

**A  
Project Report  
On  
“Vehicle Parking System”**

**Submitted to Savitribai Phule Pune University for**



**Maharashtra Education Society's  
Institute of Management and Career Courses, Pune  
For  
ITC11 - Mini Project  
MCA SEM-III**

**Guide Name:**

Prof. Mrs. D. Belsare.

Prof. Mrs. Manasi. Shirurkar

**Submitted By:**

Jayesh Bagad (2011010)

Pradip Shirsath (2012127)

## **ACKNOWLEDGEMENT**

We thank the people who were a part of this project in numerous ways, people who gave their unending support right from the stage the project idea was conceived.

Any Project requires the guidance, hard work, contribution of time along with well-organized and well-planned efforts by many people. It is a very difficult task to acknowledge all those who have been helping in our academic project work.

I would like to thank our Dr. Ravindra Vaidya (HOD of MCA) who has always been the source of inspiration and we also thankful to Prof. Mrs. Manasi Shirurkar our co-ordinator for all the help she has rendered to ensure the successful completion of the project.

we take this opportunity to offer sincere thanks to **Prof. Mrs. Dipti Belsare** who was very much kind enough to give us an idea and guide us throughout our project work.

Last, we would like to thank all our friends and other teaching staff for their supports.

# INDEX

Sr No.	Topic
1.	Introduction
	a. Overview
	I. Existing System
	II. Need of the System
	III. Overview of the Project
	b. Hardware and Software Requirement
2.	System Development Lifecycle
	a. Requirements from User
	I. End User Requirements
	II. Administrator Requirements
	b. Feasibility Study
	I. Technical Feasibility
	II. Operational Feasibility
	III. Economic Feasibility
	c. Table Design
	d. Diagrams
	I. Class Diagram
	II. Use Case Diagram
	III. Activity Diagram
	e. Implementation
	I. Input Screens
	II. Output Screens
3.	Bibliography

# **1.Introduction**

Vehicle Parking system for managing the records of the incoming and outgoing vehicles in a public parking area.

It's an easy for Admin to retrieve the data if the vehicle has been visited through number, he can get that data.

Now days in many public places such as malls, multiplex system, hospitals, offices, market areas in big cities there is a crucial problem of vehicle parking. The vehicle parking area has many lanes/slots for parking. So, to park a vehicle one has to look for all the slots. Moreover, this involves a lot of manual labour and investment. Instead of vehicle caught in towing the vehicle can park on safe and security with low cost.

The Purpose of this project is to track and manage occupancy of parking Slots and allow customers to find and reserve available parking places. The parking lots currently operates with any manual system and also by computerized system. Sometimes there may be a problem to find a particular Vehicle. So, to solve this problem we design an online web base Parking Management System.

The objective of this project is to build a Parking Management System that enables the time management and control of vehicles using number of vehicles recognition. The system that will track the entry and exit of vehicles, maintain a listing of vehicles within the parking lot, and determine if the parking lot is full or not. It will determine the cost of per vehicle according to their time consumption.

# **1.Overview**

## **Existing System:**

The current system of parking lot is both computerized and manual also. Requires manpower to manage the parking system. Sometimes it may become difficult to find the particular vehicle in large numbers of vehicles in a big Parking area.

All the Records are Stored manually so it is difficult to manage the record files. In case of finding the specific record, it is very difficult to find the record in the registered file. This may also cause wastage of papers as well as get extra time to find our vehicles.

## **Need of the System:**

Present working of Parking Lot is manual and as well computerized. The information in many registers and on the papers which may cause unavailability of information. If in case, records are lost, it will cause problem in managing data. Keeping all records in registers also affects the secrecy of all business-related data. Anyone can easily access the private

information from register. Making all calculations manually managing all payment related information is also an overhead.

By making the system online web-based application-Oriented we can maintain all the records easily in a systematic and presentable way. With help of computerized system any kind of record will be stored in system and large amount of data can be handled.

Different features of system will make faster availability of record and details related vehicles. An application Oriented System is an easy to handle, easy to maintain and systematic way of recording the necessary information. Also, the user/customers will get the benefits like save their time, don't need to look for free parking lots.

### **Overview of Project:**

In the modern age, many people have vehicles. Vehicle is now a basic need. Everyplace is under the process of urbanization. There are many corporate offices and shopping centers, malls etc. There are many recreational places where people go for work. So, all these places need a parking area/space where people park their vehicles safely and easily. Every parking area

needs a system that records the detail of each vehicle to give the facility. These systems might be computerized or non-computerized. With the help of web-based system we can deliver a good service to customer who wants to park their vehicle into the reserved slots of parking areas.

