DATA STRUCUTRE PROJECT REPORT

Project Group – 3

Title : Implement a Car Rental System

Submitted To: Dinesh Patel Sir

Submitted By:
Gourav Singh Panwar
Abhay Choudhary
Ojasvi Sharma

INTRODUCTION

1. Purpose of the Car Rental Management System –

- The Car Rental Management System is designed to facilitate the process of renting vehicles for both individuals and businesses. Its primary purpose is to streamline the entire rental process, making it more efficient and user-friendly for both administrators and customers. The system allows users to perform various tasks, including browsing available cars, making reservations, managing bookings, and handling driver assignments. By automating these processes, the system reduces the administrative burden on rental agencies and enhances the overall customer experience.

- The system is built to cater to two main user roles: administrators and passengers. Administrators can manage the fleet of cars, oversee driver assignments, and handle user registrations, while passengers can search for cars based on their needs, book vehicles, and view their booking history. This dual functionality ensures that all aspects of car rental management are covered, providing a comprehensive solution for both service providers and customers.

2. Significance in the Real World –

i. **Enhanced Customer Experience –** Enables easy search, comparison, and booking for greater customer satisfaction.

ii. **Operational Efficiency –** Automates fleet and booking management, reducing errors and improving service.

iii. **Scalability –** Adapts seamlessly to business growth and changing operational needs.

3. Objective –

- To develop a user-friendly car rental management system that simplifies car rentals, booking management, and driver management, streamlining the process for administrators and passengers.

4. Technologies Used –

i. **C++ -** Utilized for its efficiency and object-oriented features, enabling modular design.

ii. **File Handling -** Ensures data persistence for users, cars, drivers, and bookings, preventing data loss.

iii. **Data Structures -** Arrays, structures and vectors manage dynamic collections like bookings and available cars.

5. Challenges Faced –

i. Designing a robust file-handling system for seamless data updates.

ii. Balancing performance with flexibility when using arrays and vectors.

iii. Creating an intuitive interface for administrators and users.

iv. Developing efficient driver assignment algorithms.

v. Debugging and handling errors in file operations and data structures.

Concepts of Data Structures Used

- The system employs several data structures to represent different entities and manage data effectively.

➢ Structs –

1. User – Represents a registered user (passenger) in the system.

i. **string username** – The username of the user.
ii. **string password** – The password of the user.

2. Car – Represents a car available for rent.

i. **string carID** – Unique identifier for the car.
ii. **string model** – The model of the car.
iii. **int capacity** – The seating capacity of the car.
iv. **double rentalCharge** – The rental charge per day.
v. **bool isAvailable** – Availability status of the car (true if available, false otherwise).

3. Driver – Represents a driver associated with the car rental service.

i. **string driverID** – Unique identifier for the driver.
ii. **string name** – The name of the driver.
iii. **string contact** – The contact number of the driver.
iv. **bool isAvailable** – Availability status of the driver.

4. Passenger – Represents a passenger who books a car.

i. **string name** – The name of the passenger.
ii. **string contact** – The contact number of the passenger.

5. **Booking –** Represents a booking made by a passenger.

i. **string carID** – The ID of the booked car.
ii. **string passengerName** – The name of the passenger.
iii. **string passengerContact** – The contact number of the passenger.
iv. **double rentalCharge** – The rental charge for the booking.
v. **string bookingDate** – The date of the booking.
vi. **string driverID** – The ID of the assigned driver (if any).

➢ Arrays and Vectors –

1. **Arrays** – The system uses fixed-size arrays to store cars, drivers, and bookings.

i. **Car cars[MAX_CARS]** – Array to hold car objects.
ii. **Driver drivers[MAX_DRIVERS]** – Array to hold driver objects.
iii. **Booking bookings[MAX_BOOKINGS]** – Array to hold booking objects.

2. **Vector** – A dynamic array is used to store registered users.

