UPES
UNIVERSITY OF TOMORROW

PROJECT
ON

PARKING LOT MANAGEMENT

MADE BY: SWASTIK
SAP ID: 590027150
BATCH: 59

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.

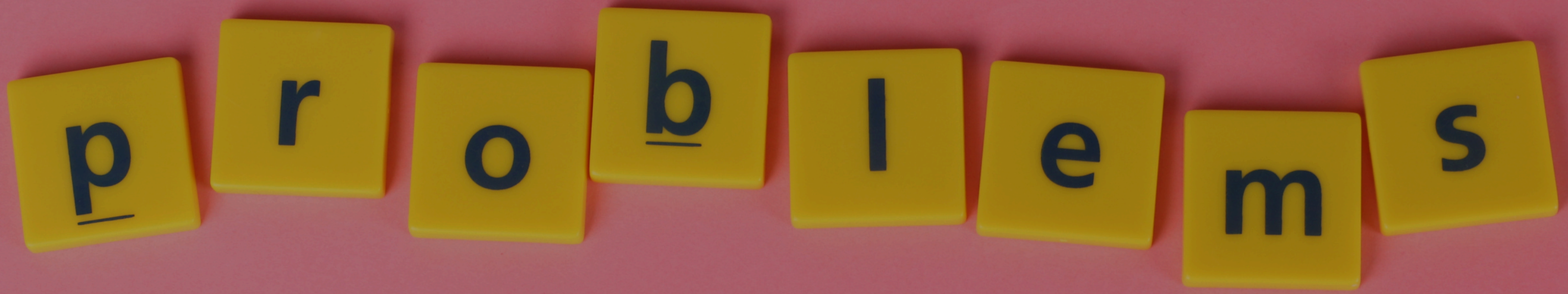
PROBLEM IDENTIFICATION

1. Operating parking areas independently(manually) requires significant effort and increases chances of mistakes.

2. Concerns encompass flawed documentation practices, challenges in monitoring open spots.

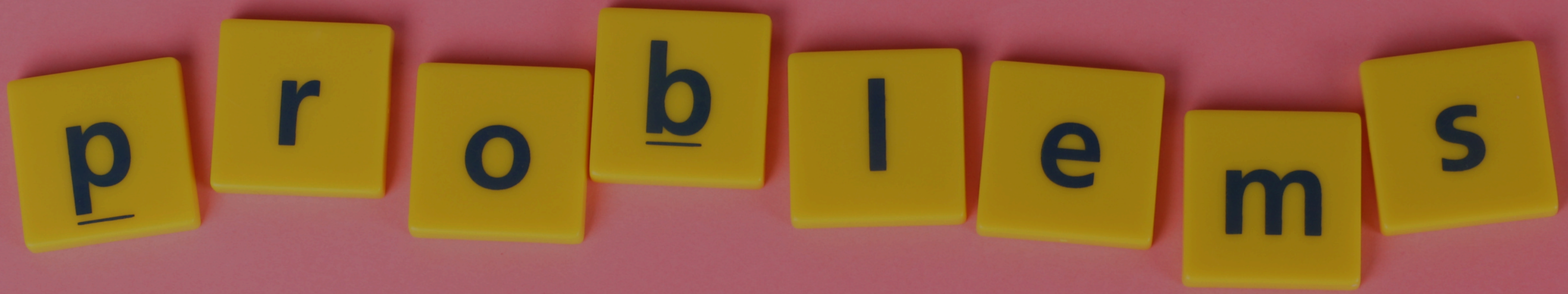
3. Errors in calculating earnings.

4. Seaching a vehicle in a parking lot is a highly extensive task the person needs to view the whole parking lot while with help of this code we can easily find if the vehicle is present or not.



SOLUTION :

- parking lot management system capable of capturing information about vehicles including their registration numbers, types, and arrival times(MAINTAING RECORD).
- 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 parking lot.
- also, created a parkingdata.txt which manages the vehicle details and removes the data when the vehicle is exited.