Parking system is a web-based online system which delivers data processing in very high speed in systematic manner. Parking is a growing need of the time. Development of this system is very useful in this area of field. We can sell this system to any organization. By using our system, they can maintain records very easily. Our system covers every area of parking management. In coming future there will be excessive need of vehicle Parking system.

Customers can register with their details in the system. They can search free parking slots through their vehicle types like two-wheelers, four-wheelers. Customers are provided with full information with receipt for their parking slots.

## **2.Hardware and Software Requirement**

### **Software Requirement:**

- Frontend: Python
- Backend: SQL lite
- Database: SQL lite
- Operating System: Windows

### **Hardware Requirement:**

- Ram: 2GB
- Processor: Dual Core
- Storage: 256GB



## **2. System Development Lifecycle**

### **1.Requirements from Users**

#### **1. End User Requirements:**

- Need for an application that makes communicating easy and comfortable.
- An application that enables user/owner to park a vehicle with safe and secure.
- Need for an application that is easy to use and widely available and hence a web application
- Handling all functions done with organization in a computerized manner. Allowing the user to park the vehicle directly.
- Users should be able to use the web-based application from any web browser.

## **2. Administrator Requirements:**

- Admin need to enter all details for registration.
- Admin need to insert all details about customer and vehicle.
- Admin need to save all the details of customer and vehicle.
- Admin can retrieve the details of customer.
- Admin must generate a report for payment.
- Administrators should be able to view all user transactions.
- Site managers should be able to view all transactions for the day.

## **2.Feasibility Study**

### **1. Technical Feasibility Study:**

- This web-based application system is designed in such way that any of its user can easily handle the system or interact with it. The user of system need not to have any higher or technical knowledge about the computer to interact with system.
- Our project is technically feasible because we develop this project as a web application there are finite number of software and hardware are required. Also, the maintenance of the project is also technically feasible, it requires limited recourses which are readily available or they are already present almost everywhere.

### **2. Operational Feasibility Study:**

- It is a study to find out whether developed and implemented system will be useful or not? It is for finding out whether the developed system would comply with the users or will user's resist of using this system.

### **3. Economic Feasibility:**

- The project is economically feasible as the only cost involved is having a computer with the minimum requirements mentioned earlier. For the users to access the application, the only cost involved will be in getting access to the Internet.

### 3.Table Design

#### Vehicle:

Field	Data Type	Description	Key	Size
id	Int	Admin ID	PK	5
parkingnumber	Varchar	Parking Number		20
vehiclecompany	Varchar	Vehicle Name		20
regno	Int	Vehicle Number		20
ownername	Varchar	Vehicle owner Name		20
ownercontact	Varchar	Contact Number		10
pdate	Date	Parking Date		10
Intime	Varchar	Vehicle Intime		10
outtime	Varchar	Vehicle Out Time		10
parkingcharge	int	Parking fees/rate		10
status	Varchar	Out/In		10
Category id	int	Vehicle type category		10