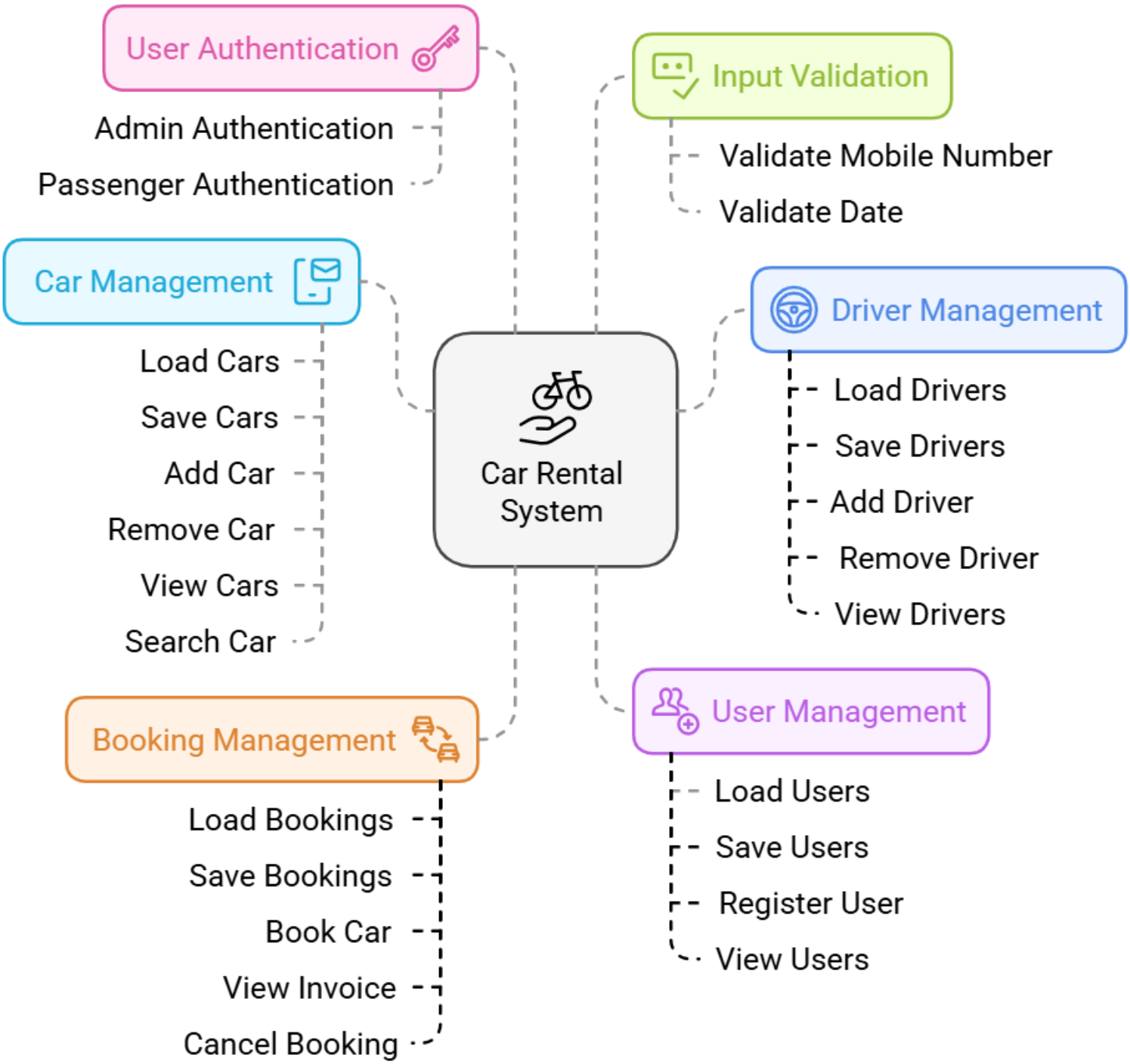
i. **vector<User> users** – A vector that can grow or shrink as users are added or removed.

➢ Global Variables –

1. **int carCount**, **driverCount**, **bookingCount** – Counters to keep track of the number of cars, drivers, and bookings.

2. **string currentPassengerContact** –Holds the contact number of the currently logged-in passenger.

Functionality and Key Points



Functionality and Key Points

1. **User Authentication –**

i. **bool authenticateUser (const string &role)** -

a. Authenticates users based on their role (admin or passenger).

b. For admin, it checks against hardcoded credentials.

c. For passengers, it checks the **users** vector for matching username and password.

