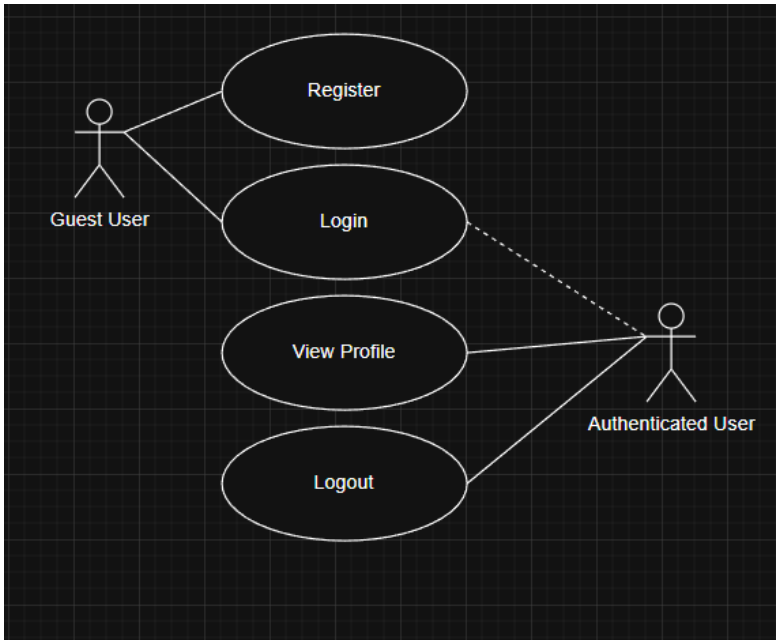
5. System Models (Diagrams)

Insert the necessary diagrams for the system:

