Project Title: FASAL

Student Name(s): Saimanti Dey,

Course/batch and Institute name: B.Tech 6th sem , Kalinga Institute of Industrial Technology

Project Guide / Mentor Name: Mr. Rounak Biswas

Period of Internship: 14th Jan 2025 - 30th April 2025

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This project, "FASAL," is a collaborative effort with the Ministry of Agriculture, designed to predict various crop yields throughout the year. As a part of this internship, the primary objective was to develop an Application Programming Interface (API) to facilitate these predictions. The project involved creating a user authentication system using HTML and CSS for the frontend and SQL for the backend to manage user data securely. This report details the methodology, development process, and the core components of the API for crop yield prediction.

2. Introduction

The agricultural sector is a cornerstone of the economy, and accurate crop yield prediction is crucial for food security, policy making, and economic stability. The FASAL project addresses this critical need by developing a robust system to forecast crop yields across different seasons. This project's relevance stems from its direct application in agricultural planning and resource allocation.

The technology involved primarily focuses on API development using Python with FastAPI for the backend, integrated with a PostgreSQL database for data management. The user interface is built with fundamental web technologies: HTML and CSS. This report outlines the background of the project, the technologies employed, and the procedure used to achieve the project's purpose of providing timely and accurate crop yield predictions.

The relevance of this component lies in ensuring that sensitive agricultural forecasts are accessible only to verified stakeholders such as government officials, agricultural scientists, and field officers.

3. Project Objective

The main objective of the FASAL project is to develop an API for predicting various crop yields throughout the year in collaboration with the Ministry of Agriculture. The specific objectives are:

- To design and implement a secure user authentication system for accessing the crop prediction services.
- To develop a robust backend API using Python (FastAPI) capable of handling crop yield prediction requests.
- To establish and manage a relational database (SQL) for storing user credentials and potentially crop-related data.
- To create a basic yet responsive web-based frontend using HTML and CSS for user login and interaction.
- To ensure the API is scalable and maintainable for future enhancements in crop prediction models.

.

4. Methodology

4.1 Tools and Technologies Used

- Frontend: HTML, CSS
- Backend: FastAPI (Python)
- Database: PostgreSQL
- Authentication: passlib for password hashing
- Development Environment: Python 3.x, PostgreSQL server, Visual Studio Code

4.2 Workflow Overview

1.Setup FastAPI Application:

- Installed FastAPI and Uvicorn for running the application.
- Configured Jinja2 templates for rendering HTML pages.

2.Designing the Frontend:

- Created a responsive login page using HTML and CSS.
- Added error handling to display login messages.

3.Setting Up the Database:

- Created a PostgreSQL database named mydatabase.
- Defined a users table with fields for username and hashed password.

4.Implementing Login Functionality:

- Developed routes for GET (display form) and POST (submit credentials).
- Used SQLAlchemy ORM for interacting with the database.
- Implemented password verification using bcrypt.

5.Testing the System:

- Inserted sample test users into the database.
- Tested login functionality with valid and invalid credentials.

4.3 Data Collection and Management

Since this is an API-based authentication module, no external datasets were collected. However, user credentials were stored securely in the PostgreSQL database. Sample users were inserted manually for testing purposes.

5. Data Analysis and Results

ince the project focuses on API development for crop yield prediction and currently implements a login system, the "Data Analysis and Results" section will primarily focus on the functional aspects of the developed API and its components.

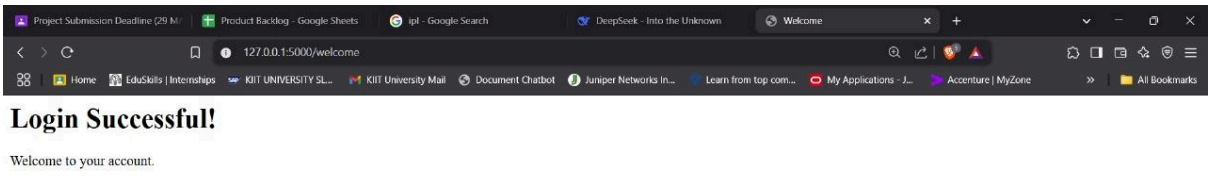
User Authentication System:

- The system successfully renders the `login.html` page when accessing the `/login` endpoint.
- Upon submission of correct credentials (e.g., 'admin' and 'password123' or 'user1' and 'mypassword'), the system successfully authenticates the user and redirects them to the `/welcome` page, displaying "Login Successful!".
- Incorrect credentials lead to an "Invalid username or password" error displayed on the login page.
- The password hashing using `bcrypt` ensures secure storage and verification of user passwords.

