# **Online Car Rental Project**

## **Course -end Project 1**

### **IITK DS Foundations: Programming Refresher**

**Submitted by:**

**Archana Sriwastava**

**Project Title: Online Car Rental Platform**

**Project Objective:** Build an online car rental platform using Object-Oriented Programming in Python.

**Problem Statement:** A car rental company needs an online platform where customers can view available cars for rent on an hourly, daily, or weekly basis. Customers should be able to choose cars based on availability, and they will receive an auto-generated bill upon returning the car.

**Project Instructions:**

1. **Create a Module for Car Rental:**
   * Create a .py file (car\_rental.py) to define classes and functions.
   * Import the datetime module for handling rental time.
2. **Define Classes:**
   * **CarRental Class:**
     + Constructor to initialize available cars and rental records.
     + Methods to display available cars and handle renting based on hourly, daily, or weekly rates.
     + Implement checks to ensure requested cars are available.
   * **Customer Class:**
     + Constructor to manage customer details and rental requests.
     + Methods to request cars, return cars, and generate bills.
3. **Implement Car Rental Logic:**
   * **Renting Methods:**
     + Calculate rental duration and update available car inventory.
     + Record rental start time and calculate total rental cost based on chosen rental period.
   * **Returning Methods:**
     + Update inventory based on returned cars and calculate total bill including any additional charges.
4. **Main Project File (.ipynb):**
   * Import car\_rental.py module.
   * Define main execution flow using Jupyter Notebook.
   * Create instances of CarRental and Customer classes.
   * Prompt user inputs to display car availability, rent cars, or return cars.
5. **User Interaction:**
   * Display available cars to users.
   * Allow users to choose rental periods (hourly, daily, weekly) and number of cars.
   * Provide feedback on successful rental and bill generation upon return.
6. **Execution:**
   * Run the main method in Jupyter Notebook to start the project.
   * Handle user inputs dynamically to simulate car rental operations.

**Supplementary Comments:**

1. Object-Oriented Design: The project utilizes Object-Oriented Programming (OOP) principles effectively by defining classes (CarRental and Customer) that encapsulate data and functionality related to car rentals and customer interactions. This approach promotes modularity, extensibility, and code reusability.

2. User Interaction: The project prioritizes user experience by providing clear prompts and feedback during interactions. Users can easily view available cars, select rental options, and receive detailed bills upon returning cars, enhancing usability and transparency.

3. Data Handling: The project demonstrates proficiency in handling data structures such as dictionaries for managing available car inventory and lists for recording rental transactions. This ensures efficient data manipulation and retrieval throughout the rental process.

4. Time Handling: Integration of the datetime module ensures accurate recording of rental start times and calculation of rental durations, enabling precise billing based on chosen rental periods (hourly, daily, weekly).

5. Error Handling: The project includes robust error handling mechanisms to manage edge cases, such as insufficient inventory or incorrect user inputs. This ensures the stability and reliability of the platform during various user interactions.

6. Scalability and Flexibility: The project design allows for scalability by accommodating future enhancements, such as additional car types, rental options, or advanced billing features. The modular structure supports easy integration of new functionalities without disrupting existing code.

7. Documentation and Code Structure: Clear documentation within the code (car\_rental.py) and structured organization in Jupyter Notebook (main\_project.ipynb) facilitate understanding and maintenance of the project. This promotes collaboration and ease of future modifications.

8. Testing and Validation: While not explicitly covered in the submission, rigorous testing and validation of different scenarios (e.g., multiple rentals, varied rental periods) would further ensure the robustness and correctness of the implemented functionalities.

9. Conclusion: Overall, the Online Car Rental Platform project demonstrates proficiency in applying Python programming concepts, particularly OOP, to solve real-world problems effectively. It provides a foundation for developing similar scalable and user-friendly applications in various domains.