

20CYS204
CAPSTONE PROJECT
REPORT

CARTK0

CARTKO (Car Rental)

Abstract:

Our Aim is to design and create a data management System for a car rental company. this enables admin can rent a vehicle that can be used by a customer This system increases customer retention and simplify vehicle and staff Management in an efficient way.

This software car Rental System has a very user friendly interface. thus the users will feel very easy to work on it. By using this system admin can manage customer confirm and cancel booking request, customer Testimonials, customer issues. The car information can be added to the system. Or existed car information can be edited or deleted too by Administrator. There is no delay in the availability of any car information, whenever needed, car information can be Captured very quickly and easily.

The customers can also use the system to get car rent. The customer should create a new account before logging in or he / she can log into the System with his/her created account. Then he/she can book the available cars and can book this car. This system will helpful to the admin as well as to the customer also.

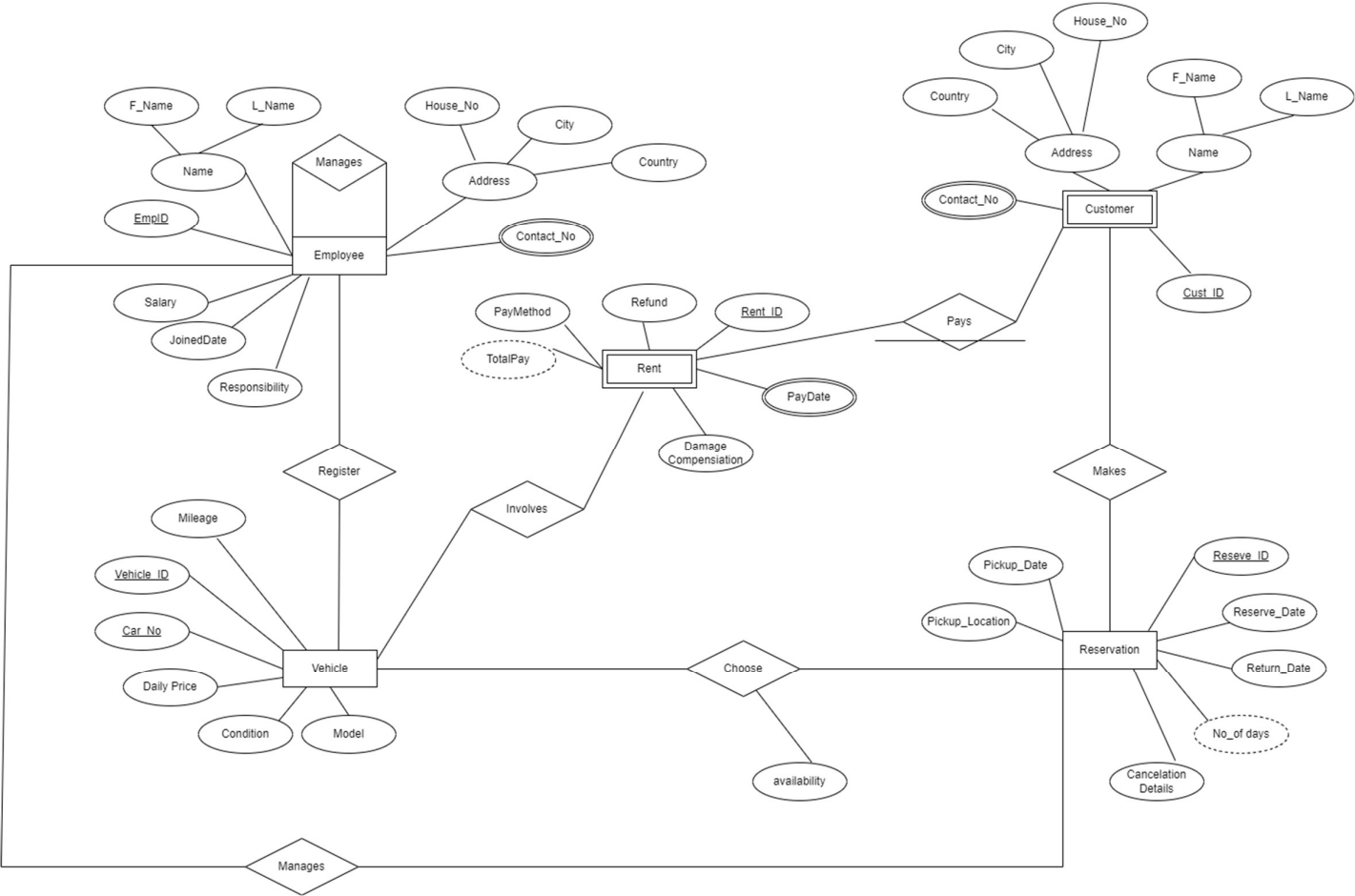
Table of Contents

S.NO	NAME	Page:No
1	Problem Statement	3
2	ER Diagram	4
3	UML Diagram	5-6
4	Snap Shots	7-10
5	Code	11
6	Conclusions	12

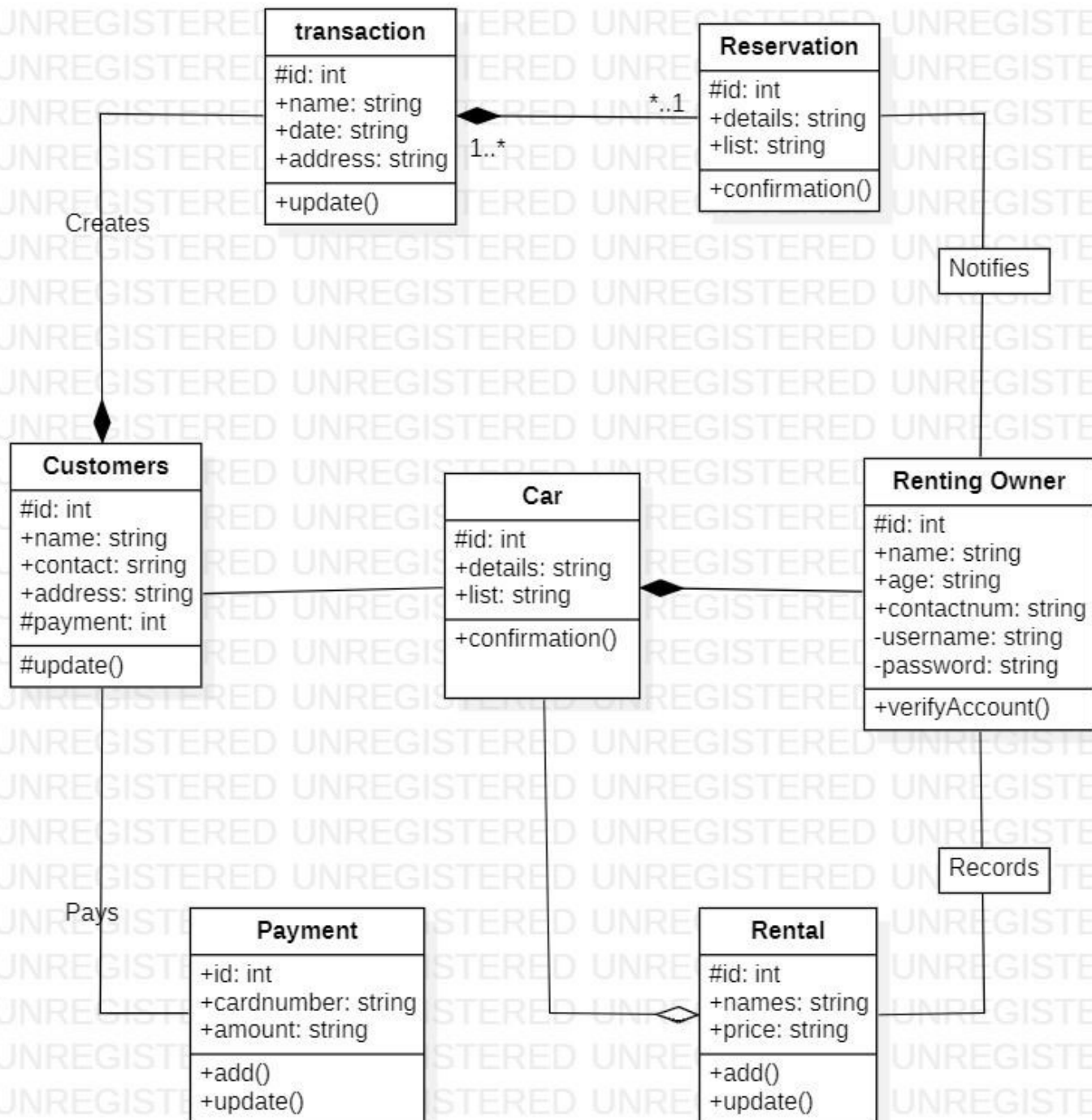
Problem Statement:

