Ex 08

1625

attach all codes (.py, html, css, …)

Generated by ChatGPT 4o

Scenario : Creating a Student Portal

You will create a basic student portal where students can view their profile, see a list of courses, and view details about each course. Teachers can add new courses and manage student enrollments.

1. Setting Up the Project: [50]

- Initialize a new Flask project.
- Create the basic project structure with templates, static files, and app configuration (folders, ...).

2. Creating the Home Page: [100]

- Create a route for the home page (`/`) that renders an HTML template (`index.html`). (50)
- Add a navigation bar with links to "Home," "Profile," and "Courses." (50)

3. Profile Page: [150]

- Create a route for the profile page (`/profile`) that renders a profile template (`profile.html`). (50)
- Pass a dictionary of student details (e.g., name, email, major) to the template and display it using Jinja. (100)

4. Courses Page: [250]

- Create a route for the courses page (`/courses`) that renders a template (`courses.html`). (50)
- Create a list of courses (each with a name, description, and teacher) and pass it to the template.(100)
- Display the list of courses in a table format. (100)

5. Course Details Page: [250]

- Create a dynamic route for course details (`/courses/<course_id>`) that renders a template (`course_detail.html`). (100)
- Display the details of the selected course based on the `course_id`. (50)
- Use URL parameters to pass the course ID and retrieve the relevant course data.(100)

6. Static Files:[150]

- Add CSS to style the pages. (50)
- Create a `static` folder and include a CSS file (e.g., `styles.css`). (50)
- Link the CSS file in your templates to apply the styles. (50)

7. Built-in Filters: (3*25 = 75)

- Use built-in filters like `upper`, `lower`, and `capitalize` to format text on the profile page.

8. Custom Filters: (150)

- Create a custom filter to format the course descriptions (e.g., limit the number of words displayed).(100)
- Register the custom filter with the Flask app and use it in the `courses.html` template.(50)

9. Forms and Data Handling: (200)

- Create a form on the profile page to update student details. (100)
- Handle form submission with a POST route and update the student details. (100)

10. Adding Courses (Teacher Role): (250)

- Create a form for adding new courses (`/add_course`) and handle the form submission. (100)
- Add validation to ensure all fields are filled out correctly. (100)
- Update the courses list to include the newly added course. (50)