fsd-demo-HLD

December 30, 2024

0.1 High-Level Design (HLD) for full-stack demo Flask and SQLite application(fsd-demo):

---

1 High-Level Design (HLD)

1.1 1. Introduction

1.1.1 1.1 Purpose

The purpose of this document is to provide a high-level overview of the architecture and design of a simple full-stack application based on Flask and SQLite. This application will manage user data, including creating, reading, updating, and deleting user information.

1.2 2. System Architecture

1.2.1 2.1 Overview

The system is a web-based application that follows the Model-View-Controller (MVC) architecture. It consists of the following components: - **Client-Side**: The user interface, which is accessed via a web browser. - **Server-Side**: The Flask application that handles HTTP requests and interacts with the database. - **Database**: SQLite database that stores user information.

1.2.2 2.2 Components

2.2.1 Client-Side

- **HTML Templates**: Used to render the user interface. Templates are rendered by Flask and sent to the client's browser.
- **Forms**: Used for user input, such as adding, updating, and deleting users.

2.2.2 Server-Side

- **Flask Application**: The main application that handles routing, request processing, and rendering templates.
  - **Routes**:
    - `/`: Home page that displays the list of users.
    - `/update`: Endpoint to update user information.
    - `/delete_user`: Endpoint to delete a user.

- **Controllers**: Functions within the Flask application that handle specific routes and interact with the database.
  - `home()`: Handles the home page and displays the list of users.
  - `update()`: Handles updating user information.
  - `delete_user_route()`: Handles deleting a user.

2.2.3 Database

- **SQLite Database**: Stores user information in a table named `users`.
  - **Tables**:
    - `users`: Contains columns for `id`, `name`, `email`, and `age`.

1.3 3. Data Flow

1.3.1 3.1 User Creation

1. The admin accesses the home page.
2. The admin fills out the form to add a new user.
3. The form data is sent to the server.
4. The server processes the data and inserts a new user into the database.
5. The server retrieves the updated list of users and renders the home page with the new user included.

1.3.2 3.2 User Retrieval

1. The admin accesses the home page.
2. The server retrieves the list of users from the database.
3. The server renders the home page with the list of users.

1.3.3 3.3 User Update

1. The admin fills out the form to update a user.
2. The form data is sent to the server.
3. The server processes the data and updates the user information in the database.
4. The server retrieves the updated list of users and renders the home page with the updated user information.

1.3.4 3.4 User Deletion

1. The admin clicks the delete button next to a user.
2. The user ID is sent to the server.
3. The server processes the request and deletes the user from the database.
4. The server retrieves the updated list of users and renders the home page without the deleted user.

1.4 4. Technology Stack

1.4.1 4.1 Frontend

- HTML
- CSS (optional for styling)

- JavaScript (optional for dynamic behavior)

1.4.2 4.2 Backend

- Python
- Flask

1.4.3 4.3 Database

- SQLite

1.5 5. Security Considerations

- Input validation to prevent SQL injection attacks.
- Proper error handling to ensure the application remains stable.

This HLD provides a high-level overview of the architecture and design of the application.

[ ]: