

# AnjaliCarRental

January 1, 2024

```
[4]: cars_data = [  
    [1, 'Toyota', 'Corolla', 'ABC123', 'Sedan', 2018, 'Blue', 50, 'Y', 'N',  
      ↪ '2023-01-01', '2023-01-10', '200'],  
    [2, 'Honda', 'Civic', 'XYZ456', 'Sedan', 2019, 'Black', 40, 'N', 'Y',  
      ↪ '2023-01-05', '2023-01-15', '150'],  
    [3, 'Ford', 'Fiesta', 'DEF789', 'Hatchback', 2020, 'Red', 45, 'Y', 'N',  
      ↪ '2023-02-10', '2023-02-20', '180'],  
    [4, 'BMW', 'X5', 'GHI012', 'SUV', 2019, 'White', 60, 'N', 'N', '', '',  
      ↪ '220'],  
    [5, 'Mercedes', 'E-Class', 'JKL345', 'Sedan', 2021, 'Silver', 55, 'Y', 'Y',  
      ↪ '2023-03-15', '2023-03-25', '250'],  
    [6, 'Audi', 'A4', 'MNO678', 'Sedan', 2020, 'Grey', 50, 'N', 'N', '', '',  
      ↪ '190'],  
    [7, 'Toyota', 'Rav4', 'PQR901', 'SUV', 2018, 'Green', 65, 'Y', 'N',  
      ↪ '2023-04-05', '2023-04-15', '210'],  
    [8, 'Honda', 'Accord', 'STU234', 'Sedan', 2017, 'Silver', 45, 'N', 'Y',  
      ↪ '2023-05-10', '2023-05-20', '180'],  
    [9, 'Nissan', 'Altima', 'VWX567', 'Sedan', 2022, 'Black', 48, 'Y', 'N',  
      ↪ '2023-06-01', '2023-06-10', '200'],  
    [10, 'Tesla', 'Model 3', 'YZA890', 'Electric', 2021, 'Blue', 70, 'N', 'N',  
      ↪ '', '', '300'],  
    [11, 'Chevrolet', 'Malibu', 'BCD123', 'Sedan', 2019, 'White', 50, 'Y', 'Y',  
      ↪ '2023-07-15', '2023-07-25', '220'],  
    [12, 'Kia', 'Soul', 'EFG456', 'Compact', 2020, 'Yellow', 42, 'N', 'N', '',  
      ↪ '', '170'],  
    [13, 'Hyundai', 'Elantra', 'HIJ789', 'Sedan', 2018, 'Red', 45, 'Y', 'N',  
      ↪ '2023-08-20', '2023-08-30', '190'],  
    [14, 'Ford', 'Escape', 'KLM012', 'SUV', 2022, 'Silver', 60, 'N', 'Y',  
      ↪ '2023-09-05', '2023-09-15', '230'],  
    [15, 'Volkswagen', 'Jetta', 'NOP345', 'Sedan', 2019, 'Grey', 48, 'Y', 'N',  
      ↪ '2023-10-10', '2023-10-20', '200'],  
    [16, 'Subaru', 'Outback', 'QRS678', 'SUV', 2021, 'Blue', 55, 'N', 'N', '',  
      ↪ '', '240'],  
    [17, 'Lexus', 'ES', 'TUV901', 'Sedan', 2020, 'Black', 52, 'Y', 'Y',  
      ↪ '2023-11-15', '2023-11-25', '230'],  
]
```

```

    [18, 'Mazda', 'CX-5', 'WXY234', 'SUV', 2018, 'Red', 58, 'N', 'N', '', '',
    ↪'210']
]

def add_car():
    print("\nAdding a New Car")

    car_id = input("Enter Car ID: ")
    brand = input("Enter Brand: ")
    model = input("Enter Model: ")
    plate_number = input("Enter Plate Number: ")
    car_type = input("Enter Car Type: ")
    year = input("Enter Year: ")
    color = input("Enter Color: ")
    daily_rate = input("Enter Daily Rate: ")

    # Assuming 'N' for both booked and returned status for a new car
    new_car = [car_id, brand, model, plate_number, car_type, year, color,
    ↪daily_rate, 'N', 'N']

    # Append the new car to the cars_data list
    cars_data.append(new_car)

    print("Car added successfully!")

def modify_car():
    print("\nModifying Car Details")

    car_id = input("Enter Car ID to modify: ")
    car_found = False

    for car in cars_data:
        if car[0] == car_id:
            car_found = True
            print("\nCurrent Details:")
            print("ID:", car[0])
            print("Brand:", car[1])
            print("Model:", car[2])
            print("Plate Number:", car[3])
            print("Car Type:", car[4])
            print("Year:", car[5])
            print("Color:", car[6])
            print("Daily Rate:", car[7])

            # Update car details

