Programming Assignment #4

Railway Reservation System

Scenario:

You are hired by a Railway Company, you are asked to design a system which provides information to passengers about the trains available on the bases of source and destination, other information like status of the train, status of passenger ticket, how many seats are available etc. Suppose you are given the following requirements for a simple database for the Railway Reservation System:

This system helps to maintain the records of different trains, the train's status, and passengers. The database consists of 4 tables:

- **Train:** Train Number, Train Name, Premium Fair, General Fair, Source Station, Destination Station
- **Train_Status**: TrainDate, TrainName, PremiumSeatsAvailable, GenSeatsAvailable, PremiumSeatsOccupied, GenSeatsOccupied
- **Passenge**r: first_name, last_name, address, city, county, phone, SSN, bdate
- **Booked**: Passanger_ssn, Train_Number, Ticket_Type, Status

Note: As the system is very large and is not feasible to develop therefore there are some assumptions that need to be considered, for example:

- Only two categories of tickets are available: Premium and General Ticket
- The total number of tickets can be booked in each category (Premium and General) is 10
- Number of tickets in waiting list is 2
- Total Number of trains are 5
- Any stops made by a train before its destination and their bookings are not considered.

Create a GUI for the RRS database SQL Queries:

Railway Reservation Application:

Create a simple and friendly GUI interface that would be able to perform the following tasks. Use Python programming languages to develop a GUI interface and Sqlite3 for the data base. The user will have to type or select the query's input parameters and post the question to your program. The program needs to return all result's rows.

Query/GUI TASK:

1. User input the passenger's last name and first name and retrieve all trains they are booked on.
2. User input the Date and list of passengers travelling on entered day with confirmed tickets displays on UI.
3. User input the age of the passenger (50 to 60) and UI display the train information (Train Number, Train Name, Source and Destination) and passenger information (Name, Address, Category, ticket status) of passengers who are between the ages of 50 to 60.
4. List all the train name along with count of passengers it is carrying.
5. Enter a train name and retrieve all the passengers with confirmed status travelling in that train.
6. User Cancel a ticket (delete a record) and show that passenger in waiting list get ticket confirmed.

Please make sure to write out team contributions at the end of the document! This will be 5 points of the overall grade.

Dataset will be available in a zip folder with the assignment.

Canvas Submission:

Create a folder with your code, named "Code.py" Inside the "Code" folder, create a Readme.docx(pdf) file, this file should have the 6 query solutions listed and step-by-step instructions and screenshots on how to install and run your program from scratch. Also, place a database backup (self-contained file), named as rrs.sql inside your "Code" folder,

Create your submission file.

In the front-page, type a title for your project submission and specify your name, and your team member's name.
On the second page, include your honor code. Failing to do so will cost. [5 points].
Task, screenshots from your GUI for each task, along with some explanation or a meaningful label.

Save your submission file as "Report.docx" or "Report.pdf". Make sure that all commands are editable!
NO HARD CODED QUERIES

Create a folder named as "teammate1lastname_teammate2lastname" and place inside the "Code" folder and your "Report.docx" file.

Zip your folder and submit it on Canvas any time before the deadline.

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

Rubric:

TASK	**POINTS**
GUI	25 Points
Query Running	40 Points
Team Contribution	5 Points
Honor Code	5 Points
Report and Readme file	15 Points
Demo	10 Points
TOTAL	100 Points