**Project Title:**

CAR RENTAL MANAGEMENT SYSTEM WITH USER AUTHENTICATION AND PAYMENT INTEGRATION

**Team Members**

|  |  |
| --- | --- |
| **Name** | **Roll Number** |
| Mahnoor Hussain | 23k-0535 |
| Urooj Baloch | 23k-0071 |

---

**1. Executive Summary**

**Overview:**  
The Car Rental Management System is a console-based application developed in C++ to facilitate car rentals. This system allows users to register, log in securely, choose a car (economy or luxury), specify rental duration, process payments (cash or credit card), and generate receipts. The project focuses on implementing core programming concepts such as object-oriented design, input validation, and user authentication.

**Key Findings:**

* The system successfully integrates user registration, login, car selection, and payment handling.
* Payment options (cash and credit card) are implemented, with validation for credit card details.
* A rental receipt is generated with all relevant information, including car details and total charges.
* The project meets all outlined objectives, including user authentication, input validation, and payment processing.

**2. Introduction**

**Background:**  
The demand for car rental services is growing, and businesses face challenges in providing a seamless, secure, and efficient user experience. This project aims to solve these challenges by developing a straightforward, console-based car rental system that incorporates user authentication and payment options, making the process smoother and more reliable.

**Project Objectives:**

* Develop a C++ application for car rental, including user registration and login.
* Enable car selection with different types (economy and luxury).
* Integrate payment methods (cash and credit card).
* Ensure input validation for fields such as email, CNIC, and passwords.
* Generate rental receipts and calculate the total bill based on rental hours and car type.
* Demonstrate object-oriented programming by designing classes for user, car, and payment details.

**3. Project Description**

**Scope:**

* **Inclusions:**
  + User management: Registration, login, and email/password validation.
  + Car selection: Display available cars (economy and luxury) and allow users to choose.
  + Payment handling: Simulate credit card and cash payments with validation.
  + Receipt generation: Generate a rental receipt with car details, total price, and transaction confirmation.
  + Input validation: Ensure proper format for email, CNIC, and user details.
* **Exclusions:**
  + Web interface: The system is console-based and does not have a web interface.
  + Complex payment gateway integration: Payments are simulated rather than processed in real-time.
  + Database management: The system does not store data permanently. All data is temporary during runtime.

**Technical Overview:**

* **Programming Language:** C++
* **Libraries Used:**
  + iostream: For input/output handling.
  + string: For string manipulation (user input).
  + iomanip: For formatting output (like setting precision).
  + ctime and cstdlib: For generating random receipt numbers and transaction dates.
  + thread and chrono: For simulating delay in payment verification.

**4. Methodology**

**Approach:**  
We worked collaboratively, dividing the project into key modules. Each group member was assigned specific responsibilities based on the project proposal. We followed an iterative approach, continuously testing and refining each module to ensure functionality and meet deadlines.

**Roles and Responsibilities:**

* **Mahnoor Hussain:** Responsible for designing and implementing the user class (registration, login), paymentdetails class (payment handling), and overall program flow.
* **Urooj Baloch:** Responsible for designing and implementing the car class (car selection, car details, receipt generation), handling testing and debugging, and ensuring the integration of user interactions with car selection and payment methods.

**5. Project Implementation**

**Design and Structure:**

* **User Class:** Handles user registration, login, email and CNIC validation.
* **PaymentDetails Class:** Handles credit card details input and validation.
* **Car Class:** Manages car details, rental rate, and generates receipts for car rentals.
* **Main Program Flow:** The user registers, logs in, selects a car, provides rental duration, makes a payment, and receives a receipt.

**Functionalities Developed:**

* **User Authentication:** Users can register, log in, and input their details with basic validation.
* **Car Selection:** A list of available cars (economy or luxury) is displayed, and users can choose a car.
* **Payment Handling:** Users can choose between cash and credit card payment options. The system simulates credit card validation and processes payments accordingly.
* **Receipt Generation:** After completing the rental, a receipt with car details, payment information, and total charges is generated.

