Project Title

Simple Record Management System using Flask

Project Overview

Take the lead by proposing a custom project of your choice. You will build your application from the ground up using a modern stack: Python/Flask for logic and HTML/CSS for design. It allows users to create, view, update, and delete information (CRUD) that is permanently stored in a database (text file in JSON format). The application uses Flask to handle the backend logic, while HTML and CSS are used to design a user-friendly interface. The stored data remains saved even after the program is closed, and the system can display organized information based on the saved records. Input validation is part of the project. The project includes input validation to ensure that all data entered by users is correct, complete, and follows the required format.

Expected Output

By the end of the project, students will have a working Flask web application that can manage records, store data permanently, and generate simple reports.

NOTE: The features that must be fully functional in the 3rd Quarter include user registration and login, the ability to add and search entries, and an aesthetic design that is relevant to the chosen topic. The remaining features will be implemented during the 4th Quarter.

A list of suggested JSON-Based Project Ideas is provided below.

1. Student Score Tracker

- Store student names, subjects, and grades in a JSON file
- Perform CRUD on database records (e.g Flask)
- Display average, highest, and lowest grades

2. Daily Mood Tracker

- Record date, mood level, and notes in a JSON file
- View mood trends using descriptive statistics

3. Book or Movie Log

- Add book/movie details to a JSON file
- Show favorite genres, most watched/read items, etc.

Other possible Projects:

1. Student Performance Tracker

- **Description**: A web app where users can input students' names, subjects, and grades.
- **Features**:
  - Add/Edit/Delete student records
  - View average, highest, and lowest scores
  - Filter by subject or student
- **Metrics**: Mean, max, min, count of grades per subject or student

2. Daily Expense Manager

- **Description**: A simple budgeting tool that lets users record and monitor expenses.
- **Features**:
  - Add/Edit/Delete daily expenses with category (e.g., food, transport)
  - View expenses by date or category
- **Metrics**: Total and average spending per category; highest spending day

3. Book Library Catalog

- **Description**: A digital catalog of books owned by the student or class.
- **Features**:
  - Add/Edit/Delete books with title, author, genre, and rating
  - Search/filter books by author or genre
- **Metrics**: Average rating per genre, number of books per author, total books

4. Covid-19 Case Tracker

- **Description**: A web-based system to monitor COVID-19 data (based on a sample/local dataset).
- **Features**:
  - Add/Edit/Delete cases with location, date, and number of cases
  - Filter by date or location
- **Metrics**: Total cases, average daily cases, peak day

5. Attendance Monitoring System

- **Description**: A tracker for students' attendance over a period.
- **Features**:
  - Add/Edit/Delete attendance records per day
  - Filter by student or date
- **Metrics**: Attendance rate per student, the number of present/absent students

6. Personal Fitness Log

- **Description**: Users can log their daily workouts, calories burned, or step count.
- **Features**:
  - Add/Edit/Delete workout entries (e.g., activity, duration, calories)
  - View logs by date or type
- **Metrics**: Total and average calories burned, most active day

7. Simple Inventory Management System

- **Description**: Manage classroom or small business inventory items.
- **Features**:
  - Add/Edit/Delete items with quantity, price, and category
  - Track items that are low in stock
- **Metrics**: Total value of inventory, average stock per item, most stocked category

8. Movie Ratings Database

- **Description**: Users can add movies they've watched and rate them.
- **Features**:
  - Add/Edit/Delete movie entries (title, genre, rating, review)
  - Filter by genre or rating
- **Metrics**: Average rating per genre, most-watched genre, top-rated movie

RUBRICS

Criteria	Excellent (8 points)	Good (7-6 points)	Fair (5-4 points)	Needs Improvement (3–0 points)
1. Project Concept & Proposal	Project idea is clear, appropriate, and well explained	Project idea is clear with minor gaps	Basic idea but lacks explanation	Unclear or incomplete idea
2. Flask Backend Functionality	Flask routes work correctly with no errors	Minor errors but mostly functional	Partially working	Not working or missing
3. User Registration & Login	Registration and login work correctly	Works with minor issues	Partially working	Not working
4. Add Entries (Create)	Users can add records correctly	Mostly works	Partially works	Not working
5. Search Entries (Read)	Search feature works correctly	Mostly works	Partially works	Not working
6. Input Validation	Invalid input are filtered	Mostly works	Partially works	Not working
7. HTML & CSS Design	Design is clean, attractive, and topic-appropriate	Design is neat with minor issues	Design is basic	Poor or unfinished design