PROJECT TITLE

Mini Project Report

Object Oriented Programming (01CE0104)

Semester 2

Student Name – Khushaliba Rathod

Enrollment Number- 92400103533

Marwadi University

Department of Computer Engineering

Rajkot, Gujarat

April, 2025

Project Description:

Car Parking System is a console-based Java application developed using Object-Oriented Programming principles. The project aims to simulate a simple parking management system where users can enter, view, and reset the count of vehicles parked.

The system provides a menu-driven interface that allows users to manage the parking of different types of vehicles such as cars, bikes, and rickshaws. Users can input the type of vehicle entering the parking, view the current number of each type of vehicle, or reset all values when needed.

The core functionality is encapsulated within two classes: `Carparking` and `parking`. The `Carparking` class handles user input and the interface logic, while the `parking` class manages the state of the parking lot using private variables and accessor methods. This separation of concerns showcases good object-oriented design.

This project is ideal for learning basic OOP principles such as encapsulation, class interaction, and input handling. It can be further enhanced with features like time-based billing, vehicle number registration, and GUI integration for a complete real-world simulation.

The application provides features like customizable budget planning, automated expense tracking, and visual reports such as pie charts and bar graphs to illustrate spending habits and trends. Users can set monthly spending limits, receive notifications when they approach or exceed their budgets, and track their progress toward financial goals such as saving for a vacation, paying off debt, or building an emergency fund.

Data security and privacy are prioritized, ensuring that all financial records are safely stored and accessible only to the user. The tracker can also generate summaries and downloadable reports for better financial planning and analysis.

Car Parking System is a console-based Java application developed using Object-Oriented Programming principles. The project aims to simulate a simple parking management system where users can enter, view, and reset the count of vehicles parked.

The system provides a menu-driven interface that allows users to manage the parking of different types of vehicles such as cars, bikes, and rickshaws. Users can input the type of vehicle entering the parking, view the current number of each type of vehicle, or reset all values when needed.

The core functionality is encapsulated within two classes: `Carparking` and `parking`. The `Carparking` class handles user input and the interface logic, while the `parking` class manages the state of the parking lot using private variables and accessor methods. This separation of concerns showcases good object-oriented design.

This project is ideal for learning basic OOP principles such as encapsulation, class interaction, and input handling. It can be further enhanced with features like time-based billing, vehicle number registration, and GUI integration for a complete real-world simulation.

Source Code:

Carparking.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package carparking;

import java.util.Scanner;

/**
 *
 * @author Lenovo1
 */
public class Carparking {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here

        parking par =new parking();
```

```
Scanner sc=new Scanner(System.in);
int choice;
int car=0,bike=0,rikshow=0;

while(true)
{
System.out.println("press 1 to enter car");
System.out.println("press 2 to enter bike");
System.out.println("press 3 to enter rikshow");
System.out.println("press 4 to view data");
System.out.println("press 5 to delete data");
System.out.println("press 6 to exit");
choice=sc.nextInt();
switch (choice)
{
   case 1:
      car++;
      par.setcar(car);
      System.out.println(car+"cars is added");
      break;
   case 2:
      bike++;
```

```
                par.setbike(bike);
                System.out.println(bike+"bikes is added");
                break;
            case 3:
                bike++;
                par.setrikshow(rikshow);
                System.out.println(rikshow+"rikshows is added");
                break;
            case 4:
                System.out.println("\n cars:"+par.getcar());
                System.out.println("\n bikes:"+par.getbike());
                System.out.println("\n rikshows:\n "+par.getrikshow());

        }

    }

    }}
```

parking.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Other/File.java to edit this template
```

```
 */
package carparking;


/**
 *
 * @author Lenovo
 */
public class parking {


    /**
     * @param args the command line arguments
     */
   private int car,bike,rikshow;
    parking()
    {
        car=0;
        bike=0;
        rikshow=0;
    }
    public int getcar(){
        return car;
    }
    public void setcar(int car){
```

```
    this.car=car;
  }
  public int getbike(){
     return bike;
  }
  public void setbike(int bike){
     this.bike=bike;
  }
  public int getrikshow(){
     return rikshow;
  }
  public void setrikshow(int rikshow){
     this.rikshow=rikshow;
  }}
```

Output Screenshots:

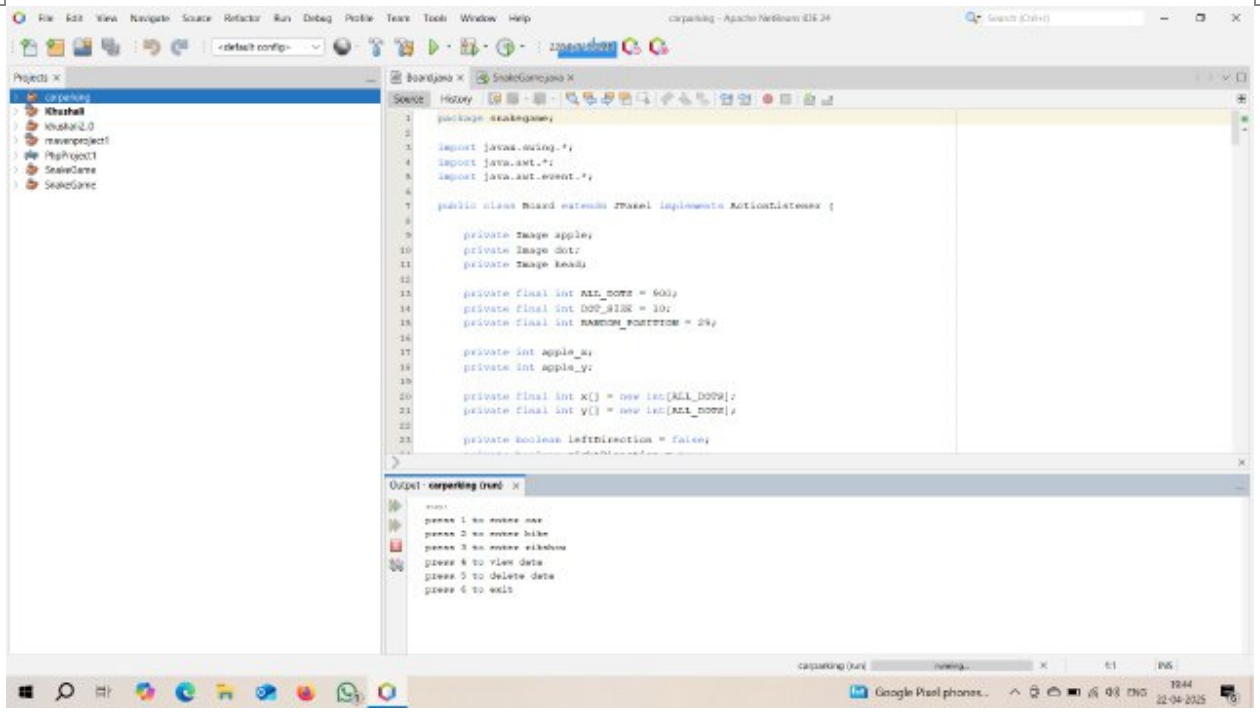

Figure 1.1 Menu Display

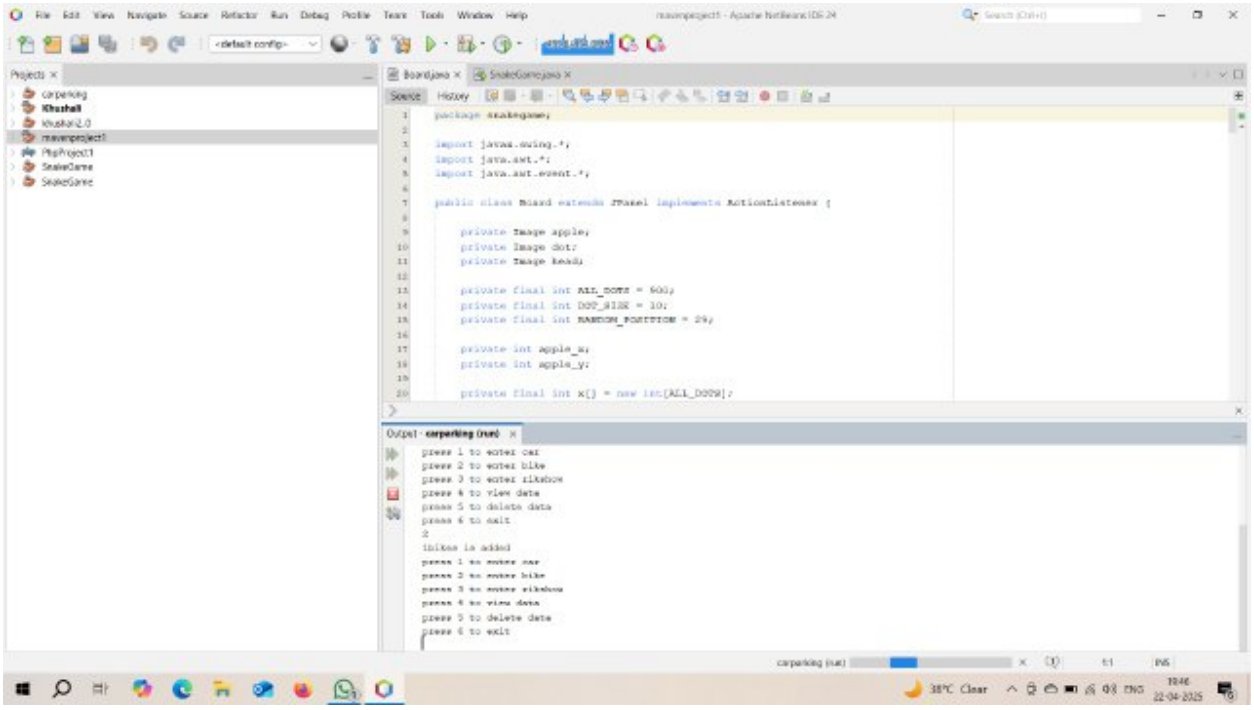