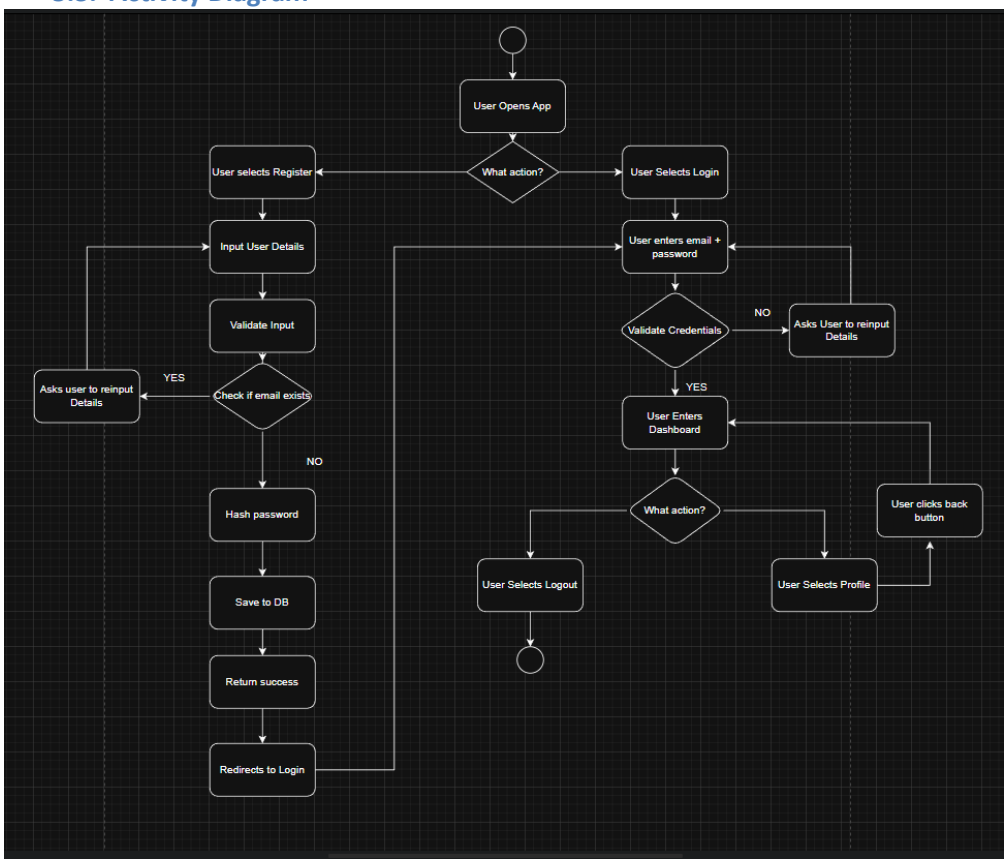
5.1. ERD



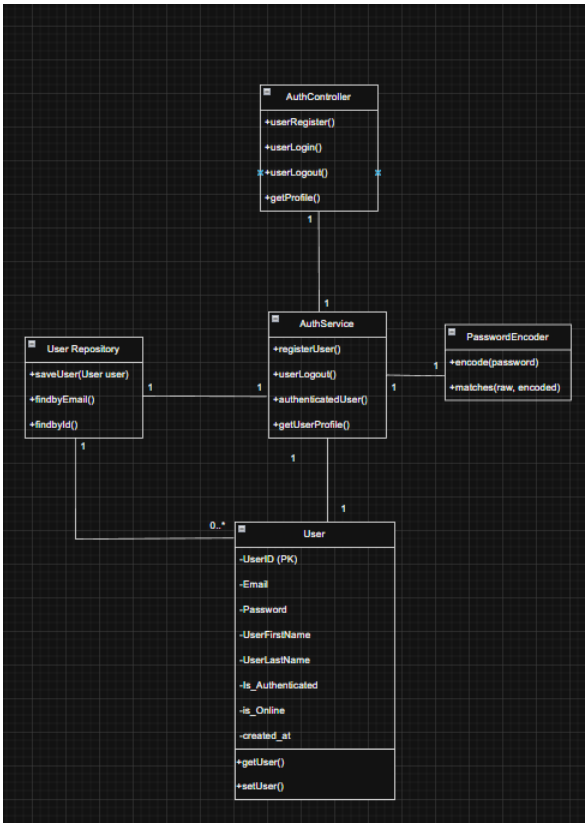
5.2. Use Case Diagram



