# SIMPLE NOTES APP

Project Report

Submitted by: Shaik Mujasim

Department of Computer Science & Engineering

KLM College of Engineering for Women

## 1. Introduction

The Simple Notes App is a Flask-based web application that enables users to create, edit, delete, and manage notes efficiently. Designed with a minimalistic interface, this app ensures easy accessibility and smooth functionality for organizing notes digitally.

It includes CRUD (Create, Read, Update, Delete) operations, a feature to mark notes as important, and dark mode support for enhanced readability. The application ensures data persistence using SQLite and provides a seamless user experience with real-time updates.

## 2. Objectives

• To develop an efficient and user-friendly note-taking application.

• To provide CRUD functionalities for managing notes.

• To allow users to mark important notes for quick access.

• To implement dark mode support for better user experience.

• To use SQLite for structured data storage.

This project aims to enhance productivity by offering a well-organized way to store and retrieve notes.

## 3. Technologies Used

### Backend Technologies

• Flask – Python web framework for handling requests.

• Flask SQLAlchemy – ORM for database interactions.

• Flask Migrate – Handles database migrations.

• SQLite – Lightweight database for storing notes.

### Frontend Technologies

• HTML – Structuring the web page.

• CSS – Styling the interface.

• JavaScript – Adding interactivity.

### Additional Features

• Dark Mode Support – Users can switch between light and dark themes.

• AJAX Requests – Real-time data updates without page refresh.

• Local Storage & Data Persistence – Notes remain saved even after closing the application.

## 4. System Architecture

### User Interface (UI) Layer

The frontend is built using HTML, CSS, and JavaScript to ensure an interactive and responsive design. Users can add, edit, delete, and mark notes as important. A dark mode toggle enhances readability based on user preference.

### Backend Layer

Flask handles requests and routes using the following key endpoints:  
• / – Home route displaying all notes.  
• /get\_notes – API endpoint that returns stored notes in JSON format.  
• /add – Handles note creation via POST requests.  
• /edit/<id> – Updates an existing note.  
• /delete/<id> – Deletes a specific note.  
• /mark\_important/<id> – Toggles the important flag for a note.

### Database Layer

SQLite is used to store note details, including title, content, and importance status. The Note model in SQLAlchemy defines the table schema.

## 5. Features & Functionalities

• Add Notes – Users can create a note by entering a title and content.

• Edit Notes – Users can modify an existing note.

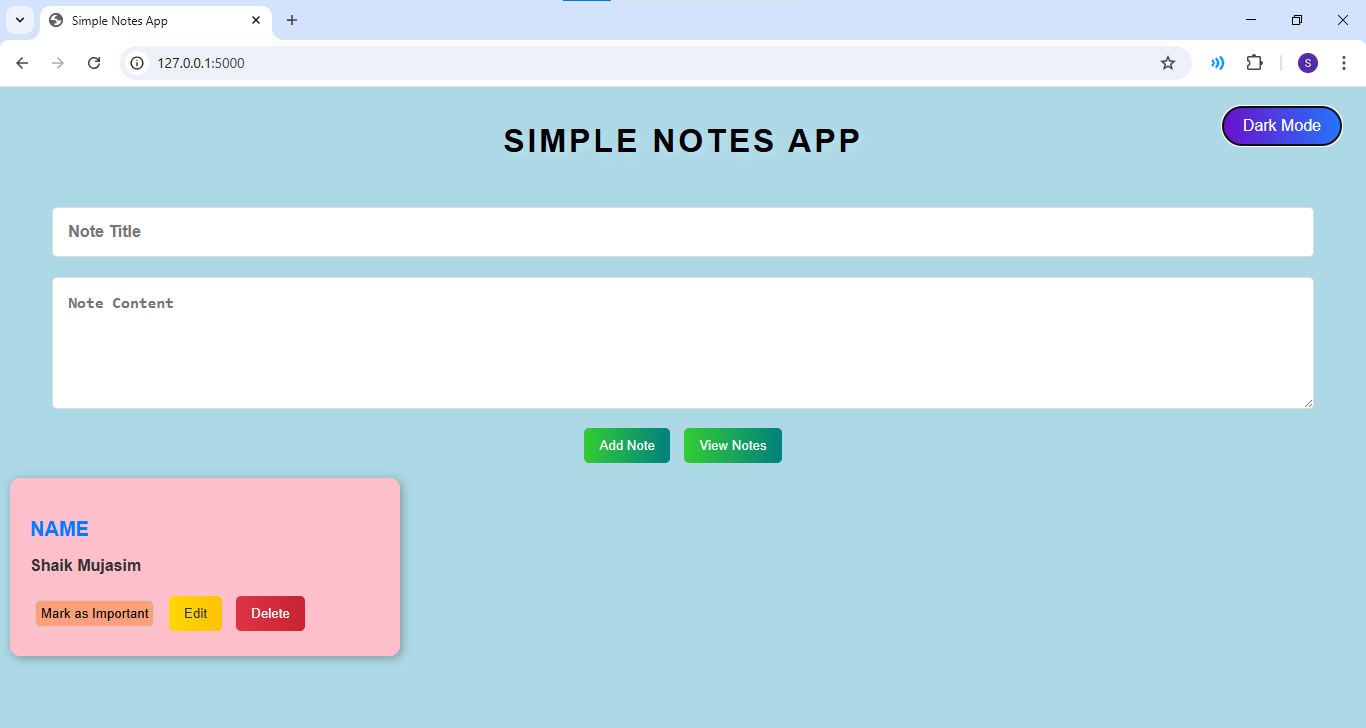
• Delete Notes – Users can remove notes.

