

Title: Development and Deployment of a Web Flask Application for Powerlifting Data

Introduction:

The project aimed to develop a web application using Flask, a Python web framework, to showcase and analyse Gym data. The application provides users information about Gym Data, individual lifters, and their performances. This report outlines the steps in the development process, the database utilized, and instructions for deploying the server and rendering the app.

Development Steps:

1. Database Design:

The project utilized SQLite as the relational database management system. Two tables, `powerlift` and `meets_data`, were designed to store powerlifting data and meet information, respectively. The tables were structured to accommodate various data types, including integers, text, and real numbers.

2. CSV Data Processing:

The Python script processed the CSV files, cleaned the data, and inserted it into the SQLite database. The script used error handling to skip duplicate entries and handle data conversion issues.

3. Web Application with Flask:

The web application was built using Flask, a lightweight and modular web framework. The application had three routes: a home route, a route to display powerlifting data, and a route to display meet information. Each route queried the SQLite database to fetch and display relevant data.

4. HTML Templates:

HTML templates were created to render the different views of the application. Templates were structured to display powerlifting data, meet information, and a home page with navigation links.

5. Unit Testing:

A set of unit tests was developed using the unittest framework to ensure the correct functioning of the routes. These tests covered the home route, powerlift route, and meets route, validating the HTTP status codes and expected content.

6. Deployment:

The application was deployed locally for testing and development purposes. To deploy the server, the Flask development server was utilized, allowing for rapid testing and debugging. Additionally, a Flask configuration was created for development, enhancing the server's performance and error handling.

Instructions for Deployment:

1. Set Up Environment:

- Install the required dependencies using `pip install -r requirements.txt`.
- Ensure Python 3.7 is installed
- Ensure Flask and SQLite are installed.

2. Run Database Initialization Script:

- Execute the `database_create.py` script to create the SQLite database Called gym.db.

3. Run the Flask Application:

- Execute the command `python app.py` to run the Flask application.
- Access the web application in a browser at ` modestohello-radaredward-5000.codio-box.uk `.

Conclusion:

The project successfully developed a Flask web application to display and analyse Gym data. The use of SQLite facilitated efficient data storage and retrieval, and the application was deployed locally for testing purposes. Further improvements and enhancements could include additional features, user authentication, and deployment to a production server for public access.