Figure 1.2 When Vehicles Added

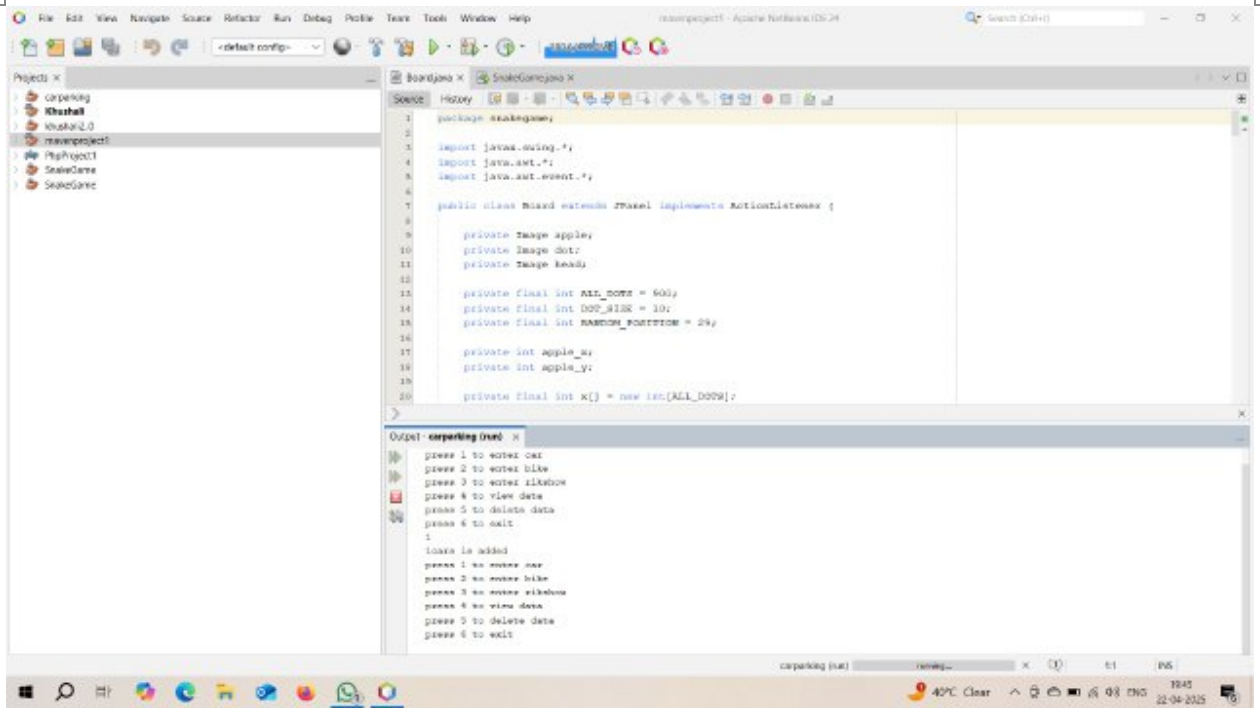

Figure 1.3 bikes added