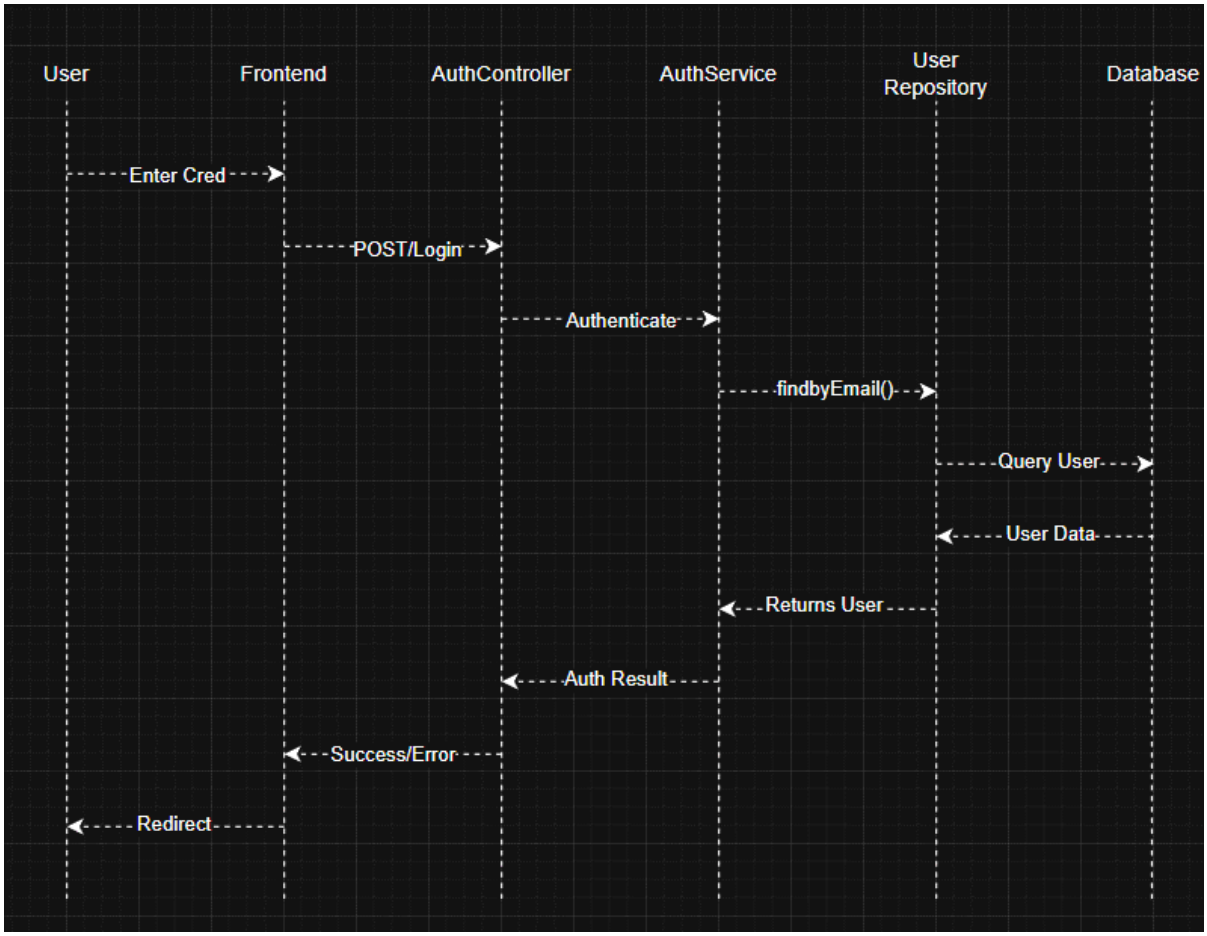
5.3. Activity Diagram



5.4. Class Diagram



5.5. Sequence Diagram

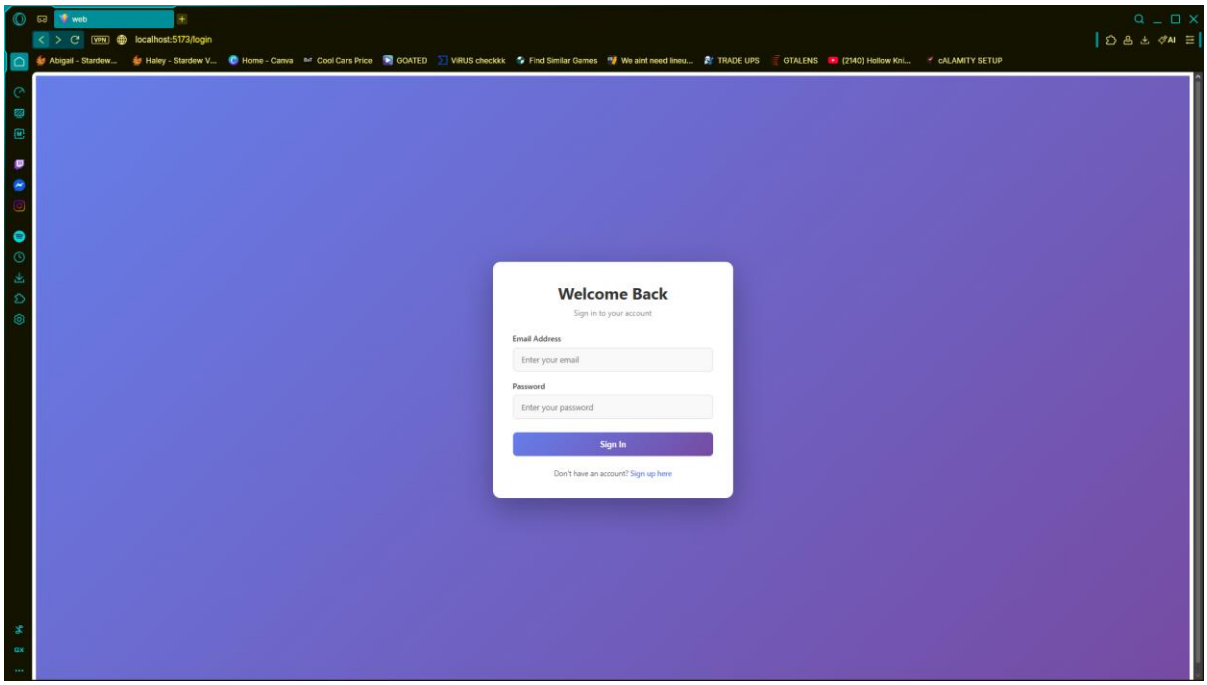