2. **User Management –**

i. **void loadUsersFromFile()** -

a. Loads registered users from a text file into the **users** vector.

ii. **void saveUsersToFile()** -

a. Saves the current list of users to a text file.

iii. **void registerUser ()** -

a. Allows new passengers to register by entering a username and password.

b. Checks for username uniqueness before adding to the **users** vector.

iv. **void viewUsers()** -

a. Displays a list of registered users in a tabular format.

3. Car Management –

i. void loadCarsFromFile() –

a. Loads car data from a text file into the **cars** array.

ii. void saveCarsToFile() –

a. Saves the current list of cars to a text file.

iii. void addCar() –

a. Allows admins to add new cars by entering details about the car, including ID, model, capacity, and rental charge. It checks for unique car IDs before adding.

iv. void removeCar() –

a. Allows admins to remove a car by entering its ID. It shifts the remaining cars in the array to fill the gap.

v. void viewCars() –

a. Displays a list of all cars in a formatted table, showing their details such as ID, model, capacity, rental charge, and availability.

vi. void searchCar() –

a. Allows passengers to search for cars based on required capacity and maximum rental charge. It displays matching cars.

4. **Driver Management –**

i. **void loadDriversFromFile()** –

a. Loads driver data from a text file into the **drivers** array.

ii **void saveDriversToFile()** –

a. Saves the current list of drivers to a text file.

iii. **void addDriver()** –

a. Allows admins to add new drivers by entering their ID, name, and contact number. It checks for unique driver IDs.

iv. **void removeDriver()** –

a. Allows admins to remove a driver by entering their ID, similar to the car removal process.

v. **void viewDrivers()** –

a. Displays a list of all drivers in a formatted table, showing their details such as ID, name, contact, and availability.

5. **Booking Management –**

i. **void loadBookingsFromFile()** –

a. Loads booking data from a text file into the **bookings** array.

ii. **void saveBookingsToFile()** –

a. Saves the current list of bookings to a text file.

iii. void bookCar() –

a. Allows passengers to book a car by entering the car ID, passenger details, and booking date. It checks for car availability and can assign a driver if requested.

iv. void viewInvoice() –

a. Displays the invoice for the current passenger's bookings, including car details, rental charge, and driver information if applicable.

v. void cancelBooking() –

a. Allows passengers to cancel a booking by entering the car ID and their contact number. It updates the availability of the car and driver if applicable.

6. Input Validation –

i. bool isValidMobileNumber(const string &number) –

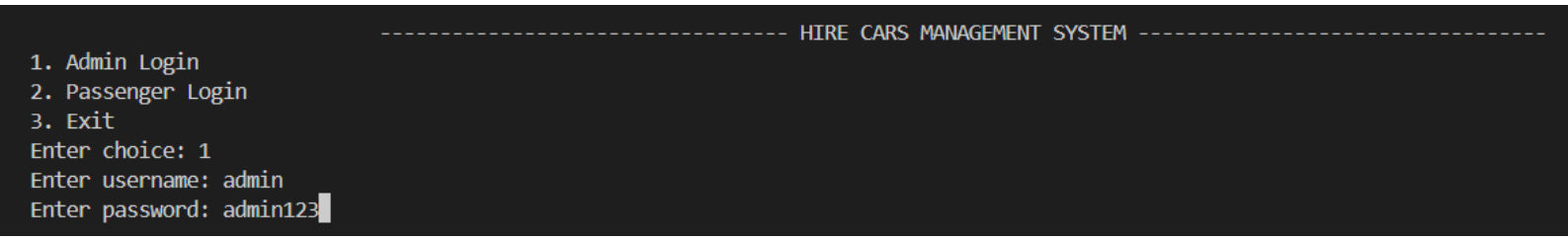
a. Validates that the mobile number is exactly 10 digits long and contains only digits.

