CASESTUDY REVIEW CAR-CONNECT

DURGESWARI M

OVERVIEW

- Introduction
- Problem statement
- Objectives
- Technologies Used
- Functionalities
- Unit Testing
- Conclusion

INTRODUCTION

CarConnect is a Python-based car rental app using SQL and object-oriented principles. It has multiple layers for code reusability and scalability. Users can register, browse vehicles, and make reservations, while admins can manage fleet data, customer records, and generate reports.

PROBLEM STATEMENT

Many car rental platforms today lack real-time vehicle tracking, smooth user experience, and robust reservation systems. CarConnect solves these challenges with a secure, scalable, and database-driven platform for efficient car rental operations.

OBECTIVES

- Implementing a modular car rental system
- Managing vehicles, customers, reservations, and admin functionalities effectively.
- To ensure secure authentication for both customers and administrators.
- To handle the vehicle availability
- To incorporate custom exception handling to improve robustness.
- To implement unit testing to validate the functionality

TECHNOLOGIES USED

- Programming Language: Python
- Database: MySQL
- Libraries:

  mysql.connector for DB connectivity

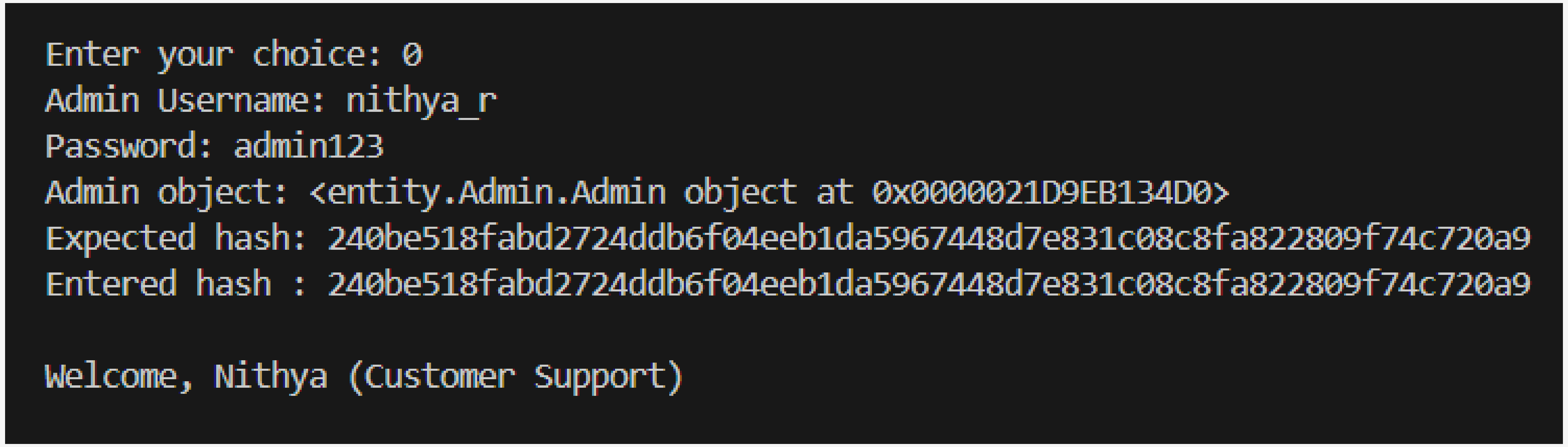
  unittest for testing

  hashlib for password hashing
- IDE: Visual Studio Code
- Other tools: GitHub

FUNCTIONALITIES

Admin Authentication:

- Admin login by entering correct username and password.
- Admin role is fetched and printed along with thier name
- Secure user authentication and authorization mechanisms.