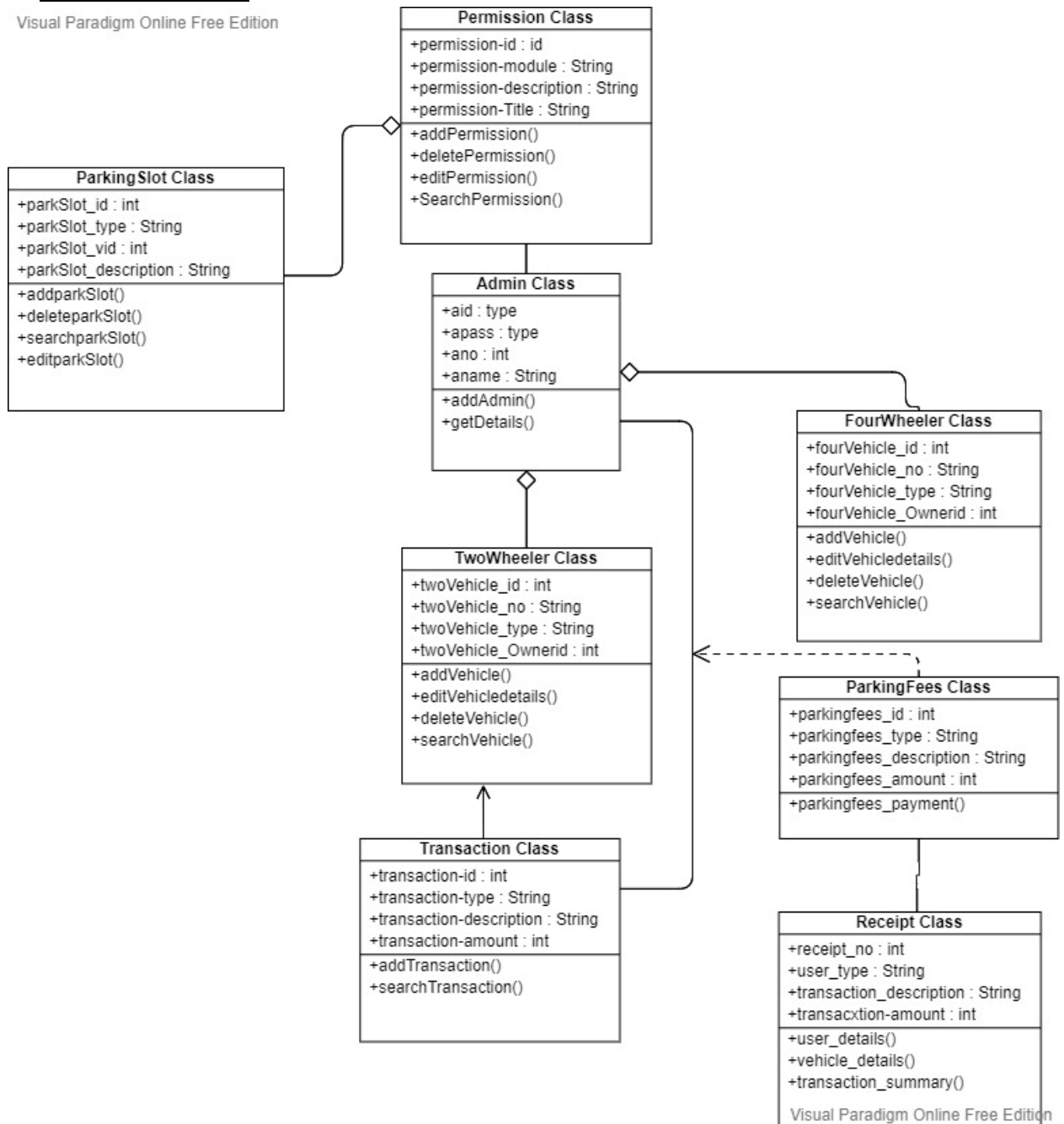
**Category:**

Field	Data Type	Description	Key	Size
id	Int	Vehicle Id	Pk	5
categoryname	Varchar	Vehicle Category Name		20
creationdate	Date	Vehicle Add Date		10
Adminname	Varchar	Admin Name		20
password	Varchar	Password		20

