**SOFTWARE SYSTEM DEVELOPMENT – Monsoon 2021**

**Assignment 3B – Flask and SQLAlchemy**

**Submission Due Date: 21 November 2021, 5 pm**

**Important Notes:**

- This assignment is an individual submission.
- This is the 2nd part of Assignment 3
- Total Marks of **100**
- All script submissions should be done as per instructions.
- Input/output should fit the criteria mentioned in respective questions.
- Submission must be done via Moodle only. Post your queries in Assignment Queries Channel in MS Teams.
- You can use the following libraries: Flask, Flask-Session, SQLAlchemy. Confirm once in the queries channel before using any other library

This part is in continuation to part A. You can utilize the previously written code for the same and modify it in accordance to the below points.

- The interface would be same as before which is cli.
- Instead of reading the menu from a csv file, this time it should be persisted in a MySQL Database
- Consider 2 types of users. Chef and Customer
- A customer should be able to perform the following activities
    - Signup, Login, Logout **(30)**
    - Read the latest menu stored in DB **(10)**
    - Order food items **(10)**
    - Generate bill (Logic and steps like previous part). Once the bill is generated, this transaction is complete and incoming order requests will contribute to the next transaction **(15)**
    - View previous bill statements. Initially a list of transactions pertaining to that customer will be displayed. When the customer selects one of the transactions, the entire bill for that transaction will be displayed. **(15)**
- A chef should be able to perform all the activities that a customer can with the addition of
    - Adding new items to the menu along with its cost. These changes should be reflected in the DB **(10)**
- Create a REST API for the above-mentioned tasks. For both types of users, a username-password based authentication mechanism must be established. Every customer will have his/her own account and new customers should have the ability to create new ones. These accounts will be stored in the DB.
- You can have one static account for the chef. However, your backend service should be able to differentiate between a customer and a chef without the client explicitly specifying the same.
- Your routes should be protected. Non logged in users should only be able to access **signup** and **login** routes**.** Customers should be **unable** to access the route for modifying the menu.

- Use Document strings to comment your code and use naming conventions as per PEP8 standards. **(10)**

### Submission format:

- There will be a minimum of 2 files. **Client.py** which will contain the CLI for interaction, while **server.py** will contain the API routes
- Create a readme file which will contain instructions on
    - How to execute the code
- Add all the files in a directory named <rollnumber> and zip the contents as **<rollnumber>.zip**