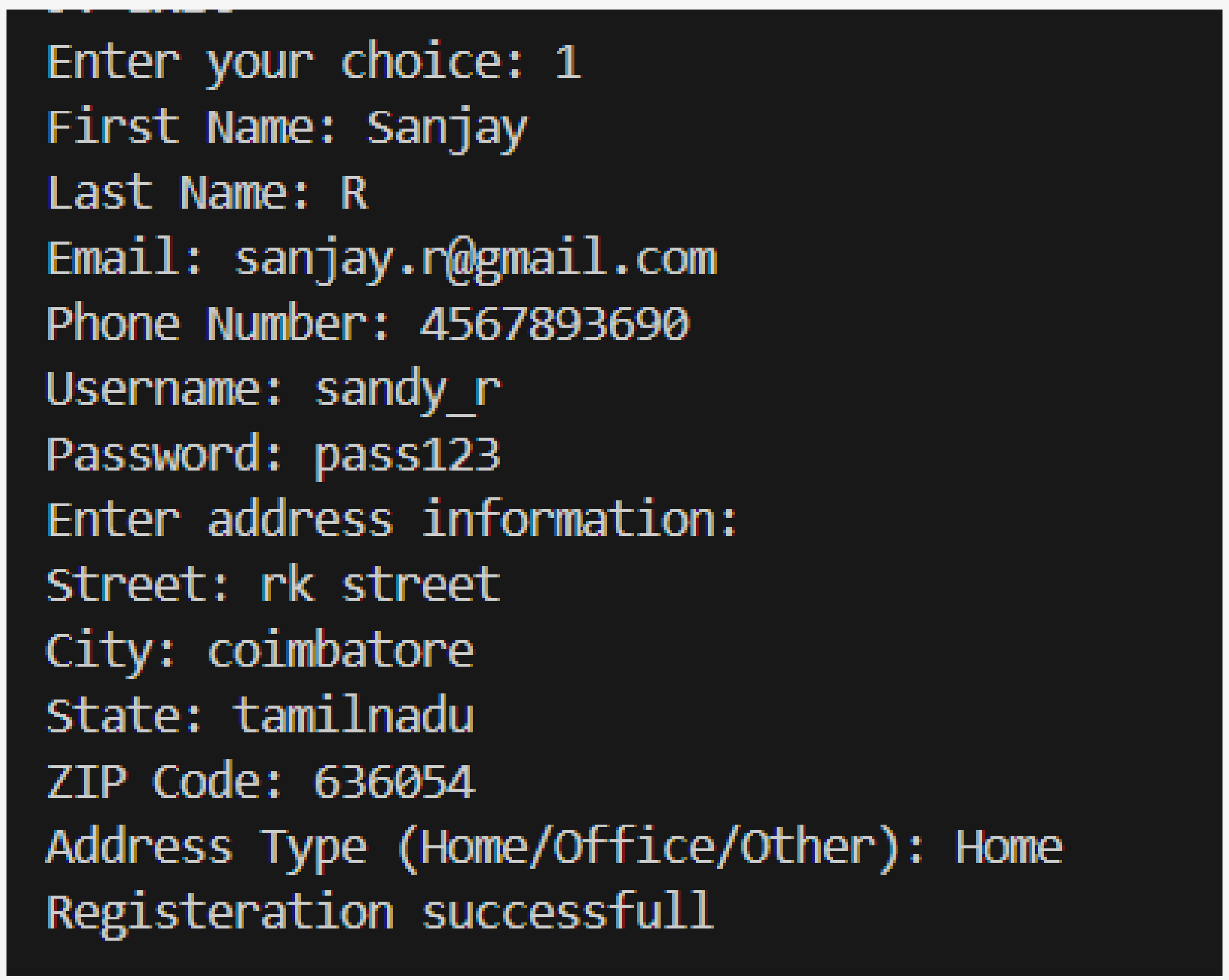


```
Enter your choice: 0
Admin Username: nithya_r
Password: admin123
Admin object: <entity.Admin.Admin object at 0x0000021D9EB134D0>
Expected hash: 240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c720a9
Entered hash : 240be518fabd2724ddb6f04eeb1da5967448d7e831c08c8fa822809f74c720a9

Welcome, Nithya (Customer Support)
```