ii. bool isValidDate(const string &date) –

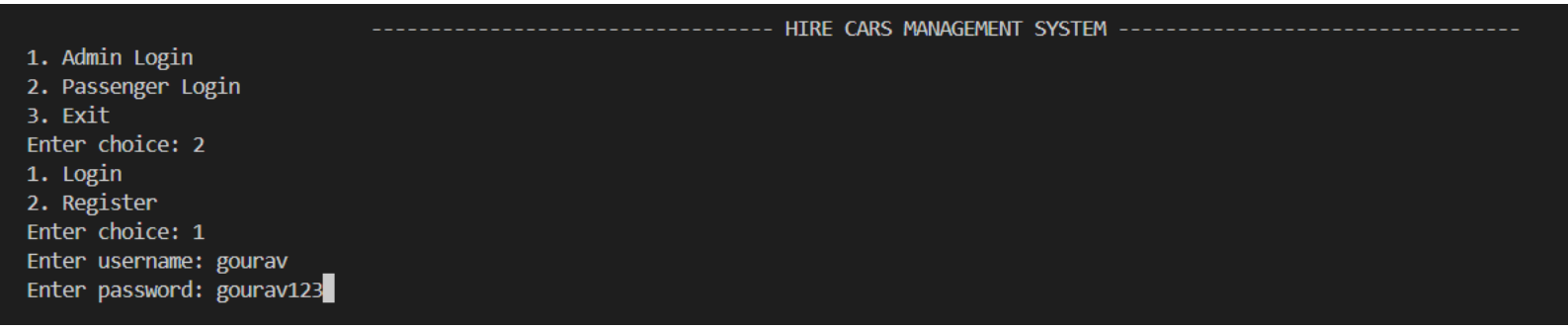
a. Validates the booking date format (DD-MM-YYYY) and checks that the date is not in the past.

SCREENSHOTS

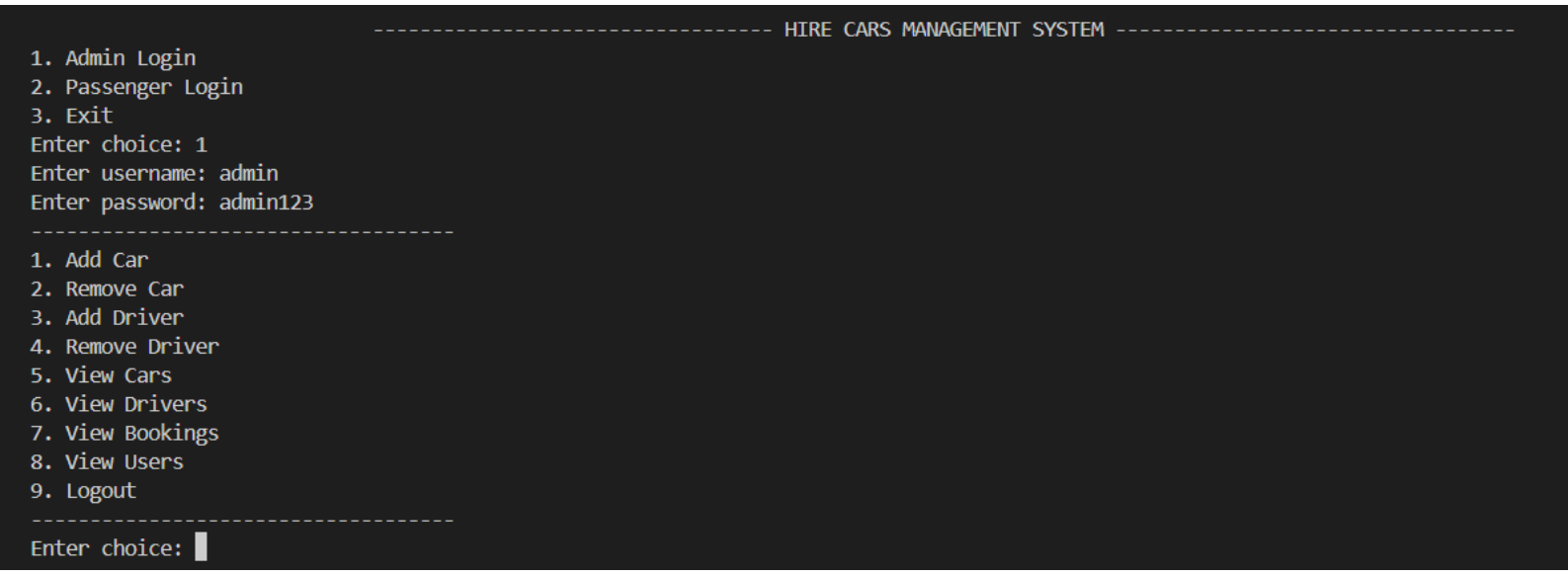
1. Login Section (Admin) -



2. Login Section (Passenger) -



3. Admin Menu / Functionalities -



4. Adding a Car -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 1
Enter Car ID: 01
Enter Car Model: Swift Dezire
Enter Capacity: 4
Enter Rental Charge: 1000/Day
Car added successfully!
-----------------------------------
```

5. Adding a Driver -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 3
Enter Driver ID: 01
Enter Driver Name: Aakash Singh
Enter Driver Contact (10 digits): 9009848332
Driver added successfully!
-----------------------------------
```