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

```c
void VehicleExit() {
    char num[20];
    printf("\nEnter Vehicle Number to Exit: ");
    scanf("%s", num);

    int found = -1;
    for (int i = 0; i < filledSlots; i++) {
        if (strcmp(vehicles[i].number, num) == 0) {
            found = i;
            break;
        }
    }

    if (found == -1) {
        printf("\nVehicle not found!\n");
        return;
    }
    //ENTERING EXIT TIME AND AND STORING AS VARIABLE NOW.
    time_t now = time(NULL);
    struct tm *tm_now = localtime(&now);
    int exitHour = tm_now->tm_hour;
    int exitMin = tm_now->tm_min;

    int entryHour, entryMin;  //ENTRY TIME
    sscanf(vehicles[found].entryTime, "%d:%d", &entryHour, &entryMin);

    int hours = exitHour - entryHour;  //CALCULATING TIME OF PARKING
    int minutes = exitMin - entryMin;

    if (minutes < 0) {            //(BASIC BORROWING OF SUBTRACTION)
        minutes += 60;
        hours -= 1;
    }
    if (hours < 0) {
        hours += 24; // handle overnight parking
    }
    // Convert partial hour if needed
    float totalHours = hours + (minutes / 60.0);

    //CALCULATING FEES
    float fee = calculateFee(vehicles[found].type, totalHours);
    printf("\nVehicle Number: %s", vehicles[found].number);
    printf("\nEntry Time: %s", vehicles[found].entryTime);
    printf("\nExit Time: %02d:%02d", exitHour, exitMin);
    printf("\nTotal Parked: %.2f hours", totalHours);
    printf("\nParking Fee: %.2f Rupees\n", fee);

    totalRevenue+=fee;

    //removing the vehicle from record.
    for (int i= found; i < filledSlots - 1; i++) {
        vehicles[i] = vehicles[i+1];
    }
    filledSlots--;

    printf("\nVehicle Exit Successful. Thank You!!\n");
}
```