## 4. Project Related Diagrams:

### 1. Class Diagram:

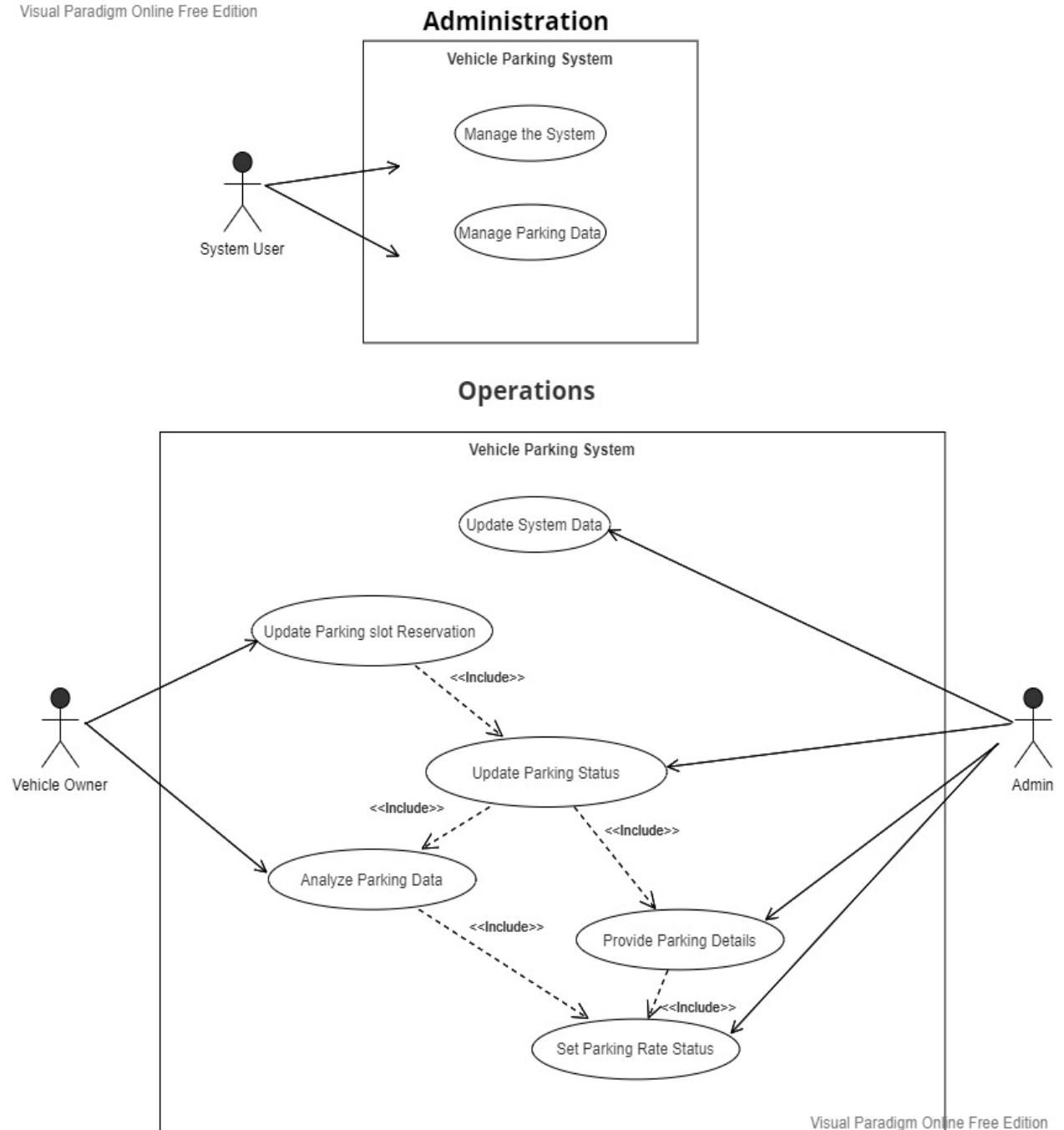
Visual Paradigm Online Free Edition



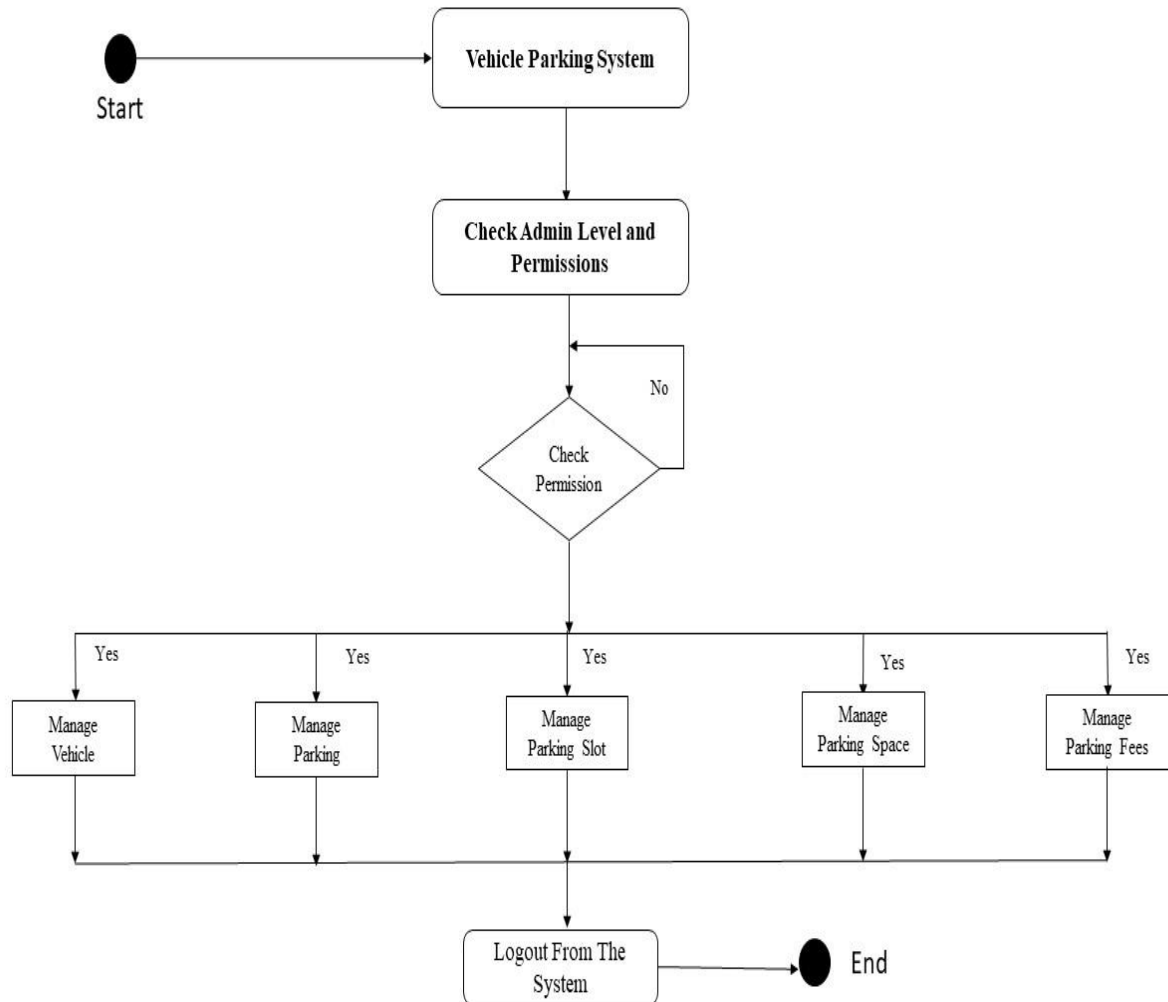
Visual Paradigm Online Free Edition

## 2. Use Case Diagram:

Visual Paradigm Online Free Edition



### 3. Activity Diagram:





## 5. Implementation

### Input Screens:

```
from django.db.models import Q
from django.shortcuts import render, redirect
from .models import *
from django.contrib.auth.models import User
from django.contrib.auth import login, logout, authenticate
from django.contrib import messages
from datetime import date
from datetime import datetime, timedelta, time
import random

# Create views here.

def Index(request):
    return render(request, 'index.html')

def about(request):
    return render(request, 'about.html')

def contact(request):
    return render(request, 'contact.html')

def admin_login(request):
    error = ""
    if request.method == 'POST':
        u = request.POST['username']
        p = request.POST['password']
```

```
user = authenticate(username=u, password=p)

try:
    if user.is_staff:
        login(request, user)
        error = "no"
    else:
        error = "yes"
except:
    error = "yes"
d = {'error': error}
return render(request, 'admin_login.html', d)
```

```
def admin_home(request):
```

```
    if not request.user.is_authenticated:
```

```
        return redirect('admin_login')
```

```
    today = datetime.now().date()
```

```
    yesterday = today - timedelta(1)
```

```
    lasts = today - timedelta(7)
```

```
    tv = Vehicle.objects.filter(pdate=today).count()
```

```
    yv = Vehicle.objects.filter(pdate=yesterday).count()
```

```
    ls = Vehicle.objects.filter(pdate__gte=lasts, pdate__lte=today).count()
```

```
    totalv = Vehicle.objects.all().count()
```

```
    d = {'tv': tv, 'yv': yv, 'ls': ls, 'totalv': totalv}
```

```
    return render(request, 'admin_home.html', d)
```

```
def Logout(request):
```

```
    logout(request)
```

```
return redirect('index')
```

```
def change_password(request):
```

```
    if not request.user.is_authenticated:
```

```
        return redirect('admin_login')
```

```
    error = ""
```

```
    if request.method == "POST":
```

```
        o = request.POST['password']
```

```
        n = request.POST['newpassword']
```

```
        try:
```

```
            u = User.objects.get(id=request.user.id)
```

```
            if u.check_password(o):
```

```
                u.set_password(n)
```

```
                u.save()
```

```
