Flask_activivty 2

💻 Flask Activity 2: Simple Student Record Manager (CRUD App)

🎯 Objective:

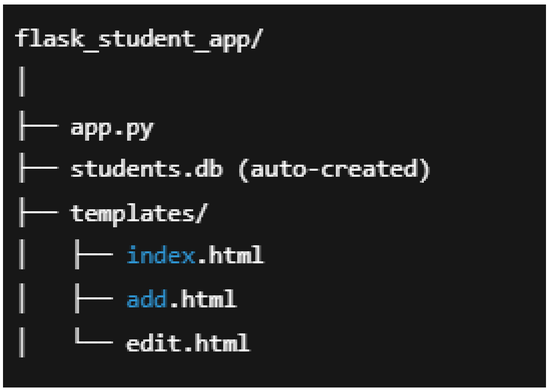
Learn how to:

- Connect Flask to a database (SQLite)
- Create, Read, Update, and Delete student records
- Display records in an HTML table

📝 Instructions:

⚙️ Step 1: Setup Project Folder

Structure:



```
flask_student_app/
│
├── app.py
├── students.db (auto-created)
├── templates/
│   ├── index.html
│   ├── add.html
│   └── edit.html
```

🛠️ Step 2: Code the App

app.py

Code block

```
from flask import Flask, render_template, request, redirect, url_for
import sqlite3

app = Flask(__name__)

# Create DB & Table (runs once)
def init_db():
    with sqlite3.connect('students.db') as conn:
        conn.execute('CREATE TABLE IF NOT EXISTS students (id INTEGER PRIMARY KEY, name TEXT, course TEXT)')

@app.route('/')
def index():
    conn = sqlite3.connect('students.db')
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM students')
    students = cursor.fetchall()
    return render_template('index.html', students=students)

@app.route('/add', methods=['GET', 'POST'])
def add_student():
    if request.method == 'POST':
        name = request.form['name']
        course = request.form['course']
        with sqlite3.connect('students.db') as conn:
            conn.execute('INSERT INTO students (name, course) VALUES (?, ?)', (name, course))
        return redirect(url_for('index'))
    return render_template('add.html')

@app.route('/edit/<int:id>', methods=['GET', 'POST'])
def edit_student(id):
    conn = sqlite3.connect('students.db')
    cursor = conn.cursor()
    if request.method == 'POST':
        name = request.form['name']
        course = request.form['course']
        cursor.execute('UPDATE students SET name=?, course=? WHERE id=?', (name, course, id))
        conn.commit()
        return redirect(url_for('index'))
    cursor.execute('SELECT * FROM students WHERE id=?', (id,))
    student = cursor.fetchone()
    return render_template('edit.html', student=student)

@app.route('/delete/<int:id>')
def delete_student(id):
    with sqlite3.connect('students.db') as conn:
        conn.execute('DELETE FROM students WHERE id=?', (id,))
    return redirect(url_for('index'))

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

🖼️ Templates

index.html

Code block

```
<h2>Student Records</h2>
<a href="/add">Add New Student</a>
<table border="1">
<tr><th>ID</th><th>Name</th><th>Course</th><th>Actions</th></tr>
{% for s in students %}
<tr>
<td>{{ s[0] }}</td>
<td>{{ s[1] }}</td>
<td>{{ s[2] }}</td>
<td>
    <a href="/edit/{{ s[0] }}">Edit</a> |
    <a href="/delete/{{ s[0] }}">Delete</a>
</td>
</tr>
{% endfor %}
</table>

```

add.html

Code block

```
<h2>Add Student</h2>
<form method="POST">
    Name: <input name="name"><br>
    Course: <input name="course"><br>
    <button type="submit">Add</button>
</form>
```

edit.html

Code block

```
<h2>Edit Student</h2>
<form method="POST">
    Name: <input name="name" value="{{ student[1] }}"><br>
    Course: <input name="course" value="{{ student[2] }}"><br>
```

```html
    <button type="submit">Update</button>
</form>

```

✅ Deliverables:

- A working CRUD app using Flask + SQLite
- Students must submit:
  - `app.py`
  - 3 HTML files
  - Screenshot of working app (with added records)

🧠 Optional Challenge:

- Add a **search bar** to filter students
- Add a **grade field**
- Display total number of students