6. View Car List -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 5
Car ID        Model              Capacity  Rental Charge  Available
-------------------------------------------------------------------
01            Swift Dezire       4         1000.00        Yes
-----------------------------------
```

7. View Driver List -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 6
Driver ID     Name               Contact        Available
-----------------------------------------------------------
01            Aakash Singh       9009848332     Yes
-----------------------------------
```

8. Removing a Car -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 2
Car ID        Model             Capacity  Rental Charge  Available
-------------------------------------------------------------------
01            Swift Dezire      4         1000.00        Yes
Enter Car ID to remove: 01
Car removed successfully!
-----------------------------------
```

9. Removing a Driver -

```
 -----------------------------------
 1. Add Car
 2. Remove Car
 3. Add Driver
 4. Remove Driver
 5. View Cars
 6. View Drivers
 7. View Bookings
 8. View Users
 9. Logout
 -----------------------------------
 Enter choice: 4
 Driver ID      Name              Contact        Available
 ------------------------------------------------------------
 01             Aakash Singh      9009848332     Yes
 Enter Driver ID to remove: 01
 Driver removed successfully!
 -----------------------------------
```

10. View Registered Users -

```
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 8
Username            Password
-----------------------------------------
gourav              gourav@123
ojasvi              ojasvi@123
abhay               abhay@123
-----------------------------------
```

11. View Bookings -

```
                    ---------------------------------- HIRE CARS MANAGEMENT SYSTEM ----------------------------------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 1
Enter username: admin
Enter password: admin123
-----------------------------------
1. Add Car
2. Remove Car
3. Add Driver
4. Remove Driver
5. View Cars
6. View Drivers
7. View Bookings
8. View Users
9. Logout
-----------------------------------
Enter choice: 7
Car ID        Passenger Name      Contact       Rental Charge  Booking Date   Driver ID
-------------------------------------------------------------------------------------------
01            Gourav Singh Panwar 9009049332    1000.00        10-02-2025     01
02            Ojasvi Sharma       9874930143    800.00         02-02-2025
03            Abhay Choudhary     9009049332    1200.00        01-02-2025
-----------------------------------
```

12. Passenger Menu / Functionalities -

```
                    ---------- HIRE CARS MANAGEMENT SYSTEM ----------
1. Admin Login
2. Passenger Login
3. Exit
Enter choice: 2
1. Login
2. Register
Enter choice: 1
Enter username: gourav
Enter password: gourav@123
-----------------------------------
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice:
```

13. View Cars -

```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 1
Car ID          Model               Capacity  Rental Charge  Available
-------------------------------------------------------------------------
01              Swift Dezire        4         1000.00        No
02              verna               4         800.00         No
03              Fortuner            6         1200.00        No
-----------------------------------
```

14. Search Car -

```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 3
Enter required capacity: 6
Enter maximum rental charge: 1500
Car ID        Model              Capacity  Rental Charge  Available
-------------------------------------------------------------------
04            Ertiga             6         1500.00        Yes
-----------------------------------
```

15. Book A Car With Driver -

```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 4
Car ID        Model              Capacity  Rental Charge  Available
-------------------------------------------------------------------
04            Ertiga             6         1500.00        Yes
-----------------------------------
Enter Car ID to book: 04
Enter Passenger Name: Abhay Choudhary
Enter Passenger Contact (10 digits): 9009049334
Enter Booking Date (DD-MM-YYYY): 10-01-2025
Do you want to book car with a driver? (y/n): y
-----------------------------------
Car booked successfully! Driver assigned: 02
-----------------------------------
```

16. Book a Car Without Driver -

```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 4
Car ID        Model              Capacity  Rental Charge  Available
-------------------------------------------------------------------------
04            Ertiga             6         1500.00        Yes
-----------------------------------
Enter Car ID to book: 04
Enter Passenger Name: Abhay Choudhary
Enter Passenger Contact (10 digits): 9009049334
Enter Booking Date (DD-MM-YYYY): 10-02-2025
Do you want to book car with a driver? (y/n): n
-----------------------------------
Car booked successfully! Car booked successfully without a driver.
-----------------------------------
```

18. Cancel Booking -

