Al-Akeel Food Ordering Platform

Introduction

Al-Akeel Company has developed an online order management platform over the past 15 Years. That platform generally allows users to make online orders to a specific restaurant from a registered restaurant list. It also tracks and delivers such orders from the restaurant's locations to the customer. The platform serves clients from either a dedicated mobile application, web app or assisted from customer service over the phone. The platform allows restaurant staff to manage their restaurants specific orders. The platform integrates with Bank Misr as a payment gateway to full-fill online payments for orders. The platform integrates with Aramex shipment company to manage the delivery of orders. Recently the company started to face a set of issues mainly reliability issues where some times the platform reported to be down. Suffers from slow responses. They also faced a huge challenge when they needed to support mobile payment as well as support Al-Akeel specific shipment fleet to work side by side with the existing Aramex shipment.

Entities:
User: Id ,name, role
Meal: id, name , price, fk_restaurantId
Order: Id, Item array, total_price, fk_runnerId, fk_restaurantId, order_status(preparing, delivered, canceled)
Runner: Id, name, status(available, busy),delivery_fees
Restaurant: Id, name, ownerId, list of meals
Note: add any additional fields if needed

System functionalities:

User can have 3 different roles **Customer, RestaurantOwner, Runner roles**
Restaurant owner, customer, runner

- Sign up and Login for each role
  Note: when creating runner account let user enter delivery fees per order , this value will be reused when calculating the total order value
- **Restaurant owner:**
    - Create restaurant menu
    - Edit restaurant: change menu meals for each restaurant
    - Get restaurant details by id
    - Create restaurant report: given a restaurant id print
      how much the restaurant earns (summation of total amount of all completed orders) , Number of completed orders, Number of canceled orders
- **Customer:**
    - Create order by customer
      Exceptations: imagine a normal restaurant receipt

Order details should contains date ,restaurant name, items list , delivery fees, runner name, total receipt value (summation of items prices , delivery fees)
list orders by customer id

    - When creating an order select a random available runner from db and assign it to an order and convert his status to busy

    - Edit order [change an order's items] make sure an order is not canceled and it is in the preparing state to be edited
    - List all restaurants

- **Runner:**
    - Runner can mark an order is delivered and his status to available
    - Get number of trips completed by a runner make sure orders are not canceled and marked as completed

<u>Project Guidelines (Please read carefully)</u>:

- You can use any relational database,you can use in-memory database (Recommended DB: SQLite)
- You **must use** GIT for source code management and create a **private** repository on GitHub
    - Add your TA to git repo
        - Eng Esraa: e.salem@fci-cu.edu.eg
        - Eng Mohamed: m.samir@fci-cu.edu.eg
        - Eng Mahmoud:m.hadad@fci-cu.edu.eg

- Teams should be of 3 - 4 members from the same group
- Each team member contribution will be calculated from GitHub, Team members with low contribution will get low grades
- Deadline is **22/05/2023 11:59 PM**
- **create a postman collection contains the created APIs to be used in the discussions**

You have to create a folder contains the txt file contains team names and ids, source code and the postman then compress it and be uploaded (SLB_ID1_ID2_ID3_ID4.zip)

- Plagiarism checks will run automatically on your code, Any cheating case, **Both teams will receive negative grades**
- <u>Bonus Tasks:</u>
    - Create a Frontend Application using any Javascript Framework (Angular, React, VueJS) including the GUI of all the requested functionalities **(+2 Marks)**
    -
- **<u>Note</u>**: Please don't implement any of the bonus tasks unless you are delivering the full requested functionalities otherwise you will not get the grade of the bonus task