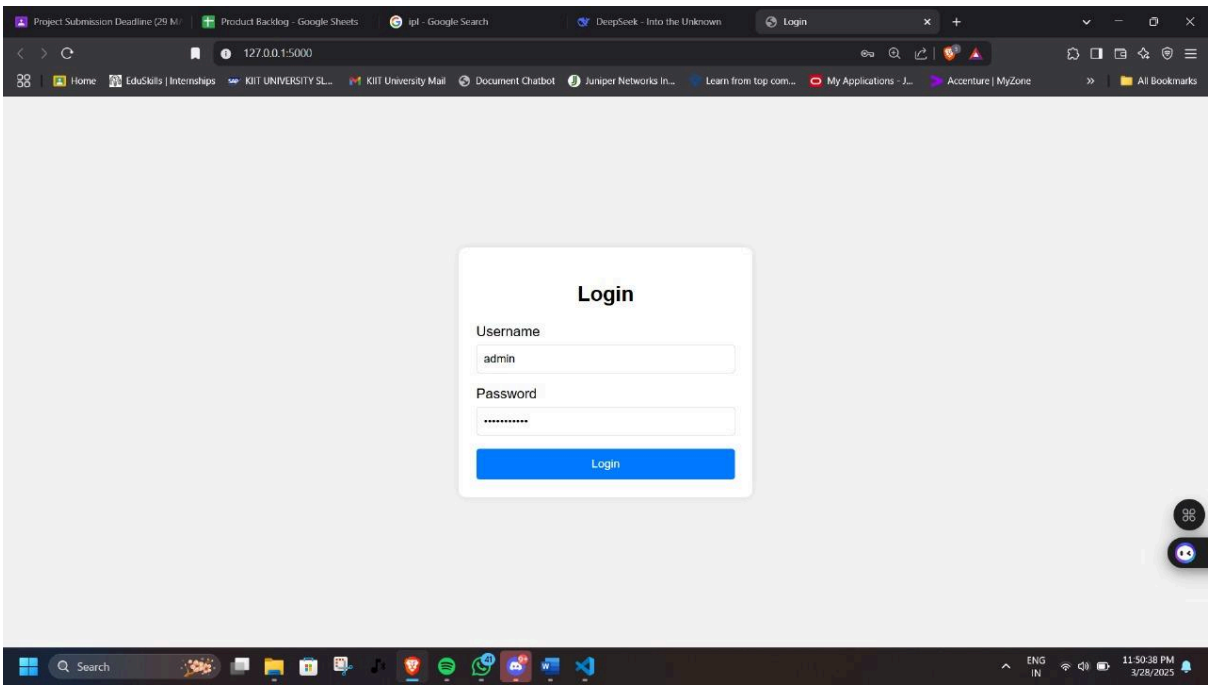
API Functionality:

- The FastAPI application (`main.py`) successfully handles GET and POST requests for the login functionality.
- The integration with the PostgreSQL database through SQLAlchemy allows for persistent storage and retrieval of user data.
- The defined `User` model and `get_db` dependency ensure proper interaction with the database

Screenshots:



- 

- 

6. Conclusion

The FASAL project has successfully established a foundational API with a secure user authentication system, serving as the entry point for the broader crop yield prediction initiative. The developed login functionality using FastAPI, HTML, CSS, and PostgreSQL is robust and demonstrates the feasibility of building a comprehensive agricultural data platform. This initial phase provides a solid framework upon which the core crop yield prediction models can be integrated.

Recommendations for Future Work:

Implement the actual crop yield prediction models within the API, potentially using machine learning algorithms.

- Integrate real-time or historical agricultural datasets for training and prediction.
- Develop additional API endpoints for submitting crop-related data and retrieving predictions.
- Enhance the frontend with interactive dashboards and visualization tools for crop yield data.
- Implement more advanced security measures, such as token-based authentication (e.g., JWT).
- Consider deploying the application to a cloud platform for wider accessibility and scalability.

7. APPENDICES

You may create separate Appendix for the following:

1. References

- FastAPI Documentation: https://fastapi.tiangolo.com/
- SQLAlchemy Documentation: https://docs.sqlalchemy.org/
- PostgreSQL Documentation: https://www.postgresql.org/docs/
- Passlib Documentation: https://passlib.readthedocs.io/

2.Survey Questionnaire (if any): Not applicable

3.Code extract : (https://github.com/Saimanti15124/FASAL-login-page)

4.DocumentLink:

(Googledrive: https://docs.google.com/document/d/1yVewrmt1PLTK598azNPfQLDf2p9HXuXHPYV5ZfbJJjQ/edit?usp=sharing)

(GitHub:)