                error = "no"
```

```
            else:
```

```
                error = "not"
```

```
        except:
```

```
            error = "yes"
```

```
    d = {'error': error}
```

```
    return render(request, 'change_password.html', d)
```

```
def add_category(request):
```

```
    if not request.user.is_authenticated:
```

```
        return redirect('admin_login')
```

```
    error = ""
```

```
    if request.method == "POST":
```

```
        cn = request.POST['categoryname']
```

```
        try:
```

```
        Category.objects.create(categoryname=cn)

        error = "no"

    except:

        error = "yes"

    d = {'error': error}

    return render(request, 'add_category.html', d)
```

```
def manage_category(request):

    if not request.user.is_authenticated:

        return redirect('admin_login')

    category = Category.objects.all()

    d = {'category': category}

    return render(request, 'manage_category.html', d)
```

```
def delete_category(request, pid):

    if not request.user.is_authenticated:

        return redirect('admin_login')

    category = Category.objects.get(id=pid)

    category.delete()

    return redirect('manage_category')
```

```
def edit_category(request, pid):

    if not request.user.is_authenticated:

        return redirect('admin_login')

    category = Category.objects.get(id=pid)

    error = ""

    if request.method == 'POST':

        cn = request.POST['categoryname']

        category.categoryname = cn
```

```

    try:
        category.save()
        error = "no"
    except:
        error = "yes"
    d = {'error': error, 'category': category}
    return render(request, 'edit_category.html', d)

def add_vehicle(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    error = ""
    category1 = Category.objects.all()
    if request.method == "POST":
        pn = str(random.randint(10000000, 99999999))
        ct = request.POST['category']
        vc = request.POST['vehiclecompany']
        rn = request.POST['regno']
        on = request.POST['ownername']
        oc = request.POST['ownercontact']
        pd = request.POST['pdate']
        it = request.POST['intime']
        status = "In"
        category = Category.objects.get(categoryname=ct)
        try:
            Vehicle.objects.create(parkingnumber=pn, category=category, vehiclecompany=vc, regno=rn,
            ownername=on,
                                ownercontact=oc, pdate=pd, intime=it, outtime="", parkingcharge="", remark="",
                                status=status)