6. Appendices

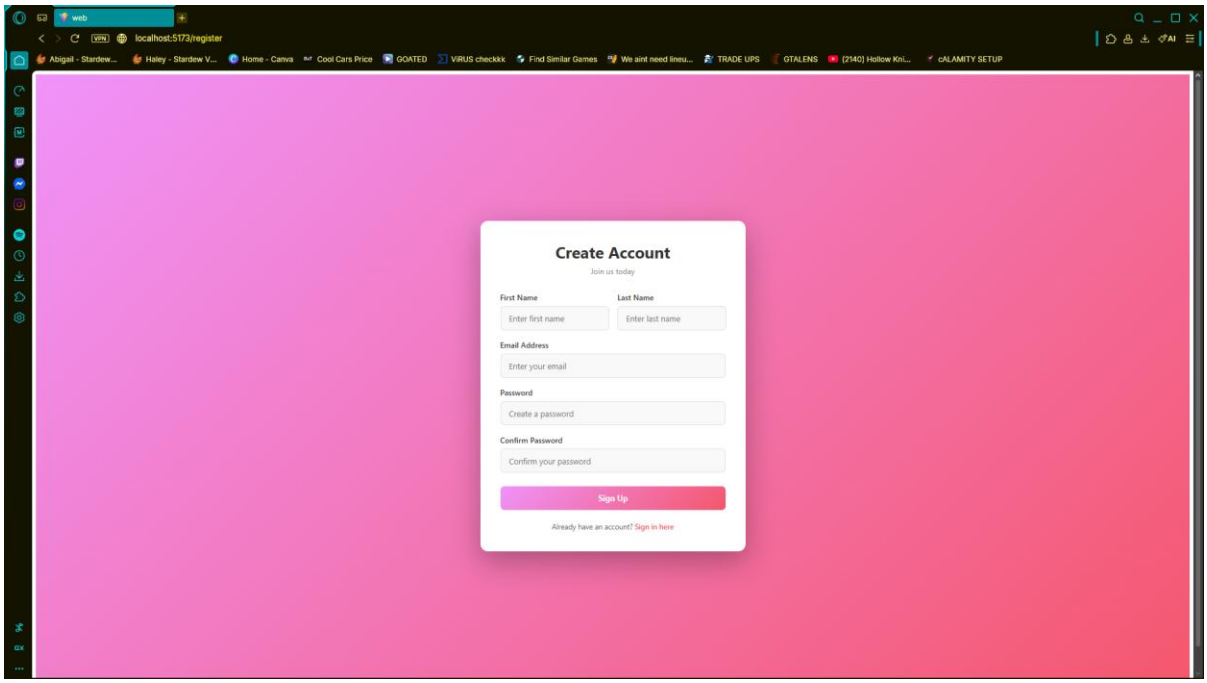
Include any additional information, references, or support materials.

