Semester Project Documentation

Semester Project Title: Car-Rental System

Student Details

(Fill according to number of team/group members)

	Student Name	Student Reg #	Student Degree
Student-1	M Dawood Bilal	2022365	FEEE
Student-2	M Ashir	2022353	FEEE
Student-3	Hassan Ahmed Rana	2022208	FEEE
Student-4	Taha Nabeegh	2022365	FEEE

Main Features

1.File handling

2.user interface

3.Admin interface

4.Entering new car details

5.Removing any car

6.Booking any car (user can book it)

7.Admin can log in and log out into the application with protected credentials (username ***admin***, password ***admin***)

8.View available lists of cars added by employee or admin itself

9.View all bookings made

10.used Vector<> library and attributes to call the objects of any class of function anywhere

11.View all employee info (admin can)

12.Add/Remove employees(admin can)

Types of Users

1. User(renter)
2. Employee
3. Admin

Requirements Breakdown

This code is a Car Rental System implemented using Object-Oriented Programming (OOP) in C++. It consists of various functions, each serving a specific purpose. The main features of the code are as follows:

- Modular Design:
  The code is divided into several functions, each responsible for a specific task. Functions like printmainmenu(), logout(), printmanagement(), cancelbooking(), viewbookings(), listofcars(), and employetab() are defined to handle different scenarios and provide a clear menu-driven interface to the user.

- Data Persistence:
  The code uses file handling operations to store and retrieve data. It utilizes text files such as "carsdata.txt", "bookings.txt", and "Employeecredentials.txt" to store information about cars, bookings, and employee credentials, respectively. This allows the system to maintain data across multiple sessions and ensure data integrity.

- Car Class:
  The code defines a Car class to represent car objects. It has attributes like name, model, year, rentCost, and available. The class provides methods like addcar(), removecar(), and showCars() to add a car to the system, remove a car, and display the available cars for booking, respectively.

- User and Management Classes:

The code defines user and management classes. The user class represents a user of the car rental system and provides functionalities like booking a car, canceling a booking, and viewing previous bookings. The management class extends the employee class and represents the administrative functionalities of the system, such as adding and removing cars, managing bookings, and managing employees.

- Employee Management:

The code includes functions like addEmployeeToList(), removeEmployee(), and viewemployelist() to manage employees. It allows adding and removing employees from the system and provides a list of employees for viewing.

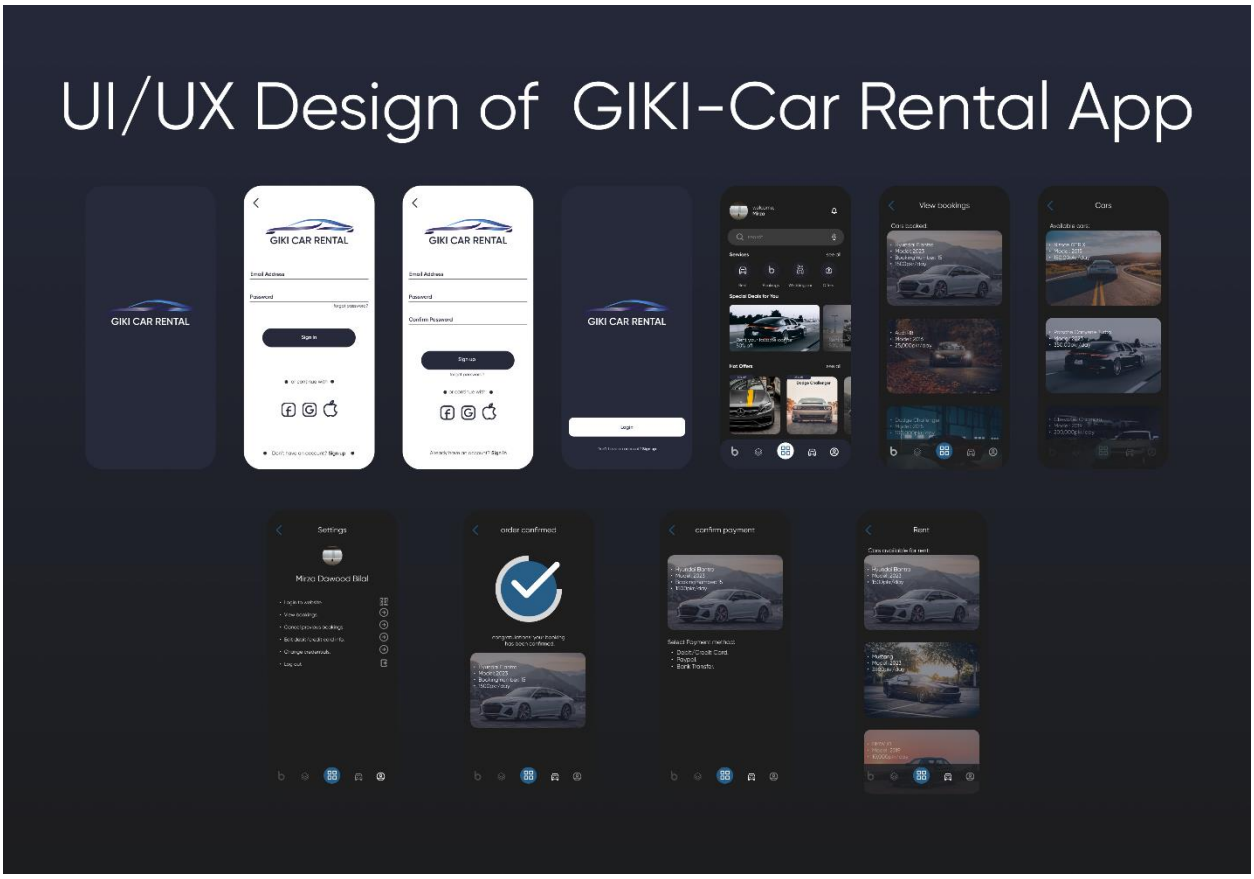
- Conclusion:

In conclusion, this code implements a Car Rental System using OOP principles in C++. It demonstrates the use of classes, file handling, and modular design to create a user-friendly interface for booking cars, managing bookings, and performing administrative tasks. The code showcases the capabilities of OOP and emphasizes code reusability, maintainability, and data persistence.

