TIME REMAINING
0:33:59

End Exam

Note: Please **SUBMIT** each question individually before ending the exam to receive score.
Note: This is a monitored test.

Ride-Booking System

Ride-Booking System

A competitor to Uber has hired you to design a ride booking application. Your task is to build a system that incorporates the requirements detailed below.

There can be two types of users; drivers or riders. For drivers, store their vehicle information in addition to personal details. Riders can book rides by providing pickup and drop-off locations. Drivers can accept or reject rides based on their availability. For security purposes, only valid drivers, whose email and phone #, along with vehicle details are recorded in the system should be able to accept the rides. Riders can also cancel the ride if a driver hasn’t accepted the ride yet. Once the ride is cancelled, the total cost charged to the rider should be 0. Riders should be able to view the estimated cost before booking a ride. The system should dynamically calculate ride cost based on the distance provided in KMs, where the cost per KM = 100 Rs. It should also include platform fee for every ride booked, which is a fixed price of 15 Rs. Riders and drivers can rate each other after a ride is completed. You should maintain a record of all ratings, and calculate the average rating for riders and drivers. Riders should be able to view their rides' history.

Tasks:

1. **Design and Implementation:**

- Design appropriate classes/methods and implement functionality keeping object oriented design principles in mind.

2. **Database:**

- After implementing the solution in your preferred programming language, you will need to design the database schema.

- Your schema should define all relevant tables, columns, data types, constraints, and relationships between entities.

- You can add your schema in the form of comments under the same file, just make sure that your code executes without any errors.

- Here’s a format you can use to write schema:

  TABLE1

- COL1 INT PRIMARY_KEY NOT_NULL

- COL2 FOREIGN_KEY

- COL3 VARCHAR

- CONSTRAINTS: COL1 value between 0-5

- RELATIONSHIPS: Child of <PARENT_TABLE>

Guidelines:

- It’s an open-ended question, feel free to include your assumptions in the comments.

- The code MUST be executable. This only includes OOP classes/methods and not the Database part.

- Ensure your code gracefully handles exceptions.

- Provide some test code to demonstrate how your code works.

- Final code should be well-commented and easy to read.

- After implementing the required use cases, you are welcome to extend the logic as you see fit.

Your Response ✓ COMPLETE