Customer Registration with Address

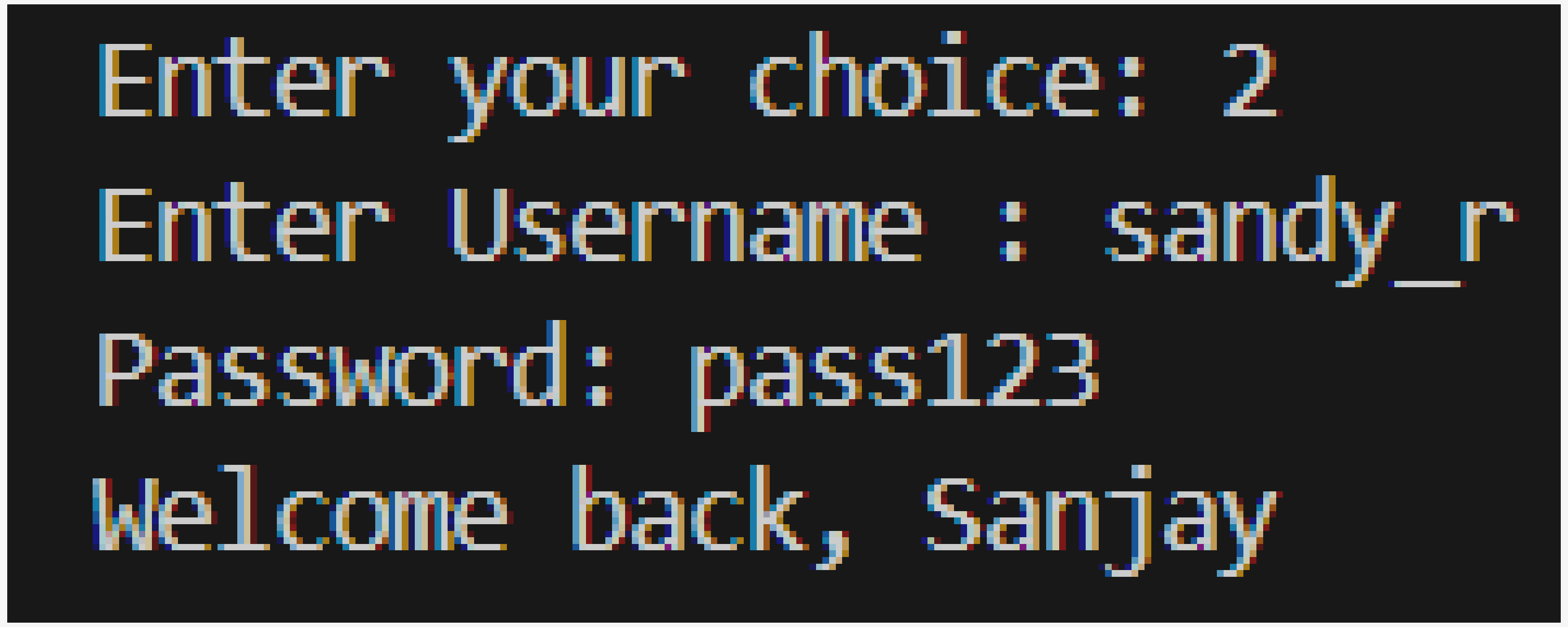
- Customer details collected and saved here.
- After saving, prompts for and saves customer’s address.



```
Enter your choice: 1
First Name: Sanjay
Last Name: R
Email: sanjay.r@gmail.com
Phone Number: 4567893690
Username: sandy_r
Password: pass123
Enter address information:
Street: rk street
City: coimbatore
State: tamilnadu
ZIP Code: 636054
Address Type (Home/Office/Other): Home
Registeration successfull
```