7. Screenshots of Web App

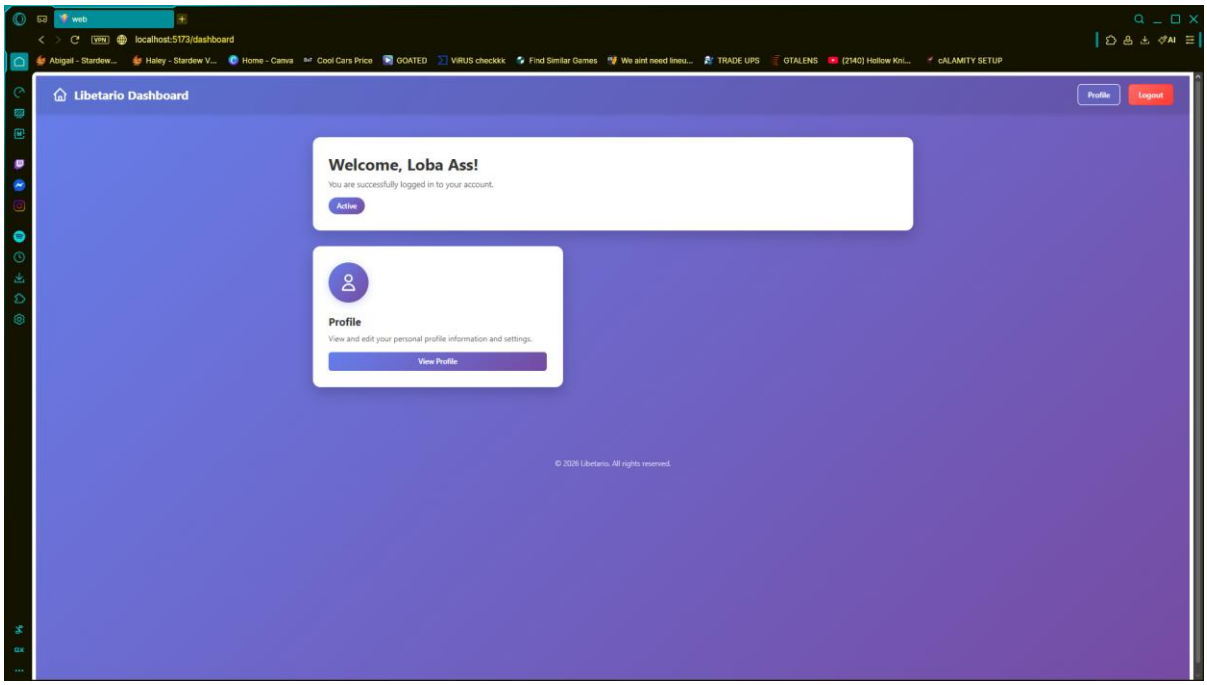
7.1. Login

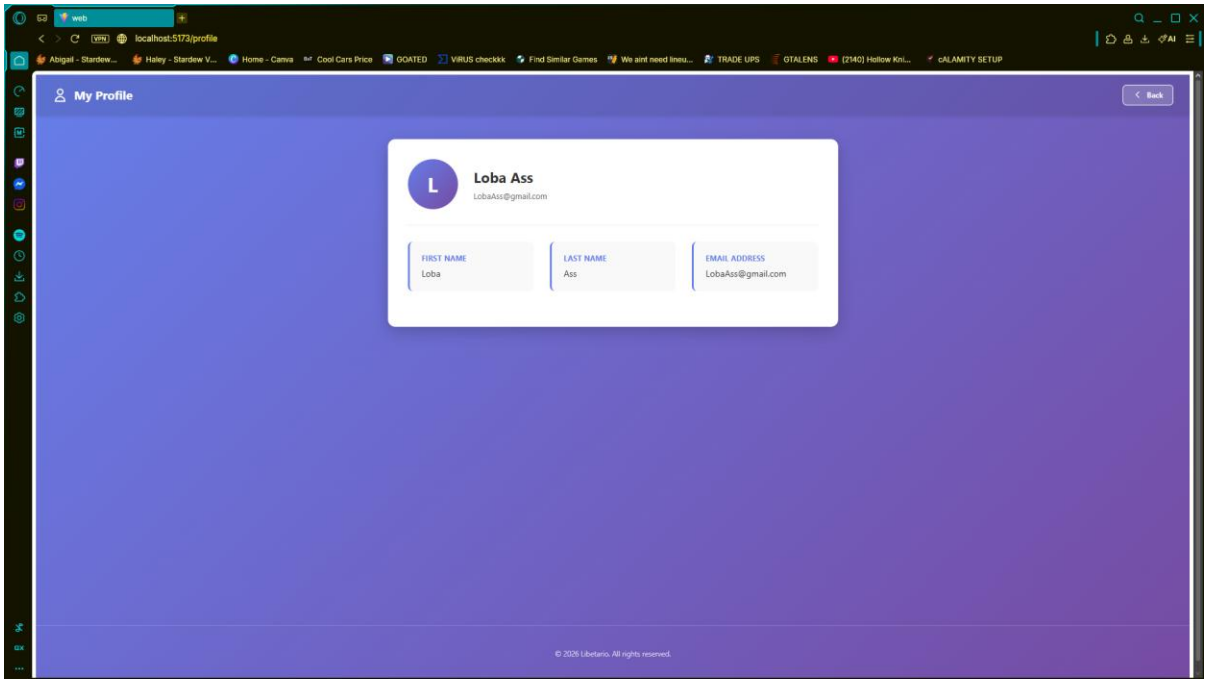


7.2. Register

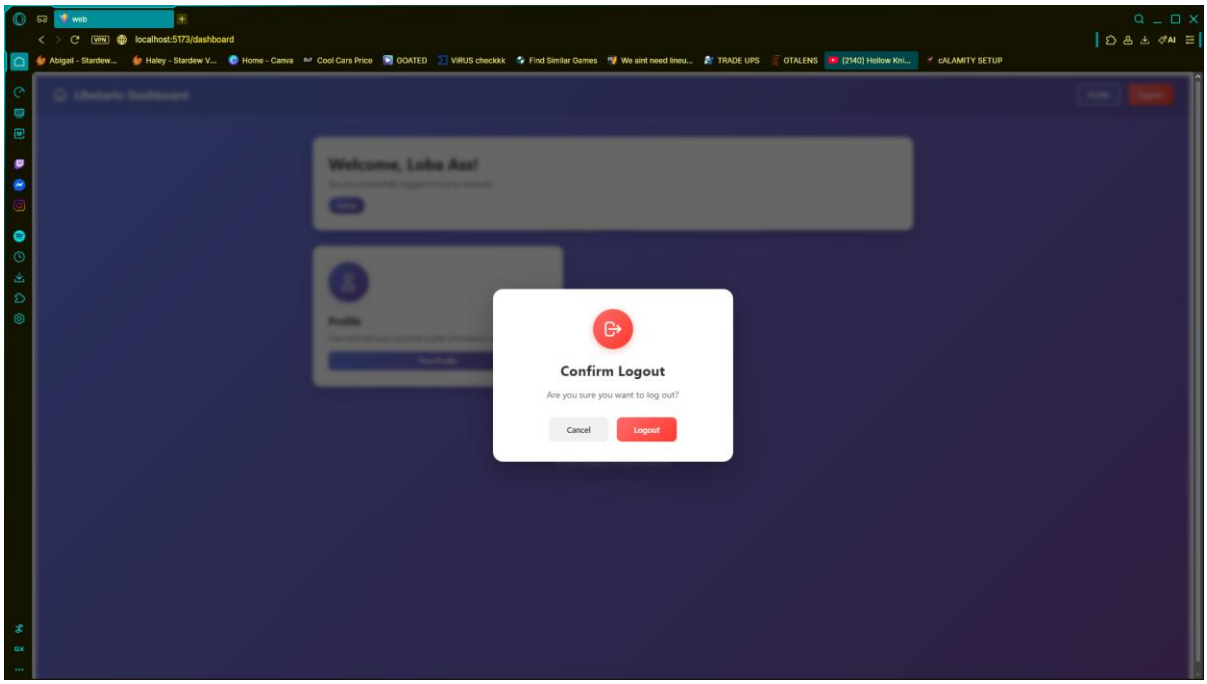


7.3. Dashboard/Profile





7.4. Logout


web
localhost:5173/dashboard
Abigail - Stardew...
Haley - Stardew V...
Home - Canva
Cool Cars Price
GOATED
VIRUS checkkk
Find Similar Games
We aint need lineu...
TRADE UPS
GTALENS
(2140) Hollow Kni...
cALAMITY SETUP
Welcome, Lobo Axl!
Profile
Confirm Logout
Are you sure you want to log out?
Cancel
Logout