```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 6
Car ID        Model              Capacity  Rental Charge  Available
-------------------------------------------------------------------------
04            Ertiga             6         1500.00        No
Enter Car ID to cancel booking: 04
Enter your contact number: 9009049334
Booking canceled successfully! Money refunded for Car ID: 04
-----------------------------------
```

16. View Invoice of Booked Cars –

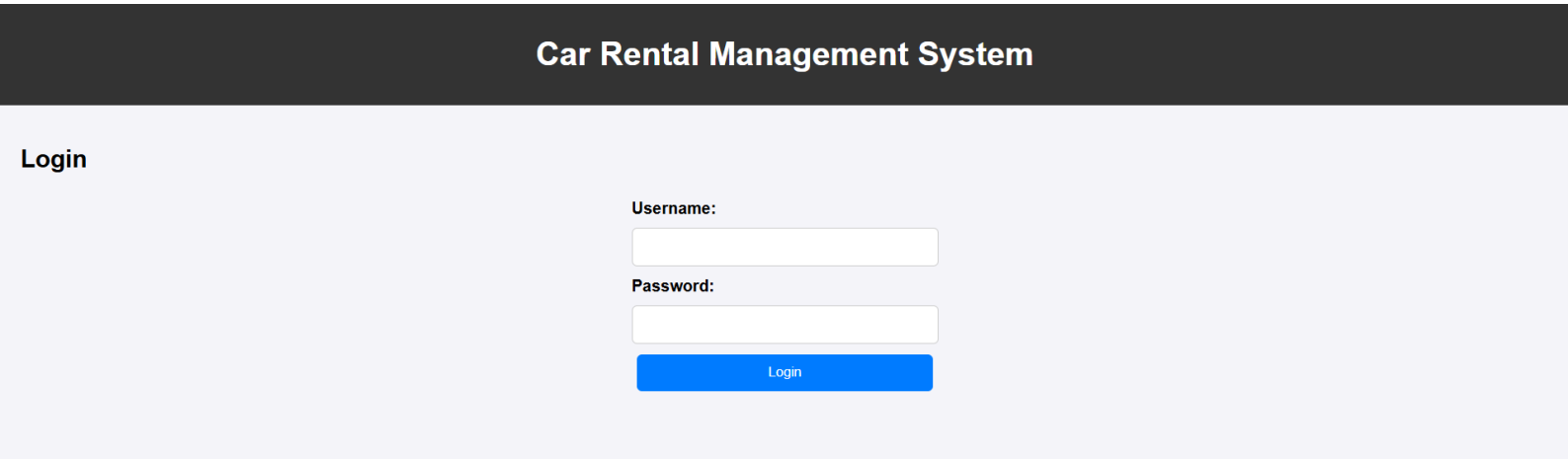
```
1. View Cars
2. View Drivers
3. Search Car
4. Book Car
5. View Invoice
6. Cancel Booking
7. Logout
-----------------------------------
Enter choice: 5
-----------------------------------
Booking Invoice:
Car ID: 01
Passenger Name: Ojasvi Sharma
Passenger Contact: 9009040334
Rental Charge: 1300.00
Booking Date: 10-01-2025
Driver Name: Verma
Driver Contact: 9009049443
-----------------------------------
```

❑ GIT-HUB LINK FOR THE ENTIRE C++ CODE –

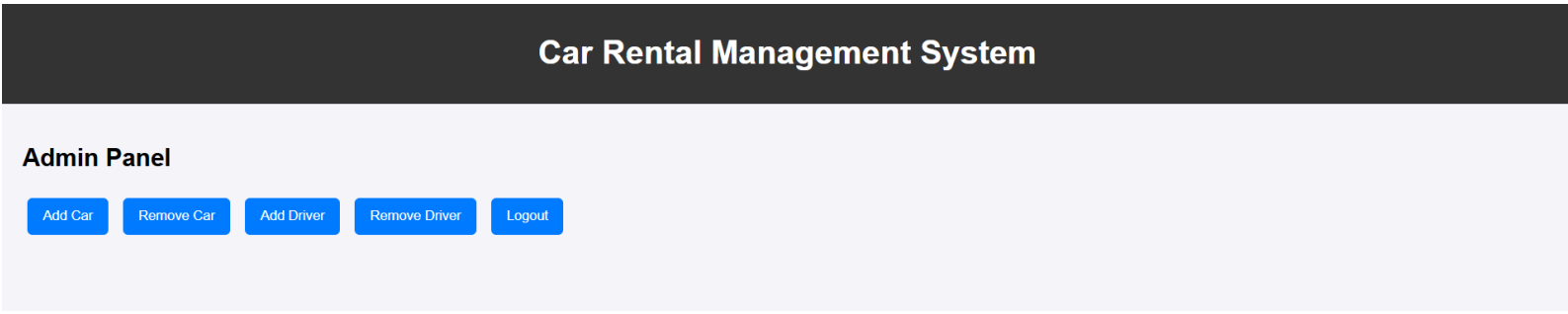
https://github.com/gouravsinghpanwar/Data-Structure

➢ Basic GUI –

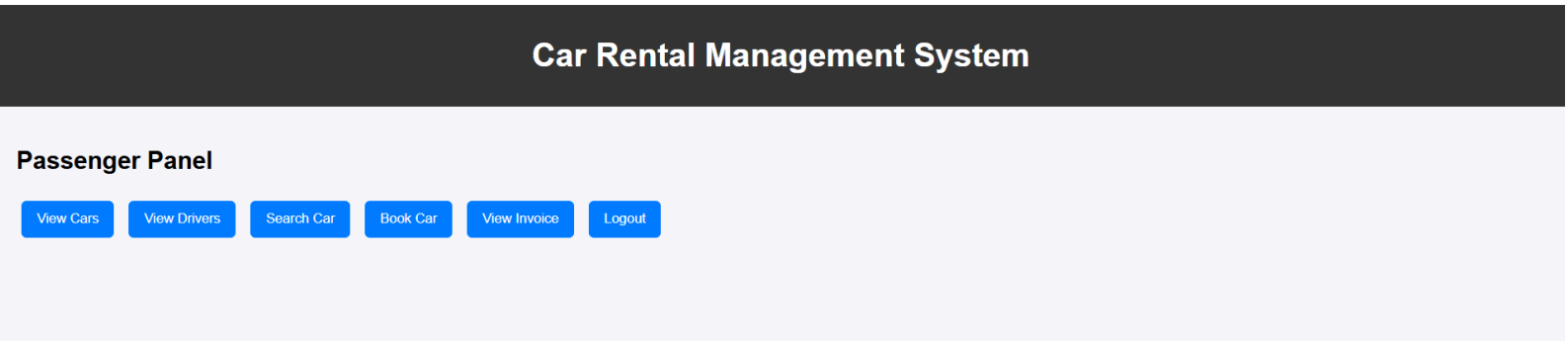
1. Login Section -



2. Admin Section -



3. Passenger Section -



- **Potential Improvements in Future –**

i. User Interface –

ii. Consider implementing a graphical user interface (GUI) for a more user-friendly experience.

2. Enhanced Security –

i. Passwords should be stored securely (e.g., hashed) instead of in plain text.

ii. Implement user roles and permissions for better access control.

3. Database Integration –

i. Transition from text file storage to a database management system (DBMS) for better data handling and scalability.

4. Additional Features –

i. Implement features such as car maintenance tracking, user reviews, and ratings for cars and drivers.
ii. Add a notification system for booking confirmations and reminders.

➢ CONCLUSION AND LEARNINGS –

- The Car Hire Management System is a functional application that effectively manages car rentals for both administrators and passengers. It incorporates essential features such as user authentication, car and driver management, and booking capabilities, all while ensuring data persistence through file handling. The project highlights the importance of modular programming, input validation, and user experience, demonstrating how a well-structured codebase can enhance usability and maintainability.

- Key learnings from this project include the significance of choosing appropriate data structures for effective data organization, the necessity of robust input validation to ensure data integrity, and the need for secure handling of user credentials. Additionally, the experience underscored the importance of scalability and the potential for future enhancements, such as transitioning to a graphical user interface and implementing a database management system. Overall, this project provided valuable insights that will be beneficial for future endeavors.

THANK YOU !!!