Flask Application Documentation

PREPARED FOR

Procto

PREPARED BY

Harsh Raj 200303125083

200303125083@paruluniverssity.ac.in

SEPT, 2023

Dear Procto Team,

I hope this message finds you well.I am writing to inform you of the successful completion of the Python/Flask task assigned to me, which involved the development of a web application for note management.

I am pleased to share that I have built a fully functional Flask web application that allows users to create, view, edit, and delete their notes. This application adheres to the CRUD (Create, Read, Update, Delete) principles and offers an intuitive and user-friendly interface for note management.

I have followed best practices in Python and Flask development to ensure the application's functionality and maintainability. The application also utilises SQLAlchemy for efficient database interactions, guaranteeing data persistence.

I have thoroughly enjoyed working on this task, and it has been a great learning experience for me. I have attached the necessary files and code for your review.

Thank you for providing me with this opportunity to showcase my skills, and I look forward to the next steps in the evaluation process.

Thanking You,

Harsh Raj.

Table of contents

EXECUTIVE SUMMARY **4**

PROJECT OVERVIEW **4**

FOLDER STRUCTURE **5**

FILE DESCRIPTION **6**

HOW TO USE **7**

- On macOS and Linux: 8
- 4. Install Dependencies 8
- 6. Create the Database 8
- 7. Run the Application 8
- 8. Interact with the Application 8

CONCLUSION **9**

EXECUTIVE SUMMARY

I am excited to present this Flask-based web application designed to streamline note management, providing users with a user-friendly interface for creating, viewing, editing, and deleting notes.

Key Features:

Create Notes: Users can effortlessly create new notes by providing a title and content.

View Notes: The application presents a clear and organized list of existing notes, allowing users to quickly access their information.

Edit Notes: Users can easily update the content of their notes, ensuring that their information remains up-to-date.

Delete Notes: Our application offers a straightforward way to remove unwanted notes, keeping users' note collections tidy.

The application's backend is powered by the Flask framework, and it uses SQLAlchemy to interact with a database (e.g., SQLite, MySQL) for data storage. This ensures data durability and persistence.

PROJECT OVERVIEW

The Flask Note Management Application is a web-based tool designed to simplify and enhance the way users manage their notes and textual information. This project aims to provide a user-friendly and efficient solution for creating, viewing, editing, and deleting notes. Leveraging the Flask web framework, SQLAlchemy for database interactions, and modern web technologies, this application offers a streamlined note management experience.

FOLDER STRUCTURE

```
my-flask-app/
|
├── app/
| ├──__init__.py
| ├── models.py
| ├── routes.py
| ├── static/
| | ├── css/
| | | ├── style.css
| |
| ├── templates/
| | ├── base.html
| | ├── create_note.html
| | ├── edit_note.html
| | ├── delete_confirm.html
| | └── view_notes.html
|
|
├── README.md
├── requirements.txt
├── run.py
```

└── database_create.py (Should be run at first instance only to create database,if database is not created)

FILE DESCRIPTION

my-flask-app/: The root directory of Flask application.

app/: The core directory containing the application code.

__init_.py: Initializes the Flask application and any necessary extensions. This file is typically empty or used for initializing extensions.

models.py: The models.py file contains the definition of the database model for the Flask application. It uses SQLAlchemy, an Object Relational Mapping (ORM) library, to define how data will be structured and stored in the database.

routes.py: Defines the routes and view functions for the Flask application.It is where I have specified the behaviour of each route, such as displaying notes, creating notes, editing notes, and deleting notes.

static/: This directory stores static assets like CSS, JavaScript, and images.

css/: Contains CSS stylesheets for styling your web pages. For example, style.css is a stylesheet that defines the visual appearance of your application.

templates/: Contains HTML templates used by Flask to render web pages.

base.html: A base template that provides the common structure and layout for the application's pages. Other templates (e.g., create_note.html, edit_note.html, delete_confirm.html, view_notes.html) inherit from this base template and insert specific content.

README.md: A documentation file that provides information about the project, including its purpose, setup instructions, and usage. Users and contributors can refer to this file to understand the application.

requirements.txt: A text file listing the Python packages and their versions required for the application.

run.py: A script that can be used to run the Flask application

database_create.py: A script that is used to create the database for the application. This script should be run only once to initialise the database structure, tables, and schemas. After running it, the database (e.g., SQLite) should be set up and ready for use with theFlask application.

This directory structure is well-organised and adheres to best practices for Flask applications. Users can navigate through the app/ directory to understand the core functionality of the application, and the README.md file provides essential documentation to guide users and contributors.

HOW TO USE

If you've downloaded the ZIP file containing the project source code, follow these steps to set up and run the "Burger Center" website on your local machine:

1. **Clone or Download:** If you haven't already, unzip the downloaded ZIP file and place the project folder in a directory of your choice.

To Clone the repository:

```
git clone https://github.com/Bandtox/back-end-Procto.git
```

2. **Open the Project:** Using a code editor of your choice (e.g., Visual Studio Code), open the project folder.

3. Create a Virtual Environment (Optional but

Recommended) On Windows:

```
python -m venv venv

venv\Scripts\activate
```

On macOS and Linux:

```
python -m venv venv

source

venv/bin/activate
```

4. Install Dependencies

```
pip install -r requirements.txt
```

6. Create the Database

```
python database_create.py
```

NOTE: RUN THIS ONLY FOR FIRST INSTANCE.

7. Run the Application

```
python run.py
```

Flask application should start, and you'll see output indicating that it's running. By default, it will be accessible at http://localhost:5000 in your web browser.

8. Interact with the Application

You can access the following routes in your web browser or use navigation buttons.

/: This is the home page where you can view existing notes.

/create: Use this route to create a new note.

/edit/<int:id>: Edit an existing note by replacing <int:id> with the actual note ID.

/delete/<int:id>: Delete an existing note by replacing <int:id> with the actual note ID.

Make sure to use the appropriate HTTP methods (GET, POST) for these routes based on their intended actions.

Github Link: https://github.com/Bandtox/back-end-Procto.git

CONCLUSION

In conclusion, the Flask Note Management Application is a powerful and user-friendly tool designed to simplify the organization and management of textual information. Throughout the development process, we focused on providing a streamlined and efficient solution for creating, viewing, editing, and deleting notes.