Customer Login

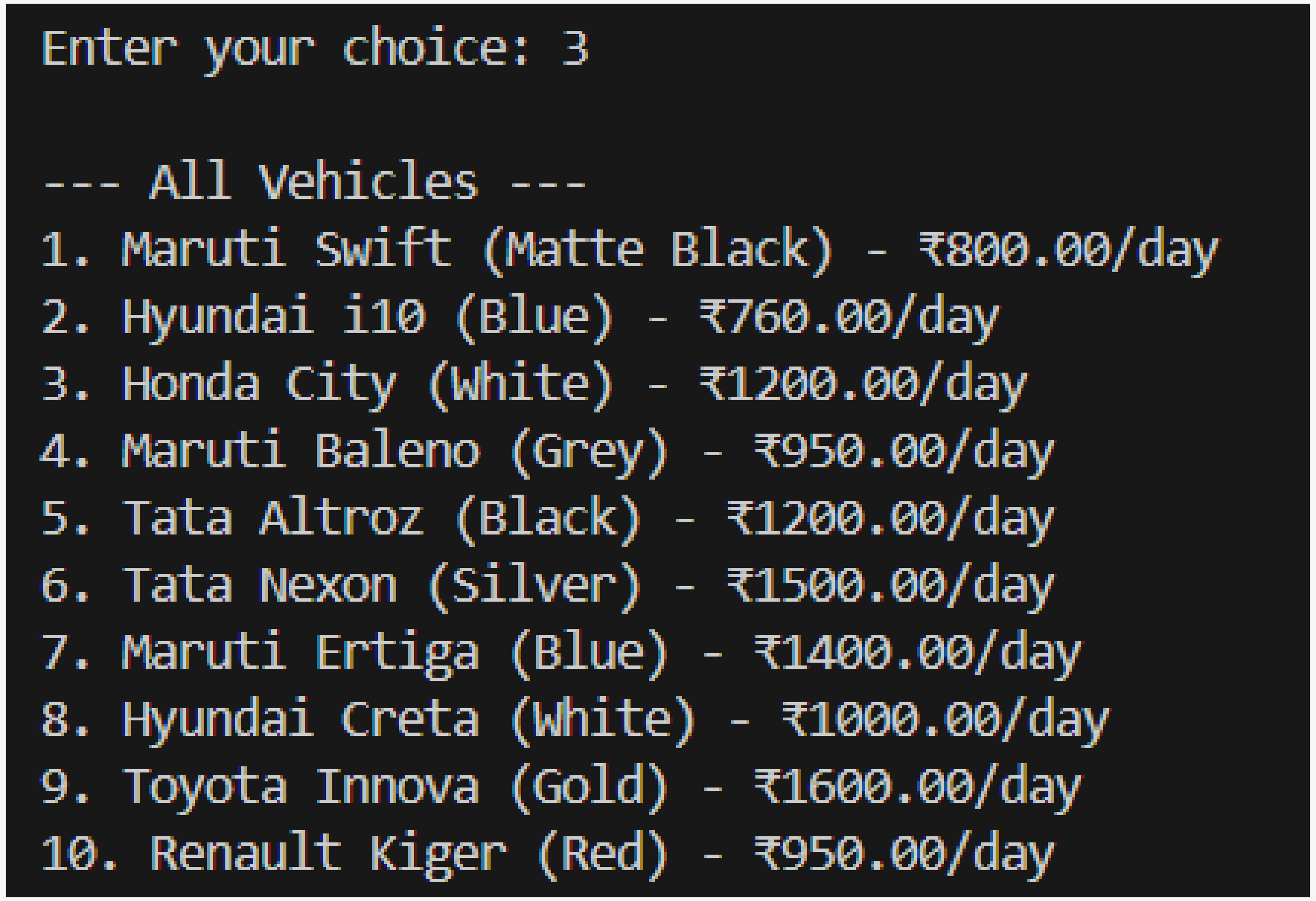
- Verifies customer login credentials using authenticate() method.
- It validates the password through customer authentication().



```
Enter your choice: 2
Enter Username : sandy_r
Password: pass123
Welcome back, Sanjay
```

View All Vehicles

- Displays all vehicle records from the database.

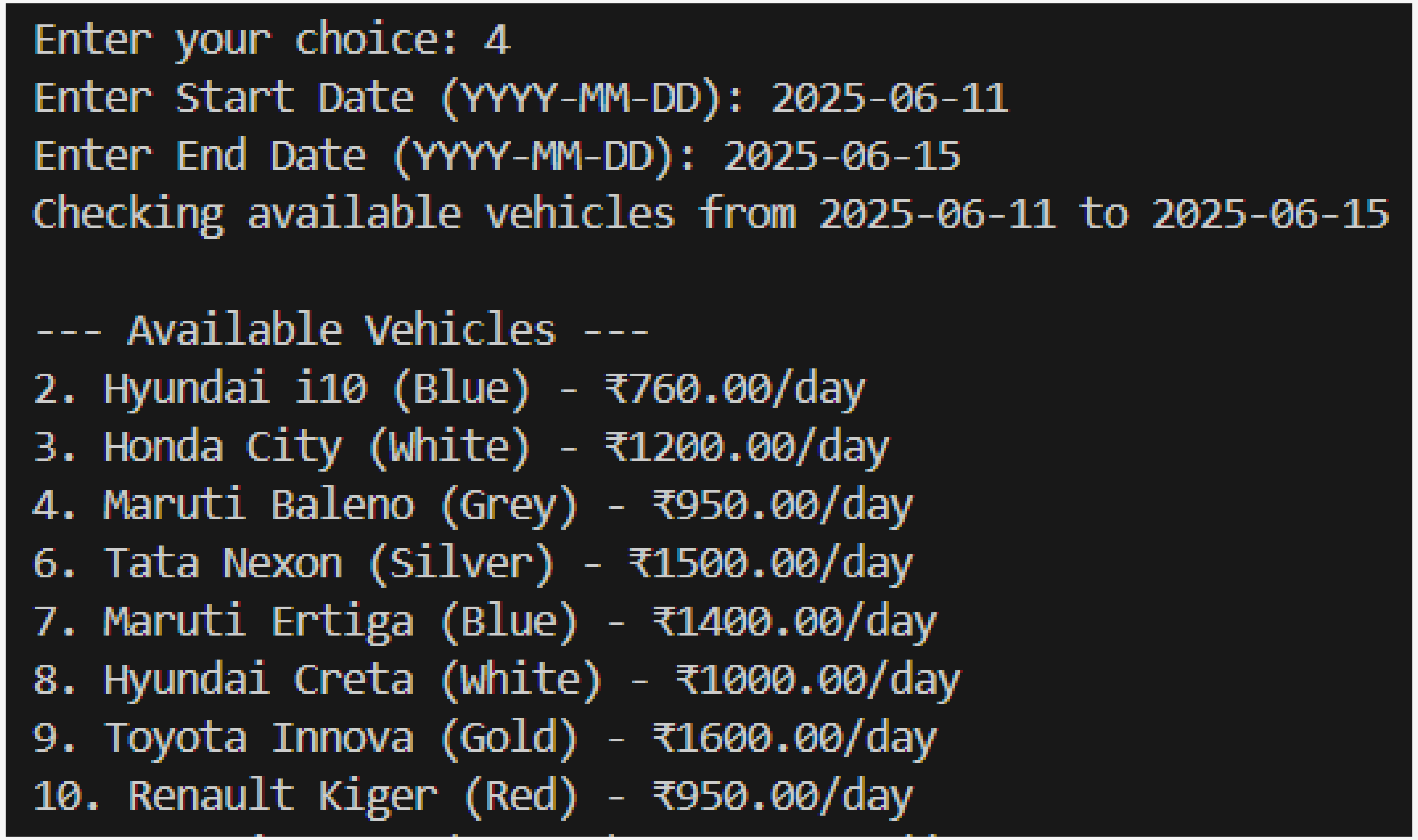


```
Enter your choice: 3

--- All Vehicles ---
1. Maruti Swift (Matte Black) - ₹800.00/day
2. Hyundai i10 (Blue) - ₹760.00/day
3. Honda City (White) - ₹1200.00/day
4. Maruti Baleno (Grey) - ₹950.00/day
5. Tata Altroz (Black) - ₹1200.00/day
6. Tata Nexon (Silver) - ₹1500.00/day
7. Maruti Ertiga (Blue) - ₹1400.00/day
8. Hyundai Creta (White) - ₹1000.00/day
9. Toyota Innova (Gold) - ₹1600.00/day
10. Renault Kiger (Red) - ₹950.00/day
```