**Challenges Faced:**

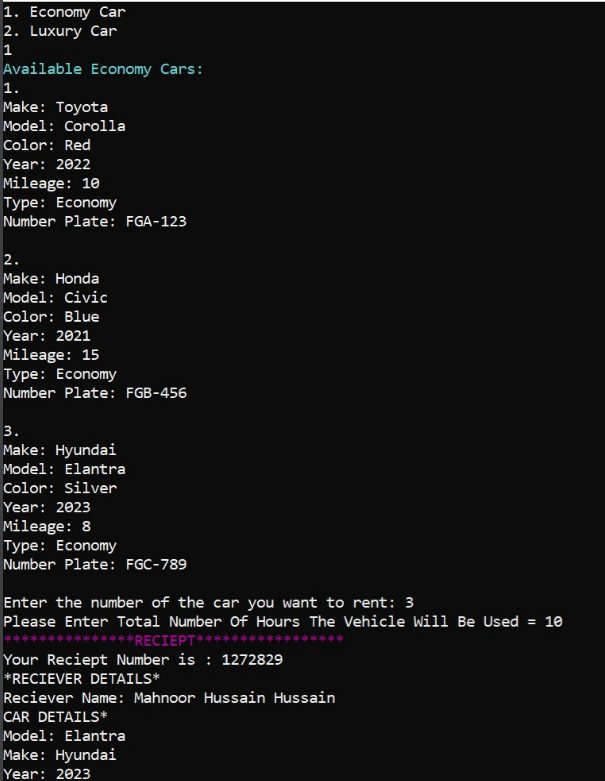
* **Input Validation:** Ensuring correct input formats for CNIC and email was challenging but was overcome by using loops and condition checks.
* **Payment Simulation:** Simulating credit card payment and transaction details required using random number generation for receipt IDs and transaction dates.

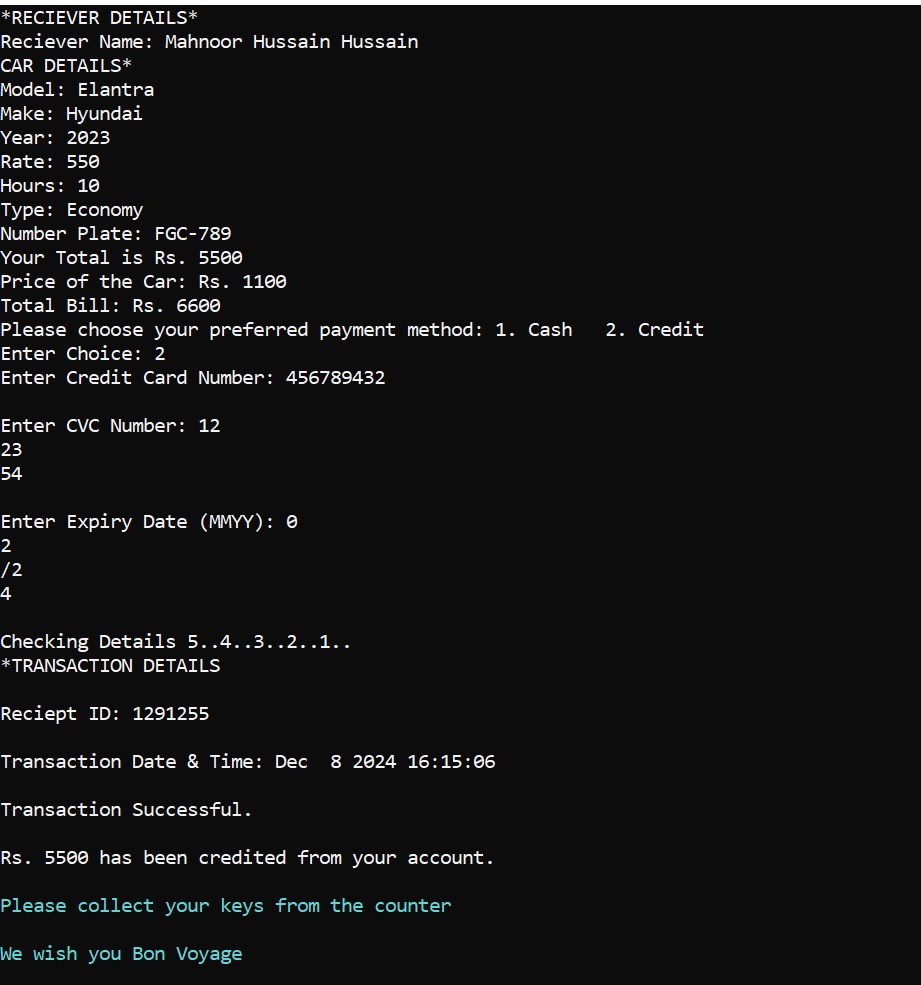
**6. Results**

**Project Outcomes:**

* The project successfully meets the objectives, including user authentication, car selection, payment processing, and receipt generation.
* The program runs smoothly, with proper validation checks and simulated payments.

**Screenshot of output:**  



**Testing and Validation:**

* Each feature was tested individually for input validation, payment processing, and car selection.
* Edge cases like incorrect CNIC or email formats were handled with error messages prompting the user to try again.

**7. Conclusion**

**Summary of Findings:**  
The Car Rental Management System was successfully developed and meets the initial goals outlined in the project proposal. It allows users to register, log in, select a car, provide rental duration, choose payment methods, and receive a receipt. The project demonstrates fundamental programming concepts such as object-oriented design, input validation, and basic payment processing.