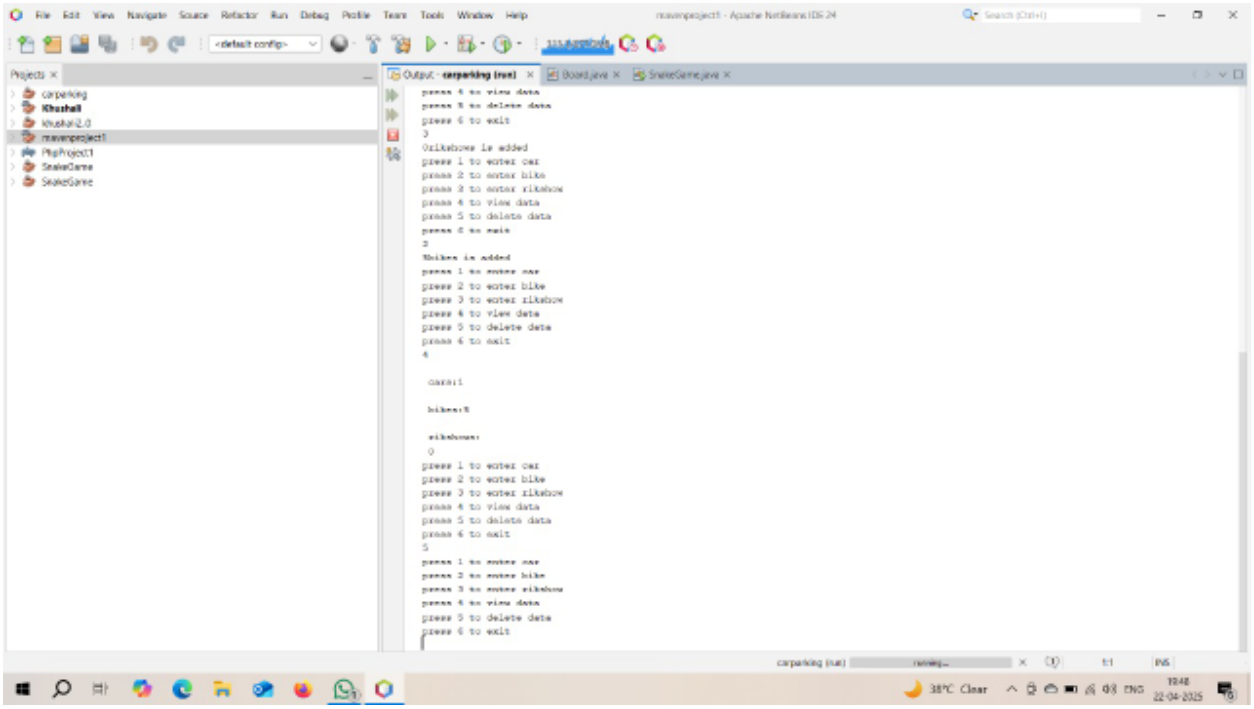

Figure 1.4 View parking

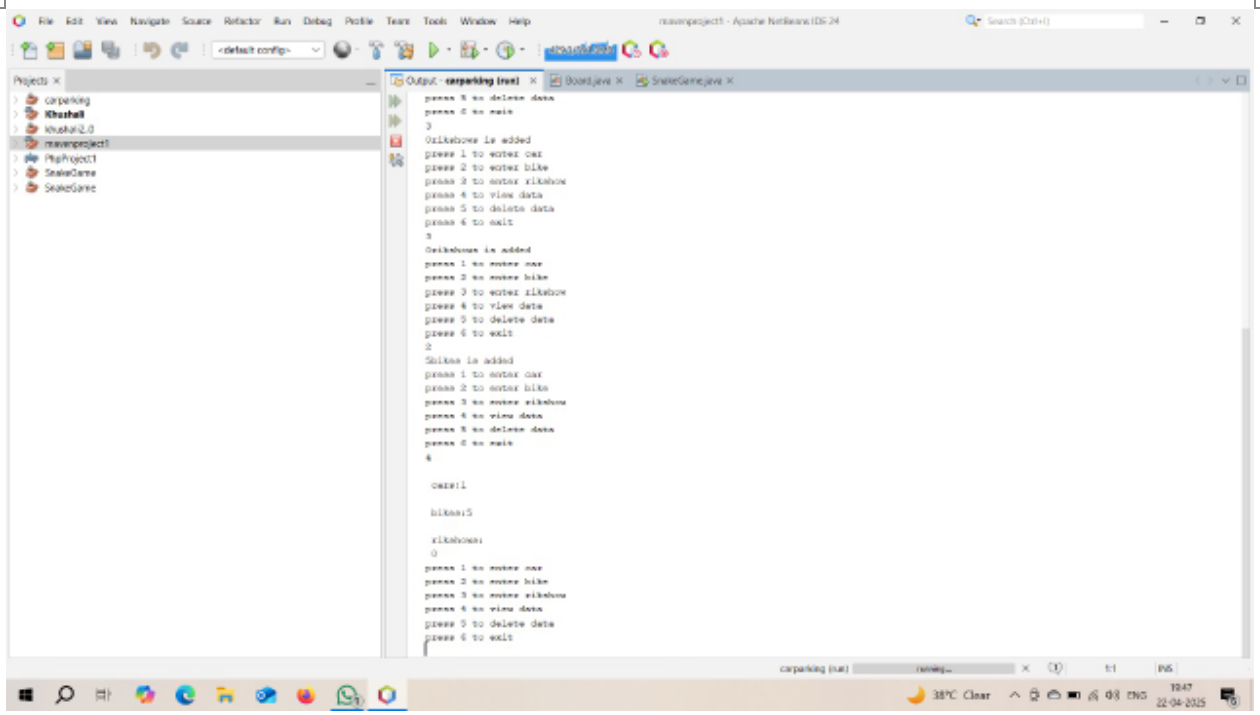

