Assignment - 2

Project: Sunbeam Online Student Portal

Implementation Using Python Flask + MySQL Connector (Postman Based)

Given Database Schema

The database `institute_management_db` contains the following tables:

- `courses`
- `students`

The database is created using the provided SQL schema.

Objective

To perform **CRUD operations (Create, Retrieve, Update, Delete)** on the `courses` and `students` tables using **Python Flask** and **mysql-connector**, and to **send and receive data using the Postman application**.

Q1. Database Connectivity

Using **Python Flask** and **mysql-connector**, establish a connection with the MySQL database `institute_management_db`.

- Install and configure the required MySQL connector.
- Create a reusable database connection function.
- Verify the database connection by running the Flask server from the terminal.

Q2. Retrieve Data from Tables (GET APIs)

Using Flask and mysql-connector, design and implement **GET APIs** to retrieve data from the database and display it in Postman.

- Retrieve all records from the `courses` table.
- Retrieve all records from the `students` table.

Q3. Insert Data into Tables (POST APIs)

Using Flask and mysql-connector, design and implement **POST APIs** to insert data into the database tables.

- Insert a new record into the `courses` table by sending JSON data through Postman.
- Insert a new record into the `students` table by sending JSON data through Postman.

Q4. Update Table Records (PUT APIs)

Using Flask and mysql-connector, design and implement **PUT APIs** to update existing records in the database.

- Update the fees of a course in the `courses` table.
- Update the mobile number of a student in the `students` table.

Q5. Delete Records from Tables (DELETE APIs)

Using Flask and mysql-connector, design and implement **DELETE APIs** to remove records from the database.

- Delete a student record from the `students` table.
- Delete a course record from the `courses` table.

API Testing Using Postman

- Start the Flask application using the terminal.
- Use Postman to test all GET, POST, PUT, and DELETE APIs.
- Pass request data in JSON format through Postman.
- Verify responses and database changes after each operation.

Execution Instructions

- Run the Flask application from the terminal.
- Perform all database operations using Postman.
- Observe and verify JSON responses in Postman.
- Confirm database changes directly in MySQL.