```

```
        error = "no"
    except:
        error = "yes"
    d = {'error': error, 'category1': category1}
    return render(request, 'add_vehicle.html', d)

def manage_incomingvehicle(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    vehicle = Vehicle.objects.filter(status="In")
    d = {'vehicle': vehicle}
    return render(request, 'manage_incomingvehicle.html', d)

def view_incomingdetail(request, pid):
    if not request.user.is_authenticated:
        return redirect('admin_home')
    error = ""
    vehicle = Vehicle.objects.get(id=pid)
    if request.method == 'POST':
        rm = request.POST['remark']
        ot = request.POST['outtime']
        pc = request.POST['parkingcharge']
        status = "Out"
        try:
            vehicle.remark = rm
            vehicle.outtime = ot
            vehicle.parkingcharge = pc
            vehicle.status = status
            vehicle.save()
```

```
        error = "no"
    except:
        error = "yes"
    d = {'vehicle': vehicle, 'error': error}
    return render(request, 'view_incomingdetail.html', d)

def manage_outgoingvehicle(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    vehicle = Vehicle.objects.filter(status="Out")
    d = {'vehicle': vehicle}
    return render(request, 'manage_outgoingvehicle.html', d)

def view_outgoingdetail(request, pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    vehicle = Vehicle.objects.get(id=pid)

    d = {'vehicle': vehicle}