Figure 1.5 More Parking Added

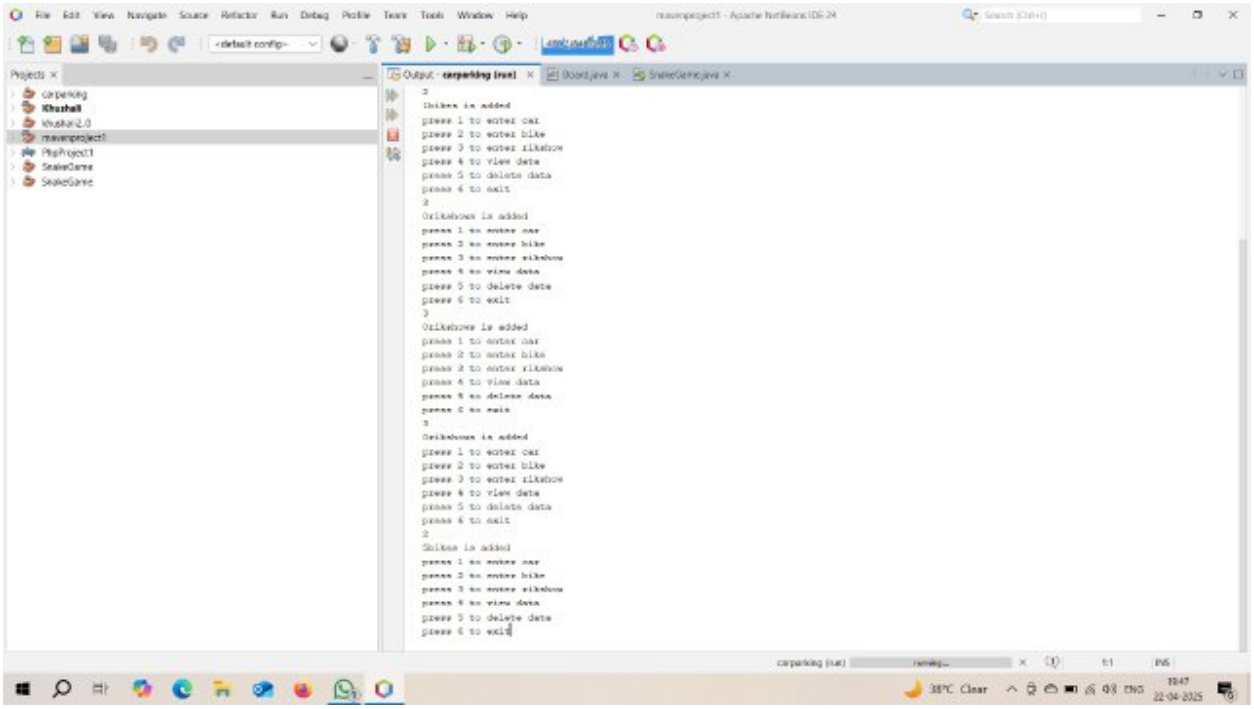

Figure 1.6 Exit