

PROJECT  
ON

# ***PARKING LOT***



# ***MANAGEMENT***

MADE BY: SWASTIK  
SAP ID: 590027150  
BATCH: 59

A close-up, front-quarter view of an orange sports car, likely a Ford Mustang, parked on a paved surface. The car's headlights, grille, and hood are visible. The background shows green foliage and a bright sky. The image is partially obscured by a black diagonal overlay on the right side.

# ***INTRODUCTION***

**NAME: SWASTIK**  
**SAP ID: 590027150**

**PROJECT TITLE: PARKING LOT MANAGEMENT SYSTEM**  
**COURSE FACULTY: Dr. Prashant Trivedi**

**COURSE: PROGRAMMING IN C**

**UNIVERSITY: UNIVERSITY OF PETROLEUM AND  
ENERGY STUDIES**

**SCHOOL : SCHOOL OF COMPUTER SCIENCE (SOCs)**

**COMPUTER SCIENCE ENGINEERING(B-59)**

BATCH- 2025-2029

**REPOSITORY:** <https://github.com/swastik292007/parking-lot-management-project>



# ***ABSTRACT***

The initiative utilizes the C programming language for managing parking lots. The mechanism stores data on cars arriving and leaving the garage, computes charges for parking spaces, monitors overall earnings, and controls vacant spots. To achieve mastery of fundamental computer science principles including data organization through structures, manipulation via arrays, control flow using functions, storage management by files, and logical decomposition in modules. The mechanism guarantees smooth operation control for car lots while offering an intuitive dashboard access point.





# ***PROBLEM IDENTIFICATION***

Operating parking areas independently requires significant effort and increases chances of mistakes. Concerns encompass flawed documentation practices, challenges in monitoring open spots, and errors in calculating earnings. This endeavor aims at creating an automated parking lot management system capable of capturing information about vehicles including their registration numbers, types, and arrival times. Count both occupied spaces and remaining spots in all designated car lots. Permit drivers to depart by removing their vehicles while calculating fees accordingly. Ensure a comprehensive collection of all revenues. Show every car that is currently stationary in your viewfinder. Enhance precision in operations while optimizing resource utilization.

**p r o b l e m s**

# IMPLEMENTATION DETAILS

## CODE SNIPPET- ADDING VEHICLE

```
//-----VEHICLE ENTRY-----
void VehicleEntry() {
    if (filledSlots>= totalSlots) {
        printf("\nSORRY! PARKING IS FULL...\n");
        return;
    }

    struct Vehicle v;
    printf("\nEnter Vehicle Number: ");
    scanf("%s",v.number);
    printf("Enter Vehicle Type(Car/Bike/Truck): ");
    scanf("%s",v.type);
    //entering the time of entry (using my laptop's time.)
    time_t t = time(NULL);
    struct tm *tm = localtime(&t);
    sprintf(v.entryTime, "%02d:%02d", tm->tm_hour, tm->tm_min);

    vehicles[filledSlots++] = v; //adding filledslots by one after every parking
    printf("\nVEHICLE PARKED SUCCESSFULLY at %s\n",v.entryTime);
}
```