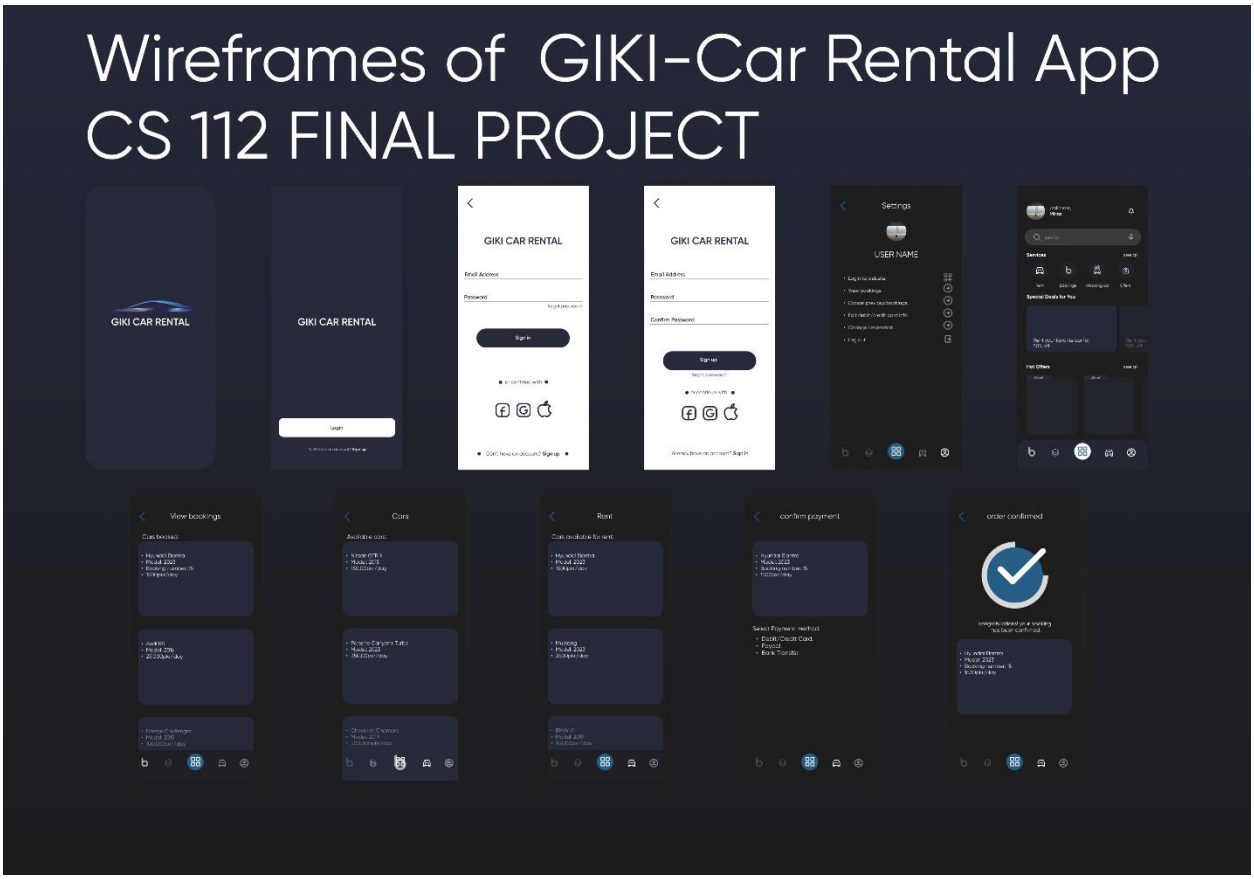
GUI Screens / Design / Wireframes

UI/UX design of GIKI Car rental app

- Full fledge design made:
  link to figma:
  https://www.figma.com/file/yp9ltF68PxJ9dl9LXYbugc/car-rental-app-design?type=design&node-id=38%3A426&t=rxtZpP0kxJz6POMP-1



- WIREFRAMES OF THE APP CREATED



MIRZA

@

- Conclusion on Design

This design was made just as a sample, about how a UI/UX design of the program of Car rental app can be if its visualized and developed.

[see full design on the figma link: https://www.figma.com/file/yp9ItF68PxJ9dl9LXYbugc/car-rental-app-design?type=design&node-id=38%3A426&t=rxtZpP0kxJz6POMP-1]

Features to Codding Matrix

Sr no.	Feature Name	OOP Type Used	Functions Created	Variables / Obj Created	Line of Code Written
1	Car rental system	Encapsulation, File handling, Inheritance (multiple, multilevel) both involved.	We have created different functions their names are below table.	Various variables are created and used each has its own importance	640

FUNCTIONS CREATED:

```
printmainmenu(), logout(), printmanagement(), cancelbooking(), viewbookings(),
listofcars(), employetab(), viewemployelist(), addEmployeeToList(),
saveAdminCredentials(), saveUserCredentials(), saveEmployeCredentials();
```

CIASSES CREATED:

1. CAR CLASS
2. CAR RENTAL CLASS
3. USER CLASS
4. EMPLOYEE CLASS
5. MANAGEMENT CLASS

THE END.