IMPLEMENTATION DETAILS

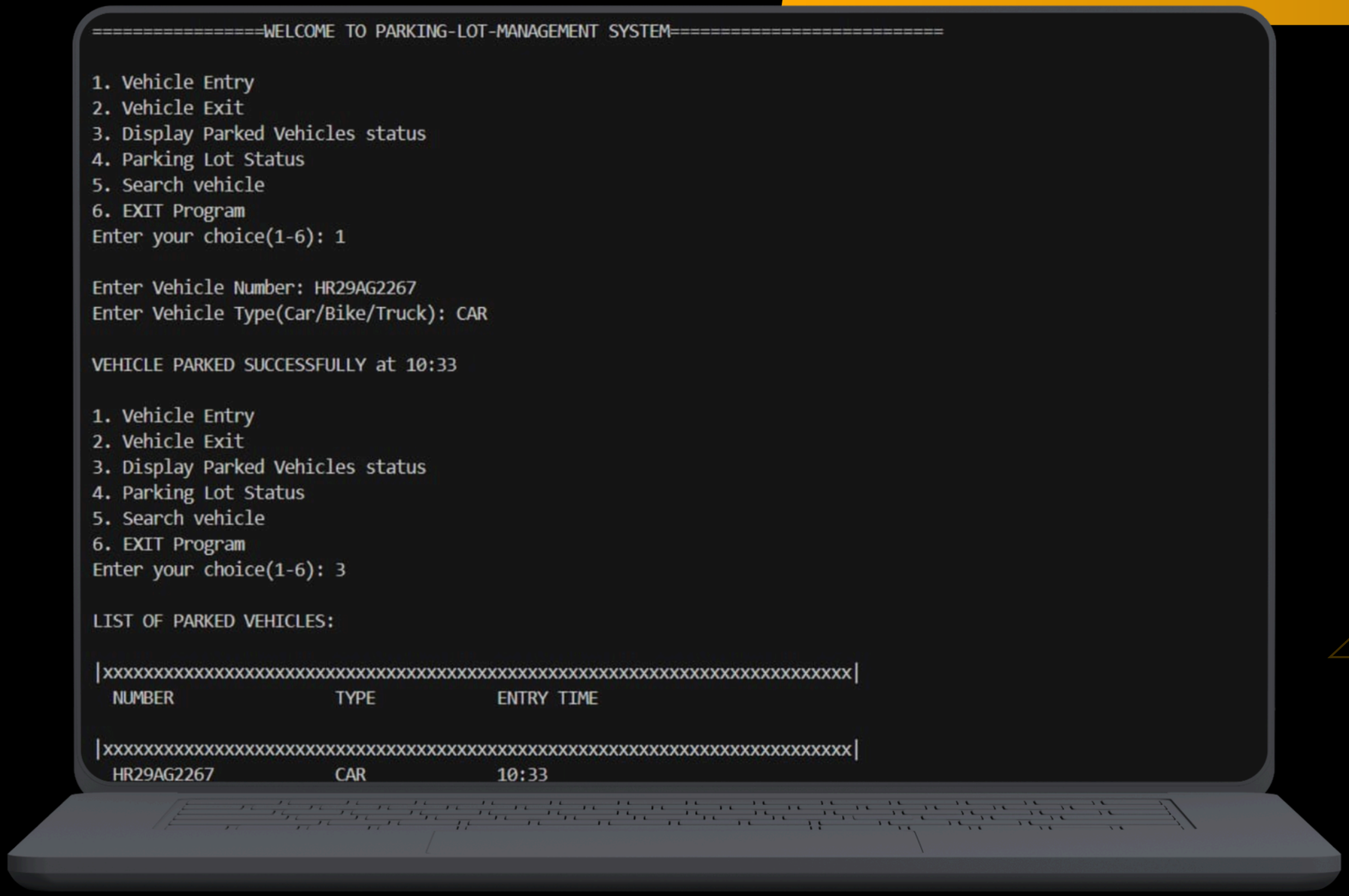
CODE SNIPPET- FILE HANDLING

```c
//------------------------------FILE HANDLING-----------------------------------------------------------------
void saveDataToFile() {
    FILE *fp = fopen("parkingdata.txt", "w");
    for (int i = 0; i < filledSlots; i++) {
        fprintf(fp, "%s %s %s\n", vehicles[i].number, vehicles[i].type, vehicles[i].entryTime);
    }
    fclose(fp);
}

void loadData() {
    FILE *fp = fopen("parkingdata.txt", "r");
    if (fp == NULL) return;
    while (fscanf(fp, "%s %s %s", vehicles[filledSlots].number, vehicles[filledSlots].type, vehicles[filledSlots].entryTime) != EOF) {
        filledSlots++;
    }
    fclose(fp);
}

//-------------------------------------------RUNNING THE MODULES-----------------------------------------------
int main() {
    printf("\n================WELCOME TO PARKING-LOT-MANAGEMENT SYSTEM==========================\n ");
    loadData();  // load old records if any
    menu();              // start the main program
    saveDataToFile();    // save before exit
    return 0;
}
```

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): 4

Total Slots: 50
Filled Slots: 1
Availaible 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
```

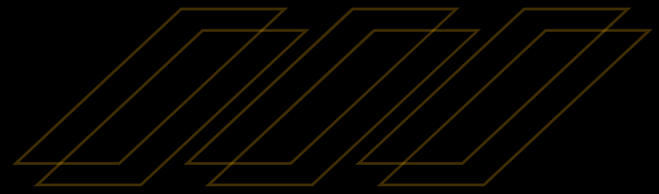
TESTING AND RESULTS

```
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!!
```

IMPROVEMENTS:

- Can introduce ev slots later.
- We can classify the slots based on type.
- We can use enum to simplify the code fee based on type.



UPES
UNIVERSITY OF TOMORROW

THANK YOU