View available Vehicles by date range:

- Filters and displays vehicles that are available between selected dates.

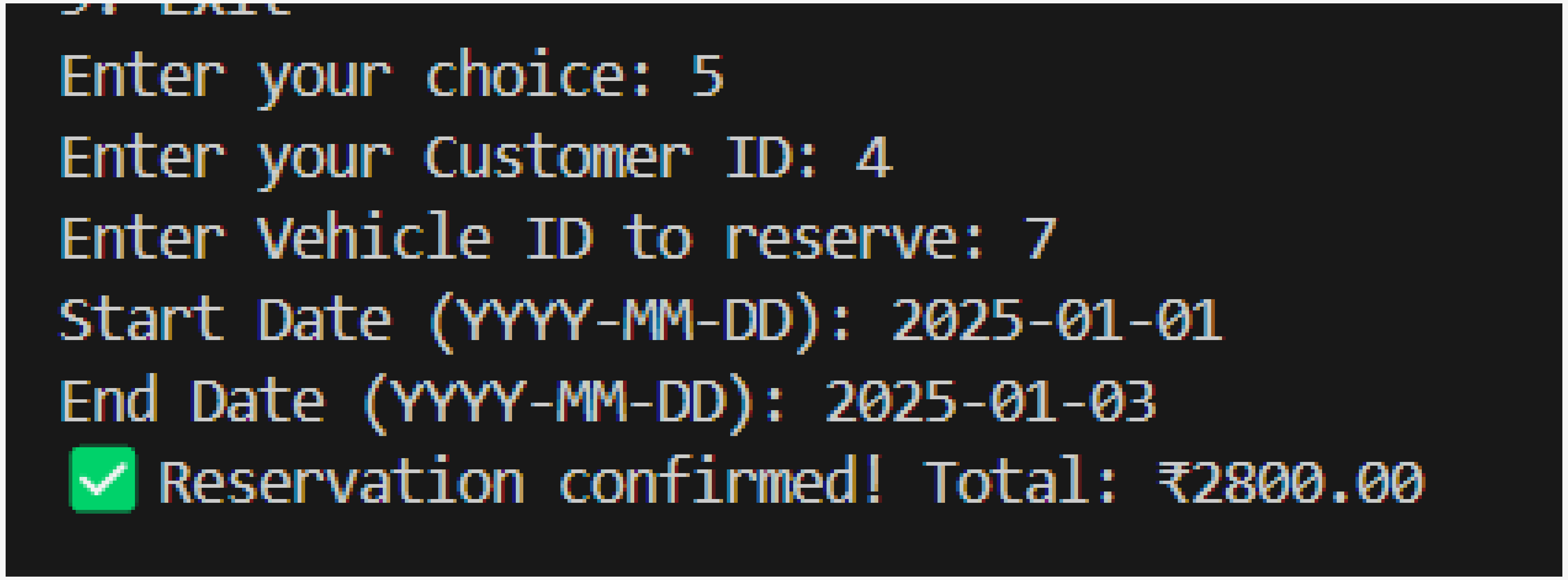


```
Enter your choice: 4
Enter Start Date (YYYY-MM-DD): 2025-06-11
Enter End Date (YYYY-MM-DD): 2025-06-15
Checking available vehicles from 2025-06-11 to 2025-06-15

--- Available Vehicles ---
2. Hyundai i10 (Blue) - ₹760.00/day
3. Honda City (White) - ₹1200.00/day
4. Maruti Baleno (Grey) - ₹950.00/day
6. Tata Nexon (Silver) - ₹1500.00/day
7. Maruti Ertiga (Blue) - ₹1400.00/day
8. Hyundai Creta (White) - ₹1000.00/day
9. Toyota Innova (Gold) - ₹1600.00/day
10. Renault Kiger (Red) - ₹950.00/day
```