    return render(request, 'view_outgoingdetail.html', d)

def print_detail(request, pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    vehicle = Vehicle.objects.get(id=pid)

    d = {'vehicle': vehicle}
    return render(request, 'print.html', d)
```

```
def search(request):
    q = None
    if request.method == 'POST':
        q = request.POST['searchdata']
    try:
        vehicle = Vehicle.objects.filter(Q(parkingnumber=q))
        vehiclecount = Vehicle.objects.filter(Q(parkingnumber=q)).count()

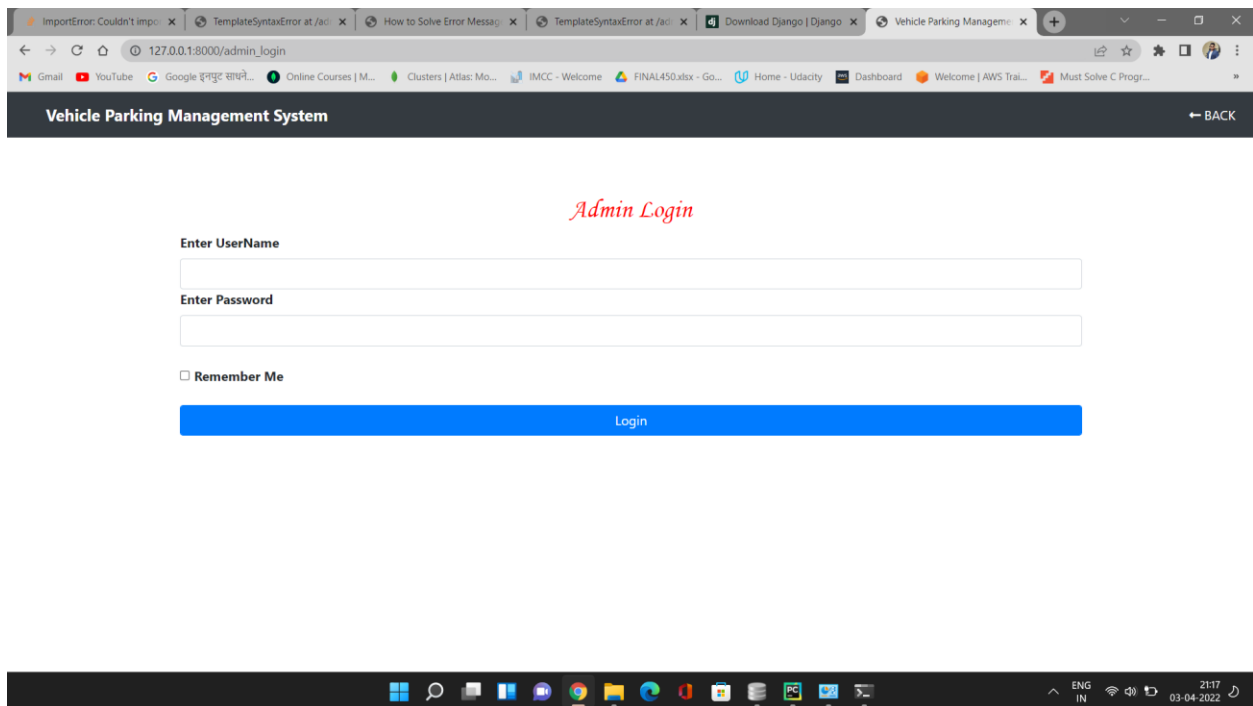
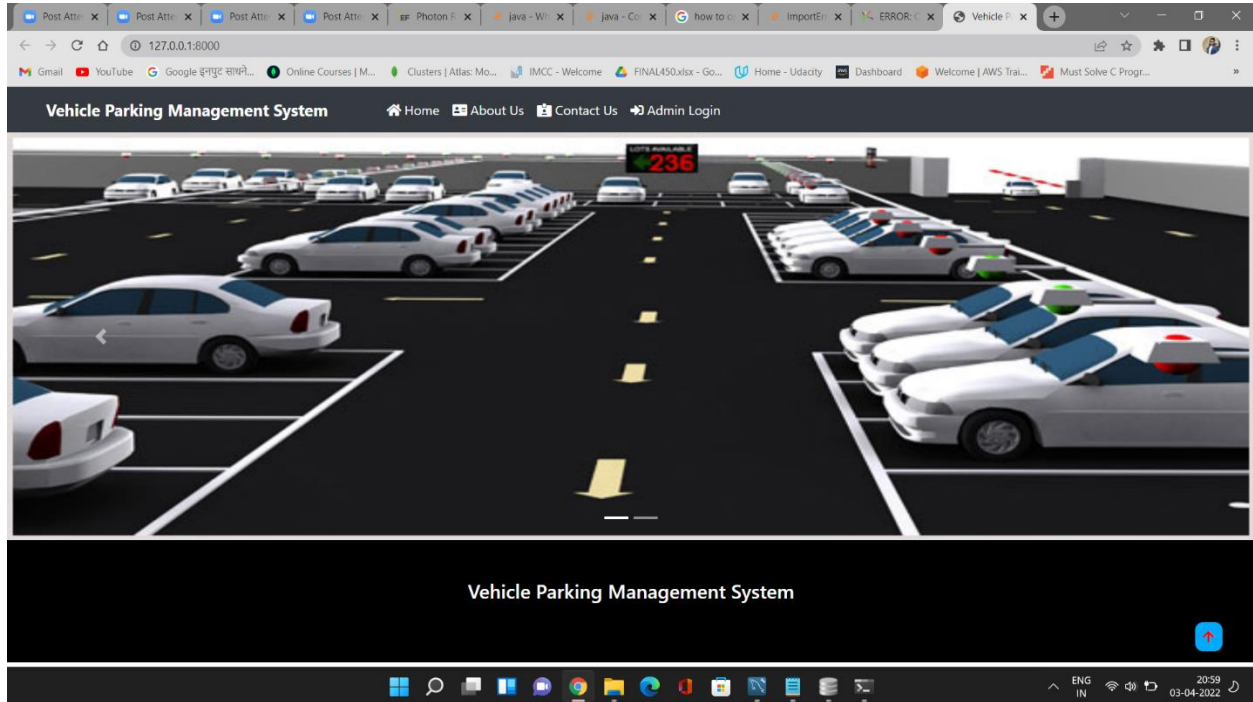
    except:
        vehicle = ""
    d = {'vehicle': vehicle, 'q': q, 'vehiclecount': vehiclecount}
    return render(request, 'search.html', d)

def betweendate_reportdetails(request):
    if not request.user.is_authenticated:
        return redirect('index')
    return render(request, 'betweendate_reportdetails.html')

def betweendate_report(request):
    if not request.user.is_authenticated:
        return redirect('index')
    if request.method == "POST":
        fd = request.POST['fromdate']
        td = request.POST['todate']
        vehicle = Vehicle.objects.filter(Q(pdate__gte=fd) & Q(pdate__lte=td))
        vehiclecount = Vehicle.objects.filter(Q(pdate__gte=fd) & Q(pdate__lte=td)).count()
        d = {'vehicle': vehicle, 'fd': fd, 'td': td, 'vehiclecount': vehiclecount}
        return render(request, 'betweendate_reportdetails.html', d)
    return render(request, 'betweendate_report.html')
```