# IMPLEMENTATION DETAILS

## CODE SNIPPET-REMOVING VEHICLE

```
81 void VehicleExit() {
82     char num[20];
83     printf("\nEnter Vehicle Number to Exit: ");
84     scanf("%s", num);
85
86     int found = -1;
87     for (int i = 0; i < filledSlots; i++) {
88         if (strcmp(vehicles[i].number, num) == 0) {
89             found = i;
90             break;
91         }
92     }
93
94     if (found == -1) {
95         printf("\nVehicle not found!\n");
96         return;
97     }
98     //ENTERING EXIT TIME AND AND STORING AS VARIABLE NOW.
99     time_t now = time(NULL);
100     struct tm *tm_now = localtime(&now);
101     int exitHour = tm_now->tm_hour;
102     int exitMin = tm_now->tm_min;
103
104     int entryHour, entryMin; //ENTRY TIME
105     sscanf(vehicles[found].entryTime, "%d:%d", &entryHour, &entryMin);
106
107     int hours = exitHour - entryHour; //CALCULATING TIME OF PARKING
108     int minutes = exitMin - entryMin;
109
110     if (minutes < 0) { //BASIC BORROWING OF SUBTRACTION
111         minutes += 60;
112         hours -= 1;
113     }
114     if (hours < 0) {
115         hours += 24; // handle overnight parking
116     }
117 }
```

```
118 // Convert partial hour if needed
119 float totalHours = hours + (minutes / 60.0);
120
121 //CALCULATING FEES
122 float fee = calculateFee(vehicles[found].type, totalHours);
123 printf("\nVehicle Number: %s", vehicles[found].number);
124 printf("\nEntry Time: %s", vehicles[found].entryTime);
125 printf("\nExit Time: %02d:%02d", exitHour, exitMin);
126 printf("\nTotal Parked: %.2f hours", totalHours);
127 printf("\nParking Fee: %.2f Rupees\n", fee);
128
129 totalRevenue+=fee;
130
131 //removing the vehicle from record.
132 for (int i= found; i < filledSlots - 1; i++) {
133     vehicles[i] = vehicles[i+1];
134 }
135 filledSlots--;
136
137 printf("\nVehicle Exit Successful. Thank You!!\n");
138 }
```



# IMPLEMENTATION DETAILS

## CODE SNIPPET- VIEWING VEHICLES PARKED AND PARKING LOT STATUS

```
140 //-----DISPLAY VEHICLES-----
141 void DisplayVehicles() {
142     if (filledSlots == 0) {
143         printf("\n NO VEHICLE PARKED YET!!!\n");
144         return;
145     }
146
147     printf("\nLIST OF PARKED VEHICLES:\n");
148     printf("\n|xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx|\n");
149     printf("  NUMBER\t\tTYPE\t\tENTRY TIME\n");
150     printf("\n|xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx|\n");
151
152     for (int i=0; i<filledSlots; i++) {
153         printf("  %s\t\t%s\t\t%s\n",vehicles[i].number,vehicles[i].type,vehicles[i].entryTime);
154     }
155
156     printf("\n|xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx|\n");
157 }
158
159 //-----PARKING LOT STATUS-----
160 void ParkingLotStatus() {
161     printf("\nTotal Slots: %d",totalSlots);
162     printf("\nFilled Slots: %d", filledSlots);
163     printf("\nAvailable Slots: %d\n",totalSlots-filledSlots);
164 }
165
```



# IMPLEMENTATION DETAILS

## CODE SNIPPET- SEARCHING AND FEE CALCULATION

```
166 //-----SEARCHING VEHICLE-----
167 void Search() {
168     char num[20];
169     printf("\nEnter Vehicle Number to search: ");
170     scanf("%s",num);
171
172     for (int i = 0; i < filledSlots; i++) {
173         if (strcmp(vehicles[i].number, num) == 0) {
174             printf("\nVehicle Found!\n");
175             printf("Number: %s\nType: %s\nEntry Time: %s\n",
176                 vehicles[i].number, vehicles[i].type, vehicles[i].entryTime);
177             return;
178         }
179     }
180     printf("\nVehicle Not Found!\n");
181 }
182
183 //-----FEE CALCULATION-----
184 float calculateFee(char type[], float hours) {
185     float rate;
186     if (strcmp(type, "Car") == 0 || strcmp(type, "car") == 0)    //20 FOR CAR
187         rate = 20;
188     else if (strcmp(type, "Bike") == 0 || strcmp(type, "bike") == 0)    //10 FOR BIKE
189         rate = 10;
190     else
191         rate = 30;    //30 FOR TRUCK
192     return rate * hours;
193 }
194
```