Create a Reservation

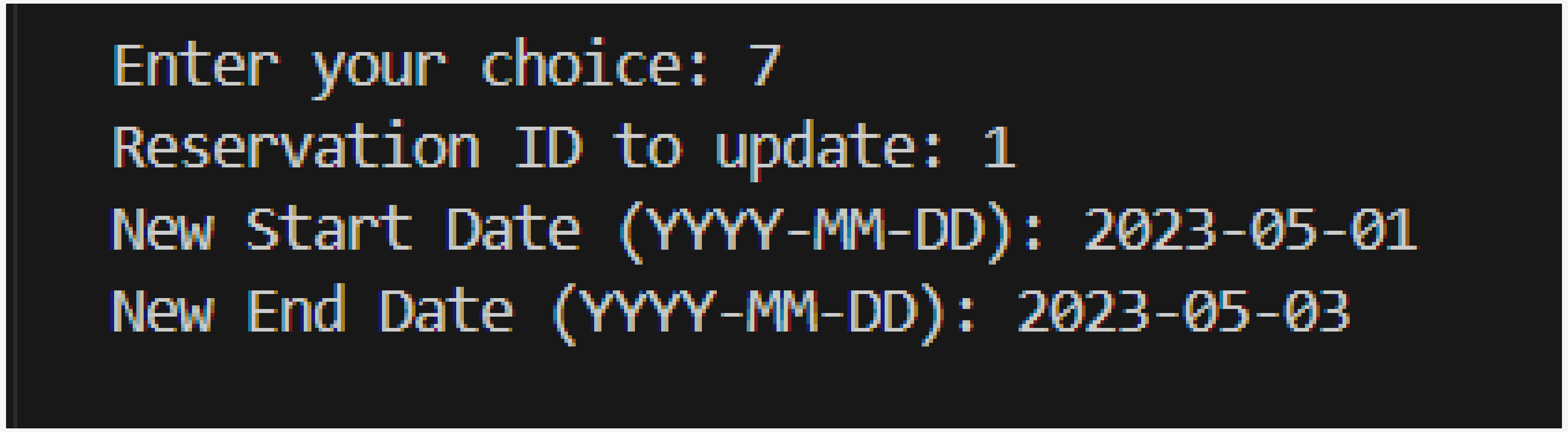
- Gets customer ID and vehicle ID.
- Checks if vehicle exists.
- Calculates reservation days and cost.



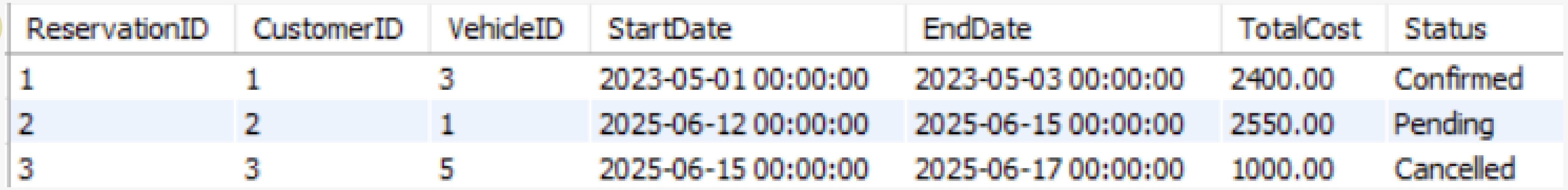
```
5. Exit
Enter your choice: 5
Enter your Customer ID: 4
Enter Vehicle ID to reserve: 7
Start Date (YYYY-MM-DD): 2025-01-01
End Date (YYYY-MM-DD): 2025-01-03
✅ Reservation confirmed! Total: ₹2800.00
```

Update Reservation

- Finds a reservation by ID.
- Updates the start/end dates, total cost, and status.