The Manual car rental system provides services only during office hours. So; customers have limited time to make any transactions or reservation of the cars. The existence of the online car rental systems nowadays has overcome the limitation of the business operation hour. However; there is still a few numbers of these online car rental systems and most of the systems offered reservation service for tourists or traveler. Besides that, there are some customers who faced a problem in choosing car to be rented which suitable with some of the important requirements.

ER Diagram:



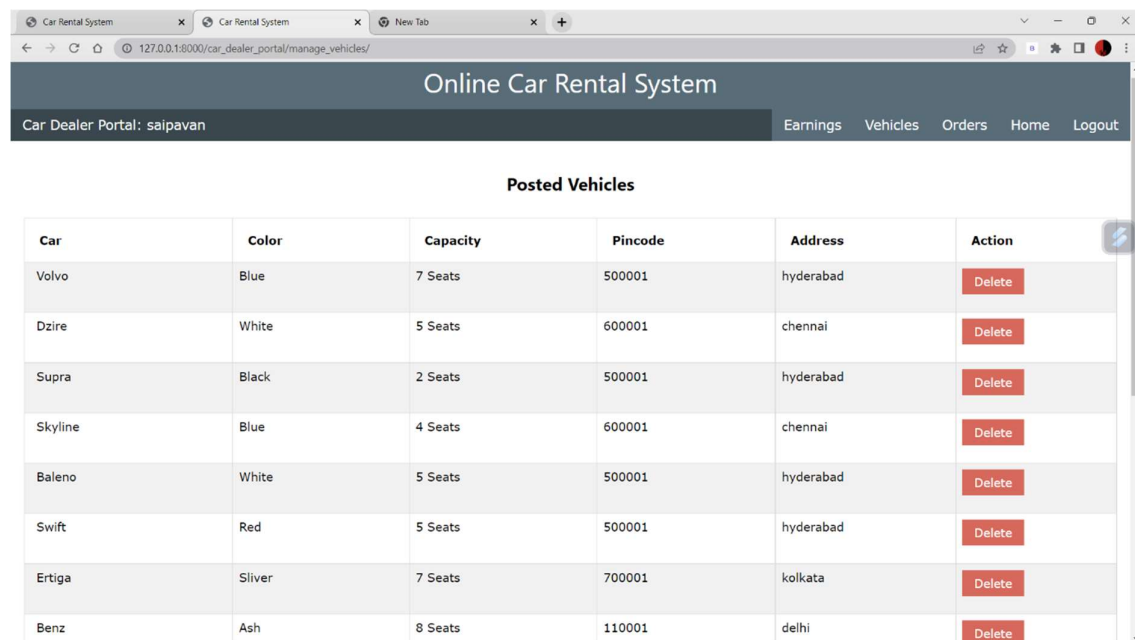
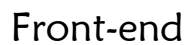
Class Diagram:



Use Case Diagram:



Bank-end



Car Rental System

Car Rental System

New Tab

127.0.0.1:8000/car_dealer_portal/auth/

Car Dealer Portal: saipavan

Earnings Vehicles Orders Logout

Post Vehicles

Car Name :

Color :


City :

