**DOCUMENTATION OF THE PROJECT**

In this project, I combined HTML, CSS, and JavaScript to create an interactive web interface that serves as a dashboard. I leveraged Flask as my back-end framework to handle HTTP requests and facilitate the communication between my front-end and the APIs that feed data into my charts.

First off, the project begins with setting up a controlled environment using Conda, which ensures that all dependencies and packages are managed efficiently.

Once the environment is activated, the project utilizes `pip` to install necessary Python packages from a `requirements.txt` file. This step is crucial for installing all the backend dependencies needed for the Flask framework and any other libraries your application depends on.

For the backend Development, I used Flask.It’s a lightweight and flexible web application framework for Python. Flask serves as the backbone, handling HTTP requests and routing them appropriately. It allows for the development of the web application's backend logic, including data processing, API calls, and server-side rendering when necessary.

Since the project involves a database to store and manage data, the users have to configure a `.env` file with database credentials, that way it facilitates a secure connection to the database, enabling the application to query, update, and manage data efficiently. This setup supports the dynamic nature of the dashboard, allowing for real-time data manipulation and retrieval.

Moving on, the frontend of my project is developed using HTML, CSS, and JavaScript. It created a visually appealing and interactive dashboard. HTML provides the structure, CSS adds styling, and JavaScript introduces interactivity, making the dashboard not just informative but also engaging for users.

Before running the application, users are instructed to configure their database credentials, highlighting the importance of data in your project. The application is then launched, bringing together all the components: the Flask backend serves the application logic and interacts with the database, while the frontend provides a rich user interface.

The steps are all included in the “Read me”! (site>Readme.md)