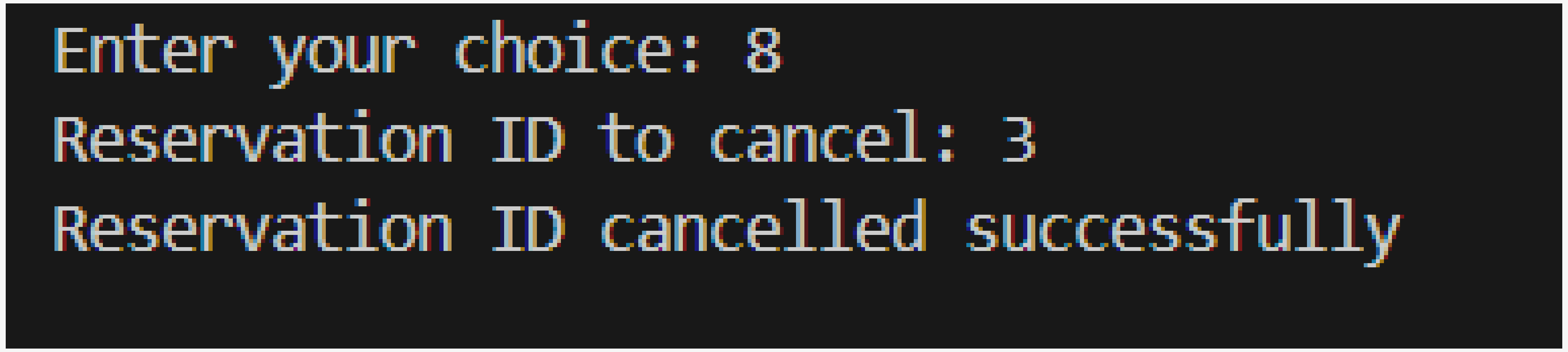
```
Enter your choice: 7
Reservation ID to update: 1
New Start Date (YYYY-MM-DD): 2023-05-01
New End Date (YYYY-MM-DD): 2023-05-03
```



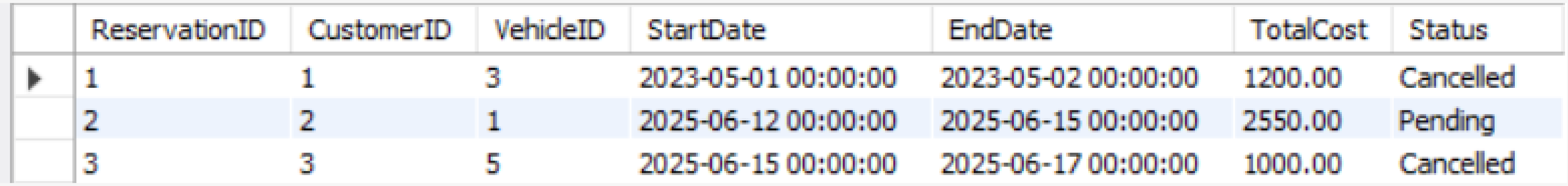
ReservationID	CustomerID	VehicleID	StartDate	EndDate	TotalCost	Status
1	1	3	2023-05-01 00:00:00	2023-05-03 00:00:00	2400.00	Confirmed
2	2	1	2025-06-12 00:00:00	2025-06-15 00:00:00	2550.00	Pending
3	3	5	2025-06-15 00:00:00	2025-06-17 00:00:00	1000.00	Cancelled

Cancel Reservation

- Cancels a reservation by ID.



```
Enter your choice: 8
Reservation ID to cancel: 3
Reservation ID cancelled successfully
```