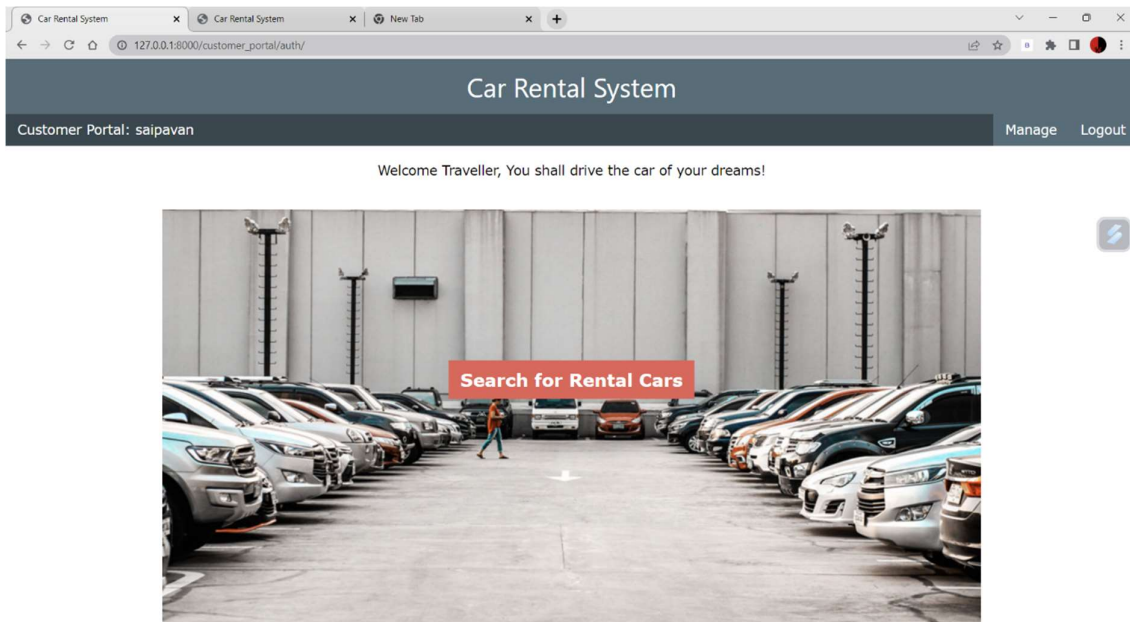
Pincode :

Description :

Capacity (Seats) :

Add Vehicle





Car Rental System

Customer Portal: saipavan

Search CarsManageLogout

Search Results

Vehicle	Color	Capacity	Description	Pincode	
Volvo	Blue	7	Good Condition	500001	Rent
Supra	Black	2	Sport Car	500001	Rent
Baleno	White	5	Family Car	500001	Rent
Swift	Red	5	Family Car	500001	Rent
Vitara brezza	Silver	5	Family Car	500001	Rent
Tata Harrier	Black	8	Family Car and Long-Drive Car	500001	Rent
KIA	Black	8	Family Car and Long-Drive Car	500001	Rent

Car Rental System

Customer Portal: saipavan

Search CarsManageLogout

Order Completed

Your Vehicle Order Has Been Processed!

Total Rent : \$1050

Vehicle : Volvo

Duration : 1 days

Car-Dealer : Sai Pavan

Car-Dealer's Contact: 8106923334

Code:

