NAME : ROHITH P

REG-NO : 231401086

OOPS WITH JAVA

MINI PROJECT : PARKING MANAGEMENT SYSTEM

PROGRAM:

```
import org.json.JSONArray;
import org.json.JSONObject;

import java.io.FileWriter;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.Scanner;

class ParkingSpot {
    int spotId;
    String vehicleNumber;
    boolean isOccupied;

    public ParkingSpot(int spotId) {
        this.spotId = spotId;
        this.vehicleNumber = "";
        this.isOccupied = false;
    }

    public void assignVehicle(String vehicleNumber) {
        this.vehicleNumber = vehicleNumber;
        this.isOccupied = true;
    }

    public void removeVehicle() {
        this.vehicleNumber = "";
        this.isOccupied = false;
    }

    @Override
    public String toString() {
        return "Spot ID: " + spotId + ", Vehicle: " + (isOccupied ? vehicleNumber
: "Empty");
    }

    public JSONObject toJSON() {
        JSONObject json = new JSONObject();
        json.put("spotId", spotId);
        json.put("vehicleNumber", vehicleNumber);
        json.put("isOccupied", isOccupied);
        return json;
```

```
    }

    public static ParkingSpot fromJSON(JSONObject json) {
        ParkingSpot spot = new ParkingSpot(json.getInt("spotId"));
        spot.vehicleNumber = json.getString("vehicleNumber");
        spot.isOccupied = json.getBoolean("isOccupied");
        return spot;
    }
}

public class ParkingManagementSystem {
    private ArrayList<ParkingSpot> parkingSpots;
    private final String DATA_FILE = "parking_data.json";

    public ParkingManagementSystem(int totalSpots) {
        parkingSpots = new ArrayList<>();
        for (int i = 1; i <= totalSpots; i++) {
            parkingSpots.add(new ParkingSpot(i));
        }
        loadData();
    }

    public void parkVehicle(String vehicleNumber) {
        for (ParkingSpot spot : parkingSpots) {
            if (!spot.isOccupied) {
                spot.assignVehicle(vehicleNumber);
                System.out.println("Vehicle parked at spot ID: " + spot.spotId);
                saveData();
                return;
            }
        }
        System.out.println("No available spots.");
    }

    public void removeVehicle(int spotId) {
        if (spotId > 0 && spotId <= parkingSpots.size()) {
            ParkingSpot spot = parkingSpots.get(spotId - 1);
            if (spot.isOccupied) {
                spot.removeVehicle();
                System.out.println("Vehicle removed from spot ID: " +
spot.spotId);
                saveData();
            } else {
                System.out.println("Spot is already empty.");
            }
```

```
        } else {
            System.out.println("Invalid spot ID.");
        }
    }

    public void displayStatus() {
        System.out.println("Parking Lot Status:");
        for (ParkingSpot spot : parkingSpots) {
            System.out.println(spot);
        }
    }

    public void saveData() {
        JSONArray jsonArray = new JSONArray();
        for (ParkingSpot spot : parkingSpots) {
            jsonArray.put(spot.toJSON());
        }

        try (FileWriter file = new FileWriter(DATA_FILE)) {
            file.write(jsonArray.toString(4)); // Indented with 4 spaces
        } catch (IOException e) {
            System.out.println("Error saving data: " + e.getMessage());
        }
    }

    public void loadData() {
        try {
            String content = new
String(Files.readAllBytes(Paths.get(DATA_FILE)));
            JSONArray jsonArray = new JSONArray(content);

            parkingSpots.clear();
            for (int i = 0; i < jsonArray.length(); i++) {
                parkingSpots.add(ParkingSpot.fromJSON(jsonArray.getJSONObject(i))
);
            }
        } catch (IOException e) {
            System.out.println("No previous data found. Starting fresh.");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ParkingManagementSystem system = new ParkingManagementSystem(10);
```

```
        while (true) {
            System.out.println("\n1. Park Vehicle");
            System.out.println("2. Remove Vehicle");
            System.out.println("3. Display Status");
            System.out.println("4. Exit");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter vehicle number: ");
                    String vehicleNumber = scanner.next();
                    system.parkVehicle(vehicleNumber);
                    break;
                case 2:
                    System.out.print("Enter spot ID to remove vehicle: ");
                    int spotId = scanner.nextInt();
                    system.removeVehicle(spotId);
                    break;
                case 3:
                    system.displayStatus();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid option. Try again.");
            }
        }
    }
}
```