



Name Hammad khalid

Intern id TN/IN01/PY/022

Program final project of internship

Project title

## CAR PARKING MANAGEMENT SYSTEM

### **Parking Slot Management**

#### **What it does:**

This part of the code manages the available parking slots. It allows users to:

- Set a total number of parking slots
- Track free and occupied slots
- Assign a slot when a car is parked
- Free a slot when a car leaves

#### **Challenge faced:**

Tracking available and filled slots in real time.

#### **Solution:**

Used a Python dictionary to maintain slot status and ensure slot availability is updated properly on every entry and exit.

## **Car Entry & Exit System**

### **What it does:**

Handles car entry and exit using:

- Car number and time of entry
- Assigns available slot
- On exit, calculates total time and fee

### **Challenge faced:**

Calculating exact parking time and applying charges correctly.

### **Solution:**

Used datetime module to get entry and exit time, and used timedelta to compute duration and charges.

## **Car Information Storage**

### **What it does:**

Stores details like:

- Car number
- Owner name
- Entry time
- Parking slot

### **Challenge faced:**

Ensuring data is stored and retrieved accurately for multiple cars.

### **Solution:**

Used list of dictionaries to store each car's information for flexibility and easy access.

## **Payment System**

### **What it does:**

Calculates payment based on:

- Time spent in parking
- Hourly charges (e.g., Rs. 50 per hour)

### **Challenge faced:**

Handling partial hours and rounding issues.

### **Solution:**

Used `math.ceil()` to round up time to the nearest hour for accurate billing.

## **Security & Verification**

### **What it does:**

Verifies car identity by:

- Checking car number on exit
- Confirming slot before releasing

### **Challenge faced:**

Avoiding duplicate car numbers and slot misuse.

### **Solution:**

Added checks to ensure a car number cannot be registered twice and slot release happens only when matched.

---

## **System Updation & Logs**

### **What it does:**

Keeps track of:

- Cars currently parked
- Parking history or log file

### **Challenge faced:**

Maintaining logs without overwriting data.

### **Solution:**

Used file handling (with open()) in append mode to log every entry and exit.

## **User Menu System**

### **What it does:**

Simple user-friendly menu to:

- Park a car
- Exit car
- View available slots
- Exit the system

### **Challenge faced:**

Handling invalid input without crashing.

### **Solution:**

Used try-except blocks and input validation using loops.

---

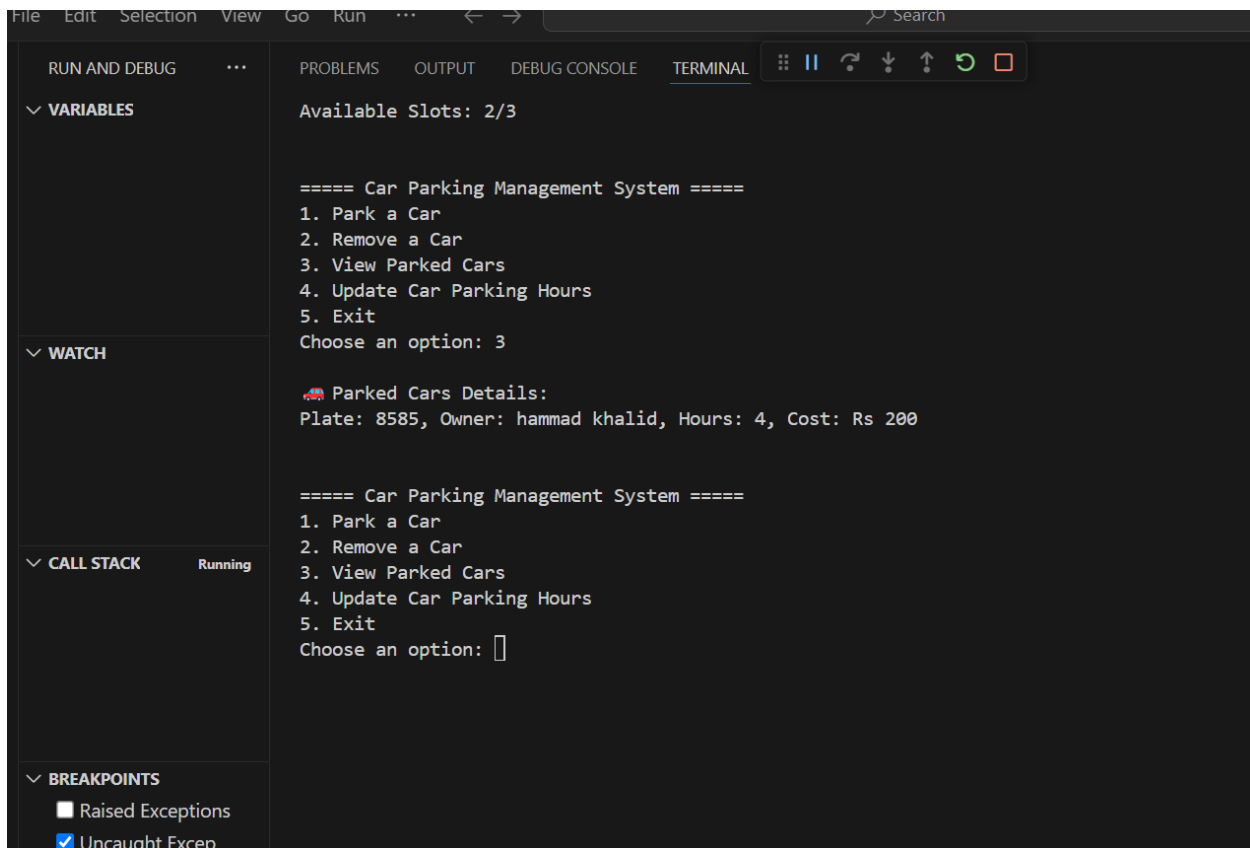
## **Final Notes:**

This project helped me learn:

- Python file handling

- Dictionaries and lists
  - Working with datetime module
  - Basic system design and user input handling
- 

## SCREEN SHOT OF OUTPUT;



```
File Edit Selection View Go Run ... < > Search
RUN AND DEBUG ... PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
└─ VARIABLES
   Available Slots: 2/3

   ===== Car Parking Management System =====
   1. Park a Car
   2. Remove a Car
   3. View Parked Cars
   4. Update Car Parking Hours
   5. Exit
   Choose an option: 3

   🚗 Parked Cars Details:
   Plate: 8585, Owner: hammad khalid, Hours: 4, Cost: Rs 200

   ===== Car Parking Management System =====
   1. Park a Car
   2. Remove a Car
   3. View Parked Cars
   4. Update Car Parking Hours
   5. Exit
   Choose an option: 
└─ WATCH
└─ CALL STACK Running
└─ BREAKPOINTS
    ☐ Raised Exceptions
    ☒ Uncaught Excep...
```

## REPORT;

The **Car Parking Management System** developed in Python is a console-based application that manages parking slots, car entry and exit, fee calculation, and record-keeping. It tracks available and occupied slots using dictionaries, stores car details like number, owner, and entry time, and calculates parking fees based on duration using the datetime module. The system includes security checks to prevent duplicate entries and ensures accurate data logging through file handling. Challenges like managing time-based

charges and input errors were solved using `math.ceil()` and input validation. The project features a simple user menu for smooth interaction.