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\OneDrive\Desktop\Car_Rental-Django> customer_portal> views.py> index
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 from django.contrib.auth.models import User
4 from django.contrib.auth import authenticate
5 from django.contrib import auth
6 from customer_portal.models import *
7 from django.contrib.auth.decorators import login_required
8 from car_dealer_portal.models import *
9 from django.http import HttpResponseRedirect
10
11
12 def index(request):
13     if not request.user.is_authenticated:
14         return render(request, 'customer/login.html')
15     else:
16         return render(request, 'customer/home_page.html')
17
18 def login(request):
19     return render(request, 'customer/login.html')
20
21 def auth_view(request):
22     if request.user.is_authenticated:
23         return render(request, 'customer/home_page.html')
24     else:
25         username = request.POST['username']
26         password = request.POST['password']
27         user = authenticate(request, username=username, password=password)
28         try:
29             customer = Customer.objects.get(user = user)
30         except:
31             customer = None
32         if customer is not None:
33             auth.login(request, user)
34             return render(request, 'customer/home_page.html')
35         else:
36             return render(request, 'customer/login_failed.html')
37
38 def logout_view(request):
39     auth.logout(request)
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\OneDrive\Desktop\Car_Rental-Django> customer_portal> views.py> index
40 else:
41     return render(request, 'customer/login_failed.html')
42
43 def logout_view(request):
44     auth.logout(request)
45     return render(request, 'customer/login.html')
46
47 def register(request):
48     return render(request, 'customer/register.html')
49
50 def registration(request):
51     username = request.POST['username']
52     password = request.POST['password']
53     mobile = request.POST['mobile']
54     firstname = request.POST['firstname']
55     lastname = request.POST['lastname']
56     email = request.POST['email']
57     city = request.POST['city']
58     city = city.lower()
59     pincode = request.POST['pincode']
60     try:
61         user = User.objects.create_user(username = username, password = password, email = email)
62         user.first_name = firstname
63         user.last_name = lastname
64         user.save()
65     except:
66         return render(request, 'customer/registration_error.html')
67     try:
68         area = Area.objects.get(city = city, pincode = pincode)
69     except:
70         area = None
71     if area is not None:
72         customer = Customer(user = user, mobile = mobile, area = area)
73     else:
74         area = Area(city = city, pincode = pincode)
75         area.save()
76         area = Area.objects.get(city = city, pincode = pincode)
77         customer = Customer(user = user, mobile = mobile, area = area)
78     customer.save()
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\ Desktop> Car_Rental-Django> customer_portal> views.py> index
73
74     customer.save()
75     return render(request, 'customer/registered.html')
76
77 @login_required
78 def search(request):
79     return render(request, 'customer/search.html')
80
81 @login_required
82 def search_results(request):
83     city = request.POST['city']
84     city = city.lower()
85     vehicles_list = []
86     area = Area.objects.filter(city = city)
87     for a in area:
88         vehicles = Vehicles.objects.filter(area = a)
89         for car in vehicles:
90             if car.is_available == True:
91                 vehicle_dictionary = {'name':car.car_name, 'color':car.color, 'id':car.id, 'pincode':car.area.pincode, 'capacity':car.capacity, 'description':car.description}
92                 vehicles_list.append(vehicle_dictionary)
93     request.session['vehicles_list'] = vehicles_list
94     return render(request, 'customer/search_results.html')
95
96
97 @login_required
98 def rent_vehicle(request):
99     id = request.POST['id']
100     vehicle = Vehicles.objects.get(id = id)
101     cost_per_day = int(vehicle.capacity)*150
102     return render(request, 'customer/confirmation.html', {'vehicle':vehicle, 'cost_per_day':cost_per_day})
103
104
105 @login_required
106 def confirm(request):
107     vehicle_id = request.POST['id']
108     username = request.user
109     user = User.objects.get(username = username)
110     days = request.POST['days']
111     vehicle = Vehicles.objects.get(id = vehicle_id)
112     if vehicle.is_available:
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\ Desktop> Car_Rental-Django> customer_portal> views.py> index
113     car_dealer = vehicle.dealer
114     rent = (int(vehicle.capacity))*150*(int(days))
115     car_dealer.wallet += rent
116     car_dealer.save()
117     try:
118         order = Orders(vehicle = vehicle, car_dealer = car_dealer, user = user, rent=rent, days=days)
119         order.save()
120     except:
121         order = Orders.objects.get(vehicle = vehicle, car_dealer = car_dealer, user = user, rent=rent, days=days)
122         vehicle.is_available = False
123         vehicle.save()
124         return render(request, 'customer/confirmed.html', {'order':order})
125     else:
126         return render(request, 'customer/order_failed.html')