```

```

        car[1] = input("\nEnter Brand (Leave blank to keep current): ") or _
    ↪car[1]
        car[2] = input("Enter Model (Leave blank to keep current): ") or _
    ↪car[2]
        car[3] = input("Enter Plate Number (Leave blank to keep current): _
    ↪") or car[3]
        car[4] = input("Enter Car Type (Leave blank to keep current): ") or _
    ↪car[4]
        car[5] = input("Enter Year (Leave blank to keep current): ") or _
    ↪car[5]
        car[6] = input("Enter Color (Leave blank to keep current): ") or _
    ↪car[6]
        car[7] = input("Enter Daily Rate (Leave blank to keep current): ") _
    ↪or car[7]

        print("\nCar details updated successfully!")
        break

    if not car_found:
        print("Car ID not found.")

def search_car():
    print("\nSearch for a Car")

    search_criteria = input("Enter Car ID to search: ")
    car_found = False

    for car in cars_data:
        if car[0] == search_criteria:
            car_found = True
            print("\nCar Details:")
            print("ID:", car[0])
            print("Brand:", car[1])
            print("Model:", car[2])
            print("Plate Number:", car[3])
            print("Car Type:", car[4])
            print("Year:", car[5])
            print("Color:", car[6])
            print("Daily Rate:", car[7])
            break

    if not car_found:
        print("Car ID not found.")

def update_payment():
    print("\nUpdate Payment Details for a Car")

```

```

car_id = input("Enter Car ID: ")
payment_type = input("Enter Payment Type (Cash/Cheque): ")

car_found = False

for car in cars_data:
    if car[0] == car_id:
        car_found = True
        car[8] = payment_type # Assuming payment type is at index 8 in the
↪car's details
        print("Payment details updated successfully.")
        break

if not car_found:
    print("Car ID not found. Payment details could not be updated.")

def return_rented_car():
    print("\nReturning a Rented Car")

    customer_name = input("Enter customer's name: ")
    brand = input("Enter car's brand: ")
    model = input("Enter car's model: ")
    carplate = input("Enter car's plate number: ")
    car_id = input("Enter car ID: ")

    car_found = False

    for car in cars_data:
        if car[1] == car_id and car[2] == brand and car[3] == model and car[4]
↪== carplate:
            car_found = True
            if car[7] == 'Y':
                print("Car is already returned.")
            else:
                car[7] = 'Y' # Assuming 'Y' indicates car returned in the car
↪details
                print("Car returned successfully.")
                break

    if not car_found:
        print("Car not found or details mismatched.")

def display_car():
    print("** Available Cars **")
    print("ID\tBrand\tModel\tPlate No\tType\tYear\tColor\tDaily
↪Rate\tBooked\tReturned")
    for car in cars_data:

```

```

    ↪print(f"{car[0]}\t{car[1]}\t{car[2]}\t{car[3]}\t{car[4]}\t{car[5]}\t{car[6]}\t{car[7]}\t{car[8]}\t{car[9]}")

def view_bookings():
    bookings = [
        {'BookingID': '1', 'CarID': '1', 'Date': '2023-01-15', 'Status': 'Active'},
        ↪{'BookingID': '2', 'CarID': '3', 'Date': '2023-02-20', 'Status': 'Inactive'},
        # ... (other booking details)
    ]

    if not bookings:
        print("You don't have any bookings.")
    else:
        print("Your Bookings:")
        for booking in bookings:
            ↪print(f"Booking ID: {booking['BookingID']}, Car ID: {booking['CarID']}, "
                  f"Date: {booking['Date']}, Status: {booking['Status']}")

    action = input("\nDo you want to add (A) or remove (R) a booking? (A/R): ").lower()

    if action == 'a':
        # Logic to add a new booking
        booking_details = {
            'BookingID': input("Enter Booking ID: "),
            'CarID': input("Enter Car ID: "),
            'Date': input("Enter Date (YYYY-MM-DD): "),
            'Status': input("Enter Status: ")
        }
        bookings.append(booking_details)
        print("New booking added successfully!")
    elif action == 'r':
        # Logic to remove a booking
        booking_id_to_remove = input("Enter the Booking ID to remove: ")
        for booking in bookings:
            if booking['BookingID'] == booking_id_to_remove:
                bookings.remove(booking)
                ↪print(f"Booking ID {booking_id_to_remove} removed.")
                break
        else:
            print(f"Booking ID {booking_id_to_remove} not found.")

# Member Customer Functionality
def customer_menu():