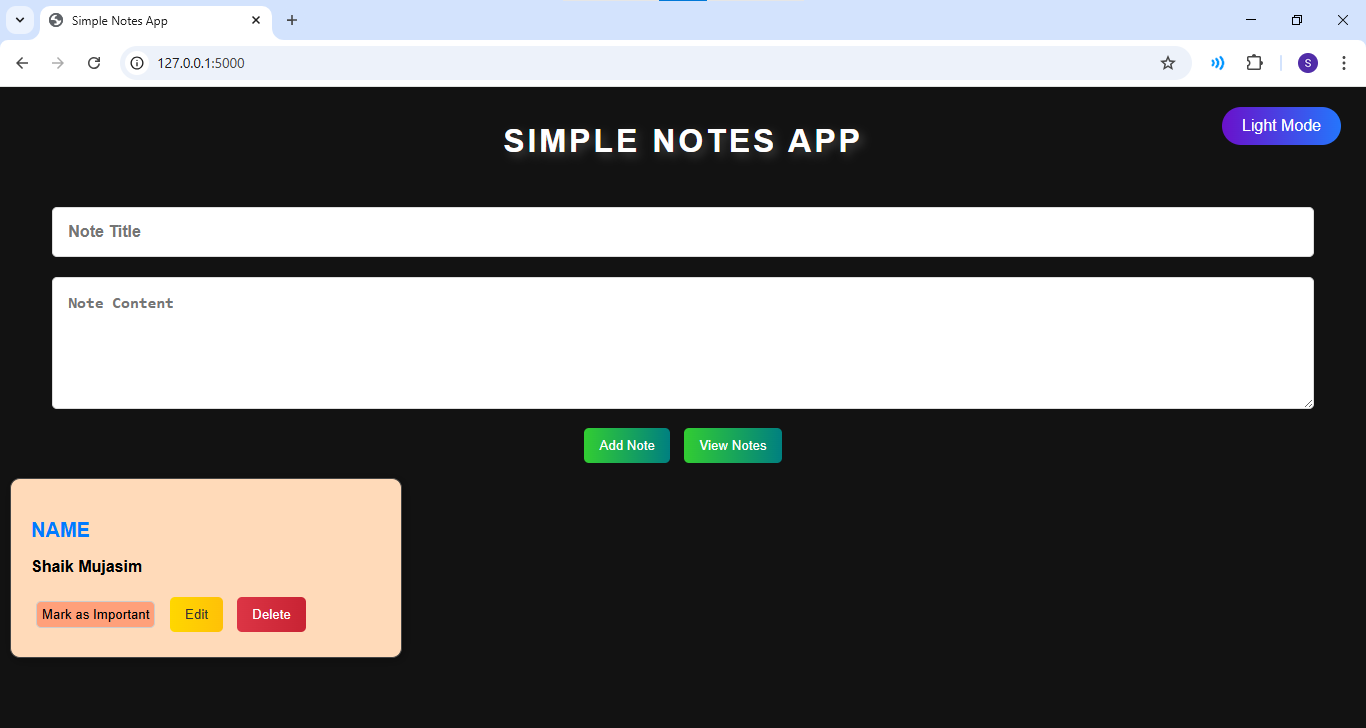
• Mark Notes as Important – Users can highlight key notes.

• Dark Mode Support – Toggle between light and dark mode.

• AJAX-based Data Fetching – View Notes without refreshing the page.

## 6. Screenshots





## 7. Challenges Faced & Solutions

• Implementing CRUD Operations Efficiently: Ensured smooth data persistence using Flask SQLAlchemy ORM.

• Implementing Dark Mode: Used JavaScript local storage to save user preference.

• Handling Note Deletion & Updates: Used AJAX requests for real-time updates without page reloads.

• Preventing Accidental Deletion: Added confirmation prompts before deletion.

## 8. Conclusion

The Simple Notes App successfully provides an efficient digital note-taking solution with CRUD operations, dark mode, and an importance marker. The development of this project has enhanced my hands-on experience in full-stack web development, including backend development using Flask, frontend UI design, and database management using SQLite.