Lab Activity 10

Topic - Flask,SQLAlchemy

For this activity you need to demonstrate CRUD operations using **SQLAlchemy** in a simple flask application.

Create two files:
App.py, client.py

- Write your flask application in app.py.
- Create a users.db database and students table using SQLAlchemy containing fields: id,name,stream.
- Keep id as the primary key with Integer type (set autoincrement=False).
- Name and stream should be of String type with maximum 80 char length.

App.py: (30 Marks)

Endpoints:
To read the POST data use `data = request.get_json()`

/create (10 Marks)

It should add a new student entry into the database with the given id,name and stream (taken from the POST request).

If the id already exists, return “id already exists.”
Else return “added successfully.”
(The details for making a POST request are given later.)

/read (5 Marks)

It should fetch all the students entry and must return a json response in the following format
{
“1”:{“name”:”nisarg”, “stream”:”cse”},
“2”:{“name”:”arvind”,“stream”:”cse”},
“3”:{“name”:”ashutosh”,“stream”:”cse”}
}
(Here 1,2,3 are the ids)

/update (10 Marks)

It should take id,name,stream as the parameters (from PUT request) and update the name and stream of the corresponding id.

If the id is not present, return “id not found”.
Else return “Update success” on updating the record successfully.

/delete (5 Marks)

It should take id as the parameter((from DELETE request)). Delete the record corresponding to the id. If the id is not present, return “id not found”.
Else return “Delete success” on deleting the record successfully.

Start the flask app on **8000 port.**

Client.py (20 Marks, 5 for each)
For making get, post, update, delete requests, use the **requests** library in python.
Create the following menu in an infinite loop:

1. Add student
2. Fetch all students
3. Update student
4. Delete student
5. Exit

If the user selects 1, take student id,name,stream as input from the user. Create a json object as data={“id”:”123”,“name”:”abc”,”stream”:”xyz”}.
Make the POST request , print the response:

```
response =
requests.post('http://localhost:8000/create',json=data).content
```

If the user selects 2, simply print the response json object.
GET request:

```
response = requests.get('http://localhost:8000/read').content
```

If the user selects 3, take student id,name,stream as input from the user. Create a json object as data={“id”:”123”,“name”:”abc”,”stream”:”xyz”}.
Make the PUT request , print the response:

```
response =
requests.put('http://localhost:8000/update',json=data).content
```

If the user selects 4 take student id from the user.
Make the DELETE request , print the response:

```
response =
requests.delete('http://localhost:8000/delete/{}'.format(id)).conten
t
```

If the user selects 5, exit the client.

Directory Structure:
Create a RollNumber folder.
Add app.py, client.py inside it, Readme is optional.
Zip the folder as RollNumber.zip