```

```

while True:
    print("\n** Member Customer Menu **")
    print("1 = View Available Cars")
    print("2 = Rent a Car")
    print("3 = Return a Car")
    print("4 = View My Bookings")
    print("5 = Exit to Main Menu")

    mselect = input("Select an option: ")

    if mselect == '1':
        def view_available_cars():
            print("Available Cars:")
            for car in cars_data:
                if car['Available']:
                    print(f"ID: {car['ID']} - {car['Brand']} {car['Model']}_{
↪({car['Plate']})")
                pass

    elif mselect == '2':
        # Logic to rent a car
        pass

    elif mselect == '3':
        # Logic to return a car
        pass

    elif mselect == '4':
        # Logic to view bookings
        pass

    elif mselect == '5':
        print("Exiting to Main Menu...")
        break # Exit the loop and return to the main menu

    else:
        print("\nInvalid input. Please select a valid option.")
        input('Press Enter to continue...')

# Non-Member Customer Functionality
def customer_menu2():
    while True:
        print("\n** Non-Member Customer Menu **")
        print("1 = View Available Cars")
        print("2 = Register as Member")
        print("3 = Exit to Main Menu")

```

```

mselect = input("Select an option: ")

if mselect == '1':
    # Logic to view available cars
    pass

elif mselect == '2':
    # Logic to register as a member
    pass

elif mselect == '3':
    print("Exiting to Main Menu...")
    break # Exit the loop and return to the main menu

else:
    print("\nInvalid input. Please select a valid option.")
    input('Press Enter to continue...')

def rent_car():
    print("\nRent a Car")
    # Logic to rent a car based on your system
    customer_name = input("Enter your name: ")
    # Take necessary inputs: car details, rental duration, etc.
    # Perform operations like updating car availability, creating a rental
    ↪record, etc.
    print(f"Hello {customer_name}! Your car has been rented successfully.")

def check_available_cars():
    print("\nAvailable Cars:")
    # Logic to display available cars
    for car in cars_data:
        if car[9] == 'N': # Assuming 'N' signifies car availability
            print(f"Car ID: {car[0]}, Brand: {car[1]}, Model: {car[2]}, Plate_
            ↪No: {car[3]}")
        if all(car[9] == 'Y' for car in cars_data):
            print("No cars available at the moment.")

```

```

[1]: ##### Admin's Function
# Admin Menu
def admin_menu():
    while True:
        print("** Admin Menu **")
        print("\nPlease select an option:")
        print("1 = Add Car")
        print("2 = Modify Car")
        print("3 = Display Car")
        print("4 = Search Car")
        print("5 = Update Customer Payment (Cash/Cheque)")

```

```

print("6 = Return Rented Car")
print("7 = Return to Main Menu\n")

select = input("Select an option: ")

if select == '1':
    add_car()

elif select == '2':
    modify_car()

elif select == '3':
    display_car()

elif select == '4':
    search_car()

elif select == '5':
    update_payment()

elif select == '6':
    return_rented_car()

elif select == '7':
    print("Returning to Main Menu...")
    break # Exit the loop to return to the main menu

else:
    print("\nPlease select a correct option...")

# Member Customer Functionality
def member_customer_actions():
    print("** Member Customer Menu **")
    print("1 = View Bookings")
    print("2 = Update Personal Information")
    action = input("Select an option: ")

    if action == '1':
        view_bookings() # Function to display the customer's bookings
    elif action == '2':
        update_personal_info() # Function to update personal information
    else:
        print("Invalid input. Please select a valid option.")

# Non-Member Customer Functionality
def non_member_customer_actions():
    print("** Non-Member Customer Menu **")

```



```

print("1 = Rent a Car")
print("2 = Check Available Cars")
action = input("Select an option: ")

if action == '1':
    rent_car() # Function to handle the process of renting a car
elif action == '2':
    check_available_cars() # Function to display available cars
else:
    print("Invalid input. Please select a valid option.")

# Main Menu
def main_menu():
    while True:
        print("** Anjali Car Rental **")
        print("\nPlease select an option:")
        print("1 = Admin User")
        print("2 = Member Customer")
        print("3 = Non-Member Customer")
        print("4 = Exit")

        mselect = input("Select an option: ")

        if mselect == '1':
            password = input('Enter Password: ')
            if password == 'pass123':
                admin_menu()
            else:
                print("\nUser ID and Password do not match!")

        elif mselect == '2':
            member_customer_actions()

        elif mselect == '3':
            print('NON-MEMBER CUSTOMERS.')
            non_member_customer_actions()

        elif mselect == '4':
            print("Exiting...")
            break

        else:
            print("Invalid input. Please select a valid option.")

# Call the main_menu function to start the program
main_menu()

```

[ ]: