**A**

**PROJECT REPORT**

**ON**

**“PARKING LOT MANAGEMENT SYSTEM ”**

**SUBMITTED BY:**

**Mr.HRUSHIL MAHESH SONAWANE(**2124UCEM1092)

**SUBJECT : PPS using C++**

**Under the guidance of :**

**Miss. ISHWARI TIRSE**



**Department of Computer Science and Engineering**

**Sanjivani Rural Education Society’s**

# SANJIVANI UNIVERSITY

**KOPARGAON – 423603, DIST : AHMEDNAGAR**

**2024-2025**

|  |  |  |
| --- | --- | --- |
| **SR.**  **No.** | **CONTENT** | **PAGE NO.** |
| **1.** | **INTRODUCTION** | **3** |
| **2.** | **CODE** | **4 - 7** |
| **3.** | **OUTPUT** | **8** |
| **4.** | **CONCLUSION** | **9** |

**INTRODUCTION**

A Parking Lot Management System is an IT system aimed to organize parking zones at commercial, residential and other areas. The system therefore seeks to improve the user experience through the performance of major tasks including entry/exit of the vehicle, space allocation and payment. It enables tracking of parking spot availability using sensors, cameras and software applications, efficient space management and minimizes instances where people have to monitor the parking lots.

Besides, this system enhances the quality of traffic flow, saves time for searching the place to park a car, and increases security. For parking lot operators, it offers many dimensional usages, which makes the management decision and raise the amount of income by the parking lots. Parking management solutions are becoming something very important as the roles of parking are increasing due to the growing population and the improving means of transport.

**CODE**

#include <iostream>

#include <vector>

Using namespace std;

Class Vehicle {

Public:

String licensePlate;

String vehicleType;

Vehicle(string license, string type) : licensePlate(license), vehicleType(type) {}

};

Class ParkingSpot {

Public:

Int spotNumber;

Bool isOccupied;

Vehicle\* vehicle;

ParkingSpot(int number) : spotNumber(number), isOccupied(false), vehicle(nullptr) {}

Void parkVehicle(Vehicle\* v) {

isOccupied = true;

vehicle = v;

cout << “Vehicle with license plate “ << v->licensePlate << “ parked at spot “ << spotNumber << endl;

}

Void unParkVehicle() {

Cout << “Vehicle with license plate “ << vehicle->licensePlate << “ left spot “ << spotNumber << endl;

isOccupied = false;

vehicle = nullptr;

}};

Class ParkingLot {

Public:

Int totalSpots;

Vector<ParkingSpot\*> spots;

ParkingLot(int size) : totalSpots(size) {

For (int i = 1; i <= totalSpots; i++) {

Spots.push\_back(new ParkingSpot(i));

}

}

Void park(Vehicle\* v) {

For (auto& spot : spots) {

If (!spot->isOccupied) {

Spot->parkVehicle(v);

Return;

}

}

Cout << “Parking full! No spots available.” << endl;

}

Void unPark(string licensePlate) {

For (auto& spot : spots) {

If (spot->isOccupied && spot->vehicle->licensePlate == licensePlate) {

Spot->unParkVehicle();

Return;

}

}

Cout << “Vehicle with license plate “ << licensePlate << “ not found!” << endl;

}

Void displayAvailableSpots() {

Cout << “Available spots: “;

Bool hasSpots = false;

For (auto& spot : spots) {

If (!spot->isOccupied) {

Cout << spot->spotNumber << “ “;

hasSpots = true;

} }

If (!hasSpots) {

Cout << “No spots available.”;

}

Cout << endl;

}};

Int main() {

Int totalSpots, choice;

Cout << “Enter total number of parking spots: “;

Cin >> totalSpots;

ParkingLot lot(totalSpots);

While (true) {

Cout << “\nMenu:\n”;

Cout << “1. Park a Vehicle\n”;

Cout << “2. Unpark a Vehicle\n”;

Cout << “3. Display Available Spots\n”;

Cout << “4. Exit\n”;

Cout << “Enter your choice: “;

Cin >> choice;

If (choice == 1) {

String licensePlate, vehicleType;

Cout << “Enter Vehicle License Plate: “;

