**Name : I Nyoman Putra Satria / 001202300126**

**Class : IT 3**

**Subject : Distributed and Parallel System**

**--------------------------------------------------------------------**

**Project Documentation: Flask-Based Web Application for User-Specific Notes Management**

**1. Introduction**  
This document provides a report on the development of a Flask-based web application designed to manage user authentication and user-personal note-taking. The project was constructed with enhancements such as dynamic note deletion and Docker integration. This report outlines the objectives, technical components, development process, and deployment configuration of the application.

**2. Project Objectives**  
The primary goals of the project are as follows:

* Develop a web application with a secure user login and registration system.
* Implement Create, Read, and Delete operations for personal notes.
* Ensure user data isolation, so each user can only access and manipulate their own notes.
* Enable interaction with using JavaScript for a smoother user experience.
* Package the application using Docker for platform-independent deployment.

**3. Technology Stack**

* **Backend Framework:** Flask (Python)
* **Frontend:** HTML, CSS, JavaScript (with Jinja2 templating)
* **Database:** SQLite (via SQLAlchemy)
* **Authentication:** Flask-Login
* **Containerization:** Docker

**4. Project Structure**

A screenshot of a computer program

AI-generated content may be incorrect.

**5. Core Functionalities**

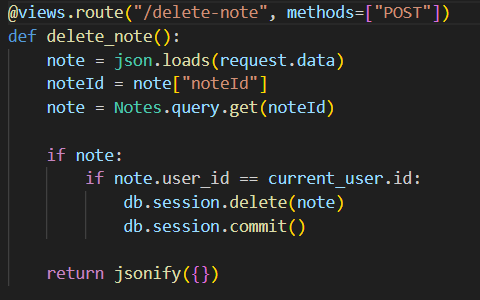
* **User Registration and Login:**  
  Users can register for an account and securely log in. The system uses hashed passwords via Werkzeug for security. Flask-Login manages user sessions and access control.
* **Notes CRUD (Create, Read, Delete):**  
  Authenticated users can add new notes and delete existing ones. All notes are user-specific.
* **Dynamic Deletion with JavaScript:**  
  A delete button implemented in each note row sends a POST request using JavaScript. Flask handles the backend deletion without requiring a full page reload.

**6. JavaScript Integration for Deletion**  
A JavaScript function in index.js sends asynchronous requests:

A screen shot of a computer code

AI-generated content may be incorrect.

On the backend, the corresponding route in Flask for deletion is:



**7. Docker Integration**  
The project includes a Dockerfile to containerize the application. A simplified example is:

A screenshot of a computer program

AI-generated content may be incorrect.

**8. Conclusion**  
This Flask-based web application demonstrates a full-stack implementation of a secure, dynamic note-taking platform. Through its use of Flask-Login, SQLAlchemy, and JavaScript, it enables smooth user interaction. The Docker container ensures easy deployment and environment consistency. This project serves as a foundational template for more advanced web systems that involving user-specific data and real-time interactivity.

**9. References**

* YouTube Video Tutorial: <https://youtu.be/dam0GPOAvVI>
* Flask Documentation: <https://flask.palletsprojects.com/>
* Flask-Login: <https://flask-login.readthedocs.io/>
* SQLAlchemy: <https://www.sqlalchemy.org/>
* Docker: <https://docs.docker.com/>