# IMPLEMENTATION DETAILS

## CODE SNIPPET- FILE HANDLING

```
5 //-----FILE HANDLING-----
6 void saveDataToFile() {
7     FILE *fp = fopen("parkingdata.txt", "w");
8     for (int i = 0; i < filledSlots; i++) {
9         fprintf(fp, "%s %s %s\n", vehicles[i].number, vehicles[i].type, vehicles[i].entryTime);
10    }
11    fclose(fp);
12 }
13
14 void loadData() {
15     FILE *fp = fopen("parkingdata.txt", "r");
16     if (fp == NULL) return;
17     while (fscanf(fp, "%s %s %s", vehicles[filledSlots].number, vehicles[filledSlots].type, vehicles[filledSlots].entryTime) != EOF) {
18         filledSlots++;
19     }
20     fclose(fp);
21 }
22
23 //-----RUNNING THE MODULES-----
24 int main() {
25     printf("\n=====WELCOME TO PARKING-LOT-MANAGEMENT SYSTEM=====\n ");
26     loadData(); // load old records if any
27     menu();     // start the main program
28     saveDataToFile(); // save before exit
29     return 0;
30 }
```



**FILE HANDLING**

# TESTING AND RESULTS

```
=====WELCOME TO PARKING-LOT-MANAGEMENT SYSTEM=====
```

1. Vehicle Entry
2. Vehicle Exit
3. Display Parked Vehicles status
4. Parking Lot Status
5. Search vehicle
6. EXIT Program

Enter your choice(1-6): 1

Enter Vehicle Number: HR29AG2267

Enter Vehicle Type(Car/Bike/Truck): CAR

VEHICLE PARKED SUCCESSFULLY at 10:33

1. Vehicle Entry
2. Vehicle Exit
3. Display Parked Vehicles status
4. Parking Lot Status
5. Search vehicle
6. EXIT Program

Enter your choice(1-6): 3

LIST OF PARKED VEHICLES:

[illegible]

NUMBER	TYPE	ENTRY TIME
--------	------	------------

[illegible]

HR29AG2267	CAR	10:33
------------	-----	-------



1. Vehicle Entry  
2. Vehicle Exit  
3. Display Parked Vehicles status  
4. Parking Lot Status  
5. Search vehicle  
6. EXIT Program  
Enter your choice(1-6): 4

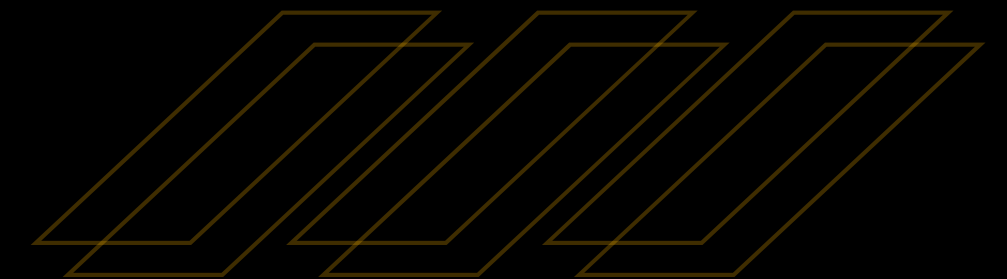
Total Slots: 50  
Filled Slots: 1  
Available Slots: 49

1. Vehicle Entry  
2. Vehicle Exit  
3. Display Parked Vehicles status  
4. Parking Lot Status  
5. Search vehicle  
6. EXIT Program  
Enter your choice(1-6): 2

Enter Vehicle Number to Exit: HR29AG2267

Vehicle Number: HR29AG2267  
Entry Time: 10:33  
Exit Time: 10:35  
Total Parked: 0.03 hours  
Parking Fee: 1.00 Rupees

# ***TESTING AND RESULTS***





# ***TESTING AND RESULTS***

1. Vehicle Entry
2. Vehicle Exit
3. Display Parked Vehicles status
4. Parking Lot Status
5. Search vehicle
6. EXIT Program

Enter your choice(1-6): 6

Total Revenue Collected: 1.00 Rupees Only.  
Thank You for using PARKING-LOT-SYSTEM!!



# ***THANK YOU***