Cin >> licensePlate;

Cout << “Enter Vehicle Type (Car/Bike/etc.): “;

Cin >> vehicleType;

Vehicle\* vehicle = new Vehicle(licensePlate, vehicleType);

Lot.park(vehicle);

} else if (choice == 2) {

String licensePlate;

Cout << “Enter Vehicle License Plate to unpark: “;

Cin >> licensePlate;

Lot.unPark(licensePlate);

} else if (choice == 3) {

Lot.displayAvailableSpots();

} else if (choice == 4) {

Cout << “Exiting... Goodbye!” << endl;

Break;

} else {

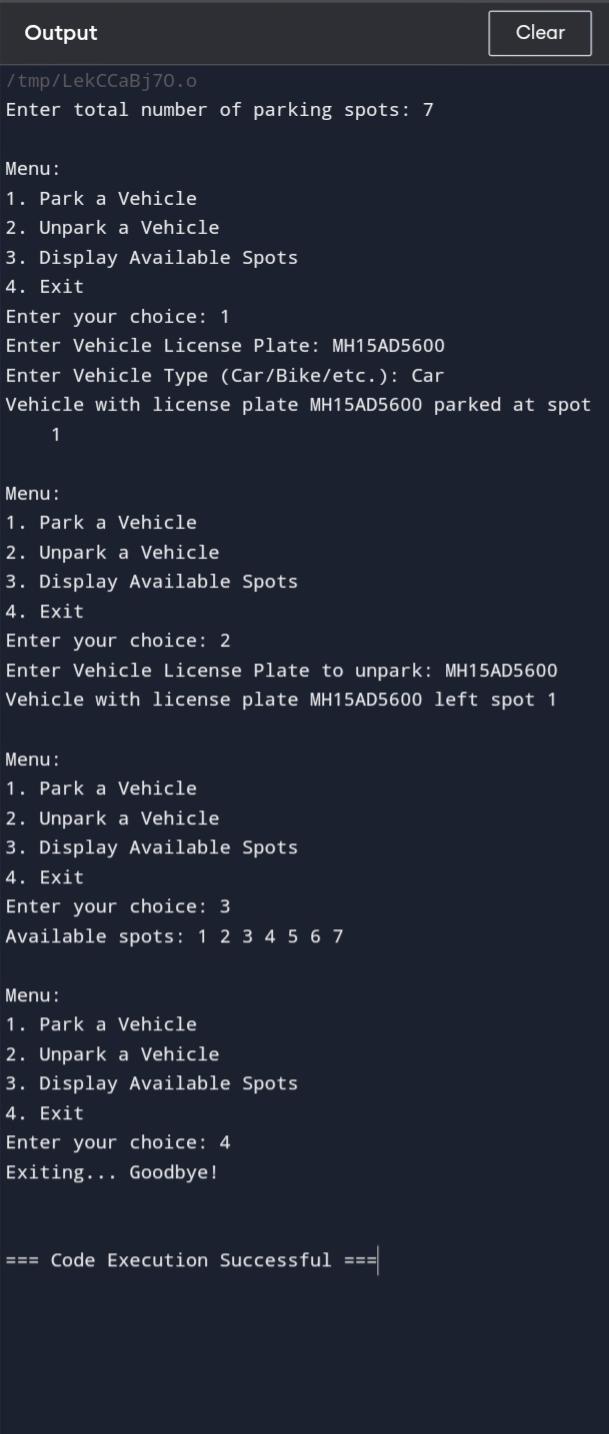
Cout << “Invalid choice! Please try again.” << endl;

} }

return 0;

}

**OUTPUT**

****

**CONCLUSION**

In conclusion, this paper has found that a Parking Lot Management System is indispensable to integrate and implement innovation of parking services and facilities for both the users and operators. Using automation, monitoring, and the efficient system of payment, it drastically cuts down the traffic, maximizes the available areas, and boosts security. It not only increases user satisfaction by easing the process of searching for parking spaces but also provides important information as well as analysis to help operators generate profit and enhance their management efficiency. In the course of the development of urban centers and expansion of avenues into mega cities, such systems will be vital to any city to have efficient infrastructure.