

Python Flask Lab

Due Date: 8th November 2023, 05:00 pm

Remember Lab 3?! You helped creating, editing and managing the menu of **Hotel Shaandar**. The owners are delighted with your previous work and have hired you again to develop a web application for them.

They have some specific requirements:

The application should have three stakeholders. Admin, waiting staff and customer. They can perform the following tasks:

1. Admin Staff

- Admin can view the Menu List. (5 marks)
- Admin can add new item to the Menu List. (10 marks)
- Admin can update item price from the Menu List. (15 marks)

2. Waiting Staff

- Waiting staff can view all orders. (10 marks)
- Wait staff can see order and current bill amount for a customer. (10 marks)

3. Customers

- Customer can view his current order. (10 marks)
- Customer can add items to their order. (15 marks)
- Customer can remove items from their order. (15 marks)
- Render Customer bill with items, their prices and total bill amount using Jinja Template. (**bill.html** is already provided in the Lab Activity folder). (10 marks)

Here are the CREATE scripts for tables **Menu** & **CustomerOrder**

```
CREATE TABLE IF NOT EXISTS Menu (  
    item_id INT NOT NULL,  
    item_name VARCHAR(80) NOT NULL,  
    item_price INT NOT NULL,  
    PRIMARY KEY (item_id)  
);
```

```
CREATE TABLE IF NOT EXISTS CustomerOrder (  
    customer_id INT NOT NULL,  
    items JSON, -- List of item ids  
    PRIMARY KEY (customer_id)  
)
```



Do not change any column name(s) or table name(s). Use these exact queries.

Feel free to insert dummy data and test your code.

We have also supplied you with the API specifications (***api_contract.pdf***), which will provide you with a clear understanding of how to implement a particular API.

It seems like a challenging task, especially to complete within 3 hours, doesn't it? 😬

Don't worry, we have you covered! 🤖 Your role is to **just** establish database connections and write the APIs.

Please Note:

- Run your flask application on **127.0.0.1:5000**.
- Follow the API Specifications document very carefully. The URIs, headers, parameters and HTTP methods specified should be strictly adhered to.
- We have added a Postman collection (***lab11_collection.postman_collection***) in the Lab Activity Folder to help you with testing of your APIs. Importing and using a Postman collection is already discussed in tutorials.
- Use of Object Relational Mapping (**Flask-SQLAlchemy**) is recommended (not compulsory).
- You can also refer to the sample code provided during tutorials. ([Link](#))
- Your code is expected to work on our machines just by changing the DB connection string. **Please adhere to all the tips and things mentioned.**

Instructions:

- Deadline mentioned during the Lab is strictly immutable. No extensions will be given.

- Any naming convention mentioned in the lab activity must be followed strictly or marks may be deducted for the same.
- Any plagiarized content will fetch zero marks for the current lab and will be followed by strict action against the students involved. However, discussion of ideas is allowed.
- You are allowed to use ChatGPT and other AI tools for assistance.

Submission Criteria:

```
2022201026.zip
|____2022201026
      |____2022201026_main.py
```