Readme.md

December 31, 2024

1 User Management Flask Application

This is a very elementary example of a Full Stack Development demo - fsd-demo This simple Flask application is for managing users. The application allows you to add, update, and delete users from a SQLite database.

1.1 Prerequisites

Before you begin, ensure you have the following installed: - Python 3.x - pip (Python package installer)

1.2 Installation

1. **Clone the Repository**: `bash    git clone https://github.com/svhari/fsd-demo.git cd your-repo`
2. **Create a Virtual Environment**: `bash    python3 -m venv venv`
3. **Activate the Virtual Environment**:
   - On Windows: `bash    venv\Scripts\activate`
   - On Unix or MacOS: `bash    source venv/bin/activate`
4. **Install Dependencies**: `bash    pip install Flask`

1.3 Project Structure

```
your_project/

   app.py
   models.py
   templates/
     index.html
   static/
   README.md
```

- **app.py**: Main application file.
- **models.py**: Contains the database interaction functions.
- **templates/**: Contains HTML templates.
- **static/**: Contains static files like CSS, JS, images (optional).
- **README.md**: This file.

1.4 Creating the Files

1. **app.py**:

```
from flask import Flask, request, render_template, jsonify
from models import delete_user, get_all_users

app = Flask(__name__)

@app.route('/')
def home():
    users = get_all_users()
    return render_template('index.html', users=users)

@app.route('/delete_user', methods=['POST'])
def delete_user_route():
    user_id = request.form['user_id']
    delete_user(user_id)
    return jsonify({'message': 'User deleted successfully!'})

if __name__ == '__main__':
    app.run(debug=True)
```

2. **models.py**:

```
import sqlite3

def get_all_users():
    conn = sqlite3.connect('example.db')
    c = conn.cursor()
    c.execute("SELECT * FROM users")
    users = c.fetchall()
    conn.close()
    return users

def delete_user(user_id):
    conn = sqlite3.connect('example.db')
    c = conn.cursor()
    c.execute("DELETE FROM users WHERE id = ?", (user_id,))
    conn.commit()
    conn.close()
```

3. **templates/index.html**:

```
<!DOCTYPE html>
<html>
<head>
    <title>Users</title>
    <style>
        .user-container {
            display: flex;
            align-items: center;
        }
        .user-info {
            margin-right: 10px;
        }
        .user-actions {
            display: inline;
        }
    </style>
</head>
<body>
    <h1>Users</h1>
    <ul>
        {% for user in users %}
        <li class="user-container">
            <span class="user-info">{{ user[1] }} - {{ user[2] }} - {{ user[3] }}</span>
            <form action="/update" method="POST" class="user-actions">
                <input type="hidden" name="user_id" value="{{ user[0] }}">
                <input type="text" name="name" value="{{ user[1] }}">
                <input type="text" name="email" value="{{ user[2] }}">
                <input type="text" name="age" value="{{ user[3] }}">
                <input type="submit" value="Update">
            </form>
            <form action="/delete_user" method="POST" class="user-actions">
                <input type="hidden" name="user_id" value="{{ user[0] }}">
                <input type="submit" value="Delete">
            </form>
        </li>
        {% endfor %}
    </ul>
</body>
</html>
```

1.5 Running the Application

1. **Initialize the Database**: Ensure that you have a SQLite database file named `example.db` with a `users` table. You can create this using the following commands in a Python shell:

   ```
   python
   import sqlite3

   conn = sqlite3.connect('example.db')
   c = conn.cursor()
   c.execute('''
       CREATE TABLE IF NOT EXISTS users (
           id INTEGER PRIMARY KEY,
           name TEXT NOT NULL,
   ```

```
        email TEXT NOT NULL,
        age INTEGER NOT NULL
    )
''')
conn.commit()
conn.close()
```

2. **Run the Application**: `bash`   `python app.py`
3. **Access the Application**: Open your web browser and navigate to `http://127.0.0.1:5000/`.

1.6 License

This project is licensed under the MIT License - see the `LICENSE` file for details.