## OUTPUT SCREEN:




Vehicle Parking Management System


[Home](#) [Category](#) [Vehicle](#) [Search](#) [Reports](#) [Change Password](#) [Logout](#)

Welcome [Admin](#)


Todays Vehicle Entries

0 


Yesterdays Vehicle Entries

0 

Last 7 days Vehicle Entries

0 

Total Vehicle Entries

5 

ENG IN

21:18

03-04-2022

Vehicle Parking Management System

[Home](#) [Category](#) [Vehicle](#) [Search](#) [Reports](#) [Change Password](#) [Logout](#)

Welcome [Admin](#)

Add New Vehicle Category

Vehicle Category Name

Vehicle Category Name

Submit

ENG IN

21:18





03-04-2022

Vehicle Parking Management System

Welcome Admin

Manage Category

Copy Excel CSV PDF Search:

S.No.	CategoryName	Action
1	Four Wheeler Vehicle	 
2	Two Wheeler Vehicle	 

Showing 1 to 2 of 2 entries

Previous 1 Next

Windows taskbar: 21:18 03-04-2022

Vehicle Parking Management System

Welcome Admin

Add Vehicle Detail

Vehicle Category  
---Select Vehicle Category---

Vehicle Company

Registration Number

Owner Name

Owner Contact Number

Parking Date

In Time

Submit

Windows taskbar: 21:18 03-04-2022

Vehicle Parking Management System

HomeCategoryVehicleSearchReportsChange PasswordLogout

Welcome Admin

Manage Incoming Vehicle

CopyExcelCSVPDF

Search:

S.No.	Parking Number	Owner Name	Vehicle Reg No.	Action
No data available in table				

Showing 0 to 0 of 0 entries

PreviousNext

21:1803-04-2022

Vehicle Parking Management System

HomeCategoryVehicleSearchReportsChange PasswordLogout

Welcome Admin

Manage Outgoing Vehicle

CopyExcelCSVPDF

Search:

S.No.	Parking Number	Owner Name	Vehicle Reg No.	Action
1	88498579	Jayesh	MH04A2300	<a href="#">View</a> <a href="#">Edit</a>
2	68449824	Pradip Shirsath	MH41AK1161	<a href="#">View</a> <a href="#">Edit</a>
3	59590060	Shubham	MH41AK1225	<a href="#">View</a> <a href="#">Edit</a>
4	70191708	Pradip	MH41PS2325	<a href="#">View</a> <a href="#">Edit</a>
5	69952265	jayessh	MH15KL2222	<a href="#">View</a> <a href="#">Edit</a>

Showing 1 to 5 of 5 entries

Previous1Next

21:1803-04-2022

Vehicle Parking Management System

Home

Category

Vehicle

Search

Reports

Change Password

Logout

Welcome Admin

Search

Search by parking number

Submit

Result against "None" keyword

S.NO	Parking Number	Owner Name	Vehicle Reg No.	Action
No record found against this search				

ENG IN

21:18

03-04-2022

Vehicle Parking Management System

Home

Category

Vehicle

Search

Reports

Change Password

Logout

Welcome Admin

Between Dates Report

Report from 2022-03-17 to 2022-04-03

Copy

Excel

CSV

PDF

Search:

S.NO	Parking Number	Owner Name	Vehicle Reg No.	Action
1	88498579	Jayesh	MH04A2300	<a href="#">View</a> <a href="#">Edit</a>
2	68449824	Pradip Shirsath	MH41AK1161	<a href="#">View</a> <a href="#">Edit</a>
3	59590060	Shubham	MH41AK1225	<a href="#">View</a> <a href="#">Edit</a>
4	70191708	Pradip	MH41PS2325	<a href="#">View</a> <a href="#">Edit</a>
5	69952265	jayessh	MH15KL2222	<a href="#">View</a> <a href="#">Edit</a>

Showing 1 to 5 of 5 entries

Previous

1

Next

ENG IN

21:19

03-04-2022

Vehicle Parking Management System

HomeCategoryVehicleSearchReportsChange PasswordLogout

Welcome Admin

Vehicle Details

Parking Number	88498579
Vehicle Category	Two Wheeler Vehicle
Vehicle Company Name	Bajaj
Registration Number	MH04A2300
Owner Name	Jayesh
Owner Contact Number	9229465037
In Time	22:57
OutTime	12:46
Remark	NA
Status	Out
Parking Fee	20

ENG IN

21:19

03-04-2022

Vehicle Parking Management System

HomeCategoryVehicleSearchReportsChange PasswordLogout

Welcome Admin

Vehicle Details

Vehicle Parking Receipt

Parking Number	88498579	Vehicle Category	Two Wheeler Vehicle
Vehicle Company Name	Bajaj	Registration Number	MH04A2300
Owner Name	Jayesh	Owner Contact Number	9229465037
In Time	22:57 March 17, 2022	Status	Outgoing Vehicle
Out time	12:46	Parking Charge	20
Remark	NA		

ENG IN

21:19

03-04-2022

### **3. Bibliography**

- <https://www.w3schools.com>
- [www.tutorial.com](http://www.tutorial.com)
- [www.youtube.com](http://www.youtube.com)