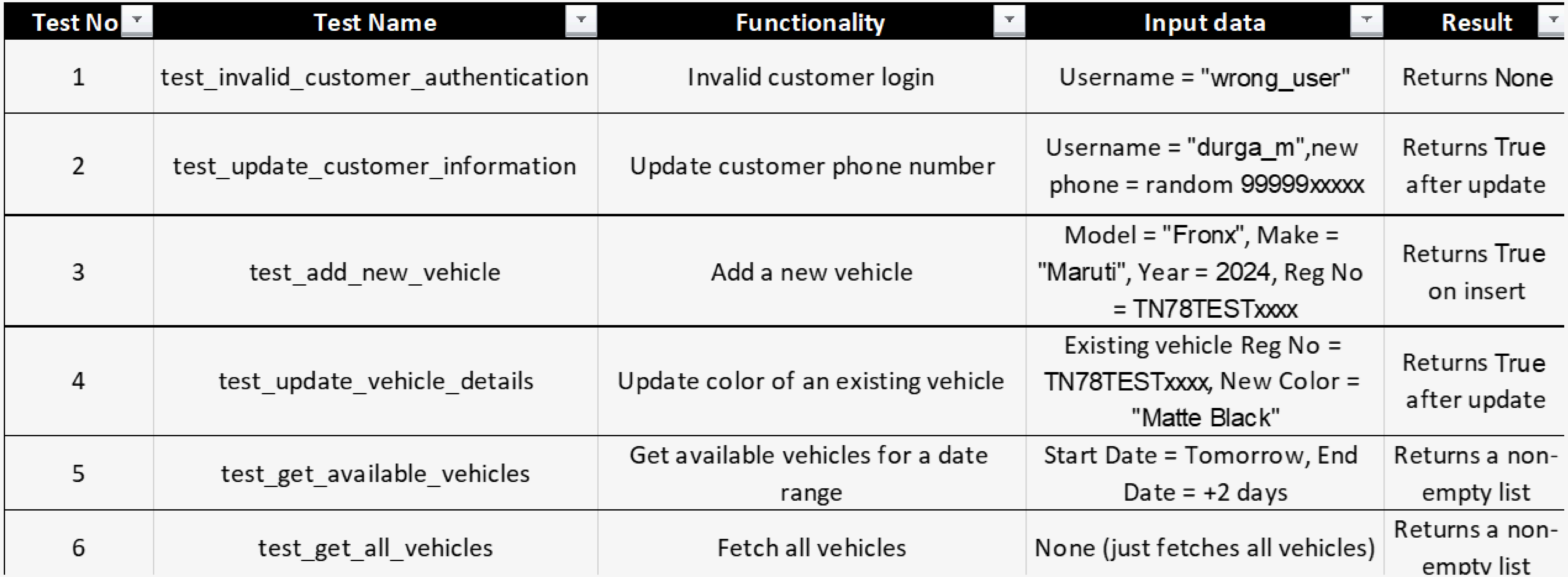


	ReservationID	CustomerID	VehicleID	StartDate	EndDate	TotalCost	Status
▶	1	1	3	2023-05-01 00:00:00	2023-05-02 00:00:00	1200.00	Cancelled
	2	2	1	2025-06-12 00:00:00	2025-06-15 00:00:00	2550.00	Pending
	3	3	5	2025-06-15 00:00:00	2025-06-17 00:00:00	1000.00	Cancelled

EXCEPTION HANDLING

- Created specific exceptions like **VehicleNotFoundException** and **ReservationException** to handle situations such as trying to access a vehicle that doesn't exist or when a reservation fails due to conflicts.
- Used **AuthenticationException** to handle login failures when customers or admins enter incorrect credentials.
- Implemented **InvalidInputException** to catch invalid user inputs, such as entering an end date earlier than the start date during reservation.
- Handled **DatabaseConnectionException** to manage errors during database connectivity and prevent application crashes.

UNIT TESTING



Test No	Test Name	Functionality	Input data	Result
1	test_invalid_customer_authentication	Invalid customer login	Username = "wrong_user"	Returns None
2	test_update_customer_information	Update customer phone number	Username = "durga_m",new phone = random 99999xxxxx	Returns True after update
3	test_add_new_vehicle	Add a new vehicle	Model = "Fronx", Make = "Maruti", Year = 2024, Reg No = TN78TESTxxxx	Returns True on insert
4	test_update_vehicle_details	Update color of an existing vehicle	Existing vehicle Reg No = TN78TESTxxxx, New Color = "Matte Black"	Returns True after update
5	test_get_available_vehicles	Get available vehicles for a date range	Start Date = Tomorrow, End Date = +2 days	Returns a non-empty list
6	test_get_all_vehicles	Fetch all vehicles	None (just fetches all vehicles)	Returns a non-empty list

Thank you for listening