127
128 @login_required
129 def manage(request):
130     order_list = []
131     user = User.objects.get(username = request.user)
132     try:
133         orders = Orders.objects.filter(user = user)
134     except:
135         orders = None
136     if orders is not None:
137         for o in orders:
138             if o.is_complete == False:
139                 order_dictionary = {'id':o.id, 'rent':o.rent, 'vehicle':o.vehicle, 'days':o.days, 'car_dealer':o.car_dealer}
140                 order_list.append(order_dictionary)
141     return render(request, 'customer/manage.html', {'od':order_list})
142
143 @login_required
144 def update_order(request):
145     order_id = request.POST['id']
146     order = Orders.objects.get(id = order_id)
147     vehicle = order.vehicle
148     vehicle.is_available = True
149     vehicle.save()
150     car_dealer = order.car_dealer
151     car_dealer.wallet -= int(order.rent)
152     car_dealer.save()
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\ Desktop\ Car_Rental-Django> customer_portal> views.py> index
133 except:
134     orders = None
135     if orders is not None:
136         for o in orders:
137             if o.is_complete == False:
138                 order_dictionary = {'id':o.id,'rent':o.rent, 'vehicle':o.vehicle, 'days':o.days, 'car_dealer':o.car_dealer}
139                 order_list.append(order_dictionary)
140     return render(request, 'customer/manage.html', {'od':order_list})
141
142 @login_required
143 def update_order(request):
144     order_id = request.POST['id']
145     order = Orders.objects.get(id = order_id)
146     vehicle = order.vehicle
147     vehicle.is_available = True
148     vehicle.save()
149     car_dealer = order.car_dealer
150     car_dealer.wallet -= int(order.rent)
151     car_dealer.save()
152     order.delete()
153     cost_per_day = int(vehicle.capacity)*15
154     return render(request, 'customer/confirmation.html', {'vehicle':vehicle}, {'cost_per_day':cost_per_day})
155
156 @login_required
157 def delete_order(request):
158     order_id = request.POST['id']
159     order = Orders.objects.get(id = order_id)
160     car_dealer = order.car_dealer
161     car_dealer.wallet -= int(order.rent)
162     car_dealer.save()
163     vehicle = order.vehicle
164     vehicle.is_available = True
165     vehicle.save()
166     order.delete()
167     return HttpResponseRedirect('/customer_portal/manage/')
168
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive\ Desktop\ Car_Rental-Django> ocrs> settings.py> ...
1 |
2 |
3 import os
4 |
5 |
6 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
7 |
8 |
9 SECRET_KEY = 'bz1tc7706jpd5&ndp0&x"t&t1ccmrki+4=12yfqr@lko&+wsd%'
10 |
11 |
12 DEBUG = True
13 |
14 ALLOWED_HOSTS = []
15 |
16 |
17 |
18 |
19 |
20 |
21 INSTALLED_APPS = [
22     'django.contrib.admin',
23     'django.contrib.auth',
24     'django.contrib.contenttypes',
25     'django.contrib.sessions',
26     'django.contrib.messages',
27     'django.contrib.staticfiles',
28     'customer_portal.apps.CustomerPortalConfig',
29     'car_dealer_portal.apps.CarDealerPortalConfig',
30     'home.apps.HomeConfig',
31 ]
32 |
33 MIDDLEWARE = [
34     'django.middleware.security.SecurityMiddleware',
35     'django.contrib.sessions.middleware.SessionMiddleware',
36     'django.middleware.common.CommonMiddleware',
37     'django.middleware.csrf.CsrfViewMiddleware',
38     'django.contrib.auth.middleware.AuthenticationMiddleware',
39     'django.contrib.messages.middleware.MessageMiddleware',
40     'django.middleware.clickjacking.XFrameOptionsMiddleware',
41 ]

```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py X views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive Desktop Car_Rental-Django ocrs> settings.py ...
37 django.middleware.csrf.CsrfViewMiddleware,
38 'django.contrib.auth.middleware.AuthenticationMiddleware',
39 'django.contrib.messages.middleware.MessageMiddleware',
40 'django.middleware.clickjacking.XFrameOptionsMiddleware',
41 ]
42
43 ROOT_URLCONF = 'ocrs.urls'
44
45 TEMPLATES = [
46     {
47         'BACKEND': 'django.template.backends.django.DjangoTemplates',
48         'DIRS': [],
49         'APP_DIRS': True,
50         'OPTIONS': {
51             'context_processors': [
52                 'django.template.context_processors.debug',
53                 'django.template.context_processors.request',
54                 'django.contrib.auth.context_processors.auth',
55                 'django.contrib.messages.context_processors.messages',
56             ],
57         },
58     },
59 ]
60
61 WSGI_APPLICATION = 'ocrs.wsgi.application'
62
63
64
65
66 DATABASES = {
67     'default': {
68         'ENGINE': 'django.db.backends.mysql',
69         'NAME': 'ocrsdjango',
70         'USER': 'root',
71         'PASSWORD': '',
72         'HOST': 'localhost',
73         'PORT': '3306',
74     }
75 }
76
77
```

```
File Edit Selection View Go Run Terminal Help
views.py C:\customer_portal 9+ settings.py X views.py C:\car_dealer_portal 9+
C:\Users\pavan> OneDrive Desktop Car_Rental-Django ocrs> settings.py ...
59 ]
60
61 WSGI_APPLICATION = 'ocrs.wsgi.application'
62
63
64
65
66 DATABASES = {
67     'default': {
68         'ENGINE': 'django.db.backends.mysql',
69         'NAME': 'ocrsdjango',
70         'USER': 'root',
71         'PASSWORD': '',
72         'HOST': 'localhost',
73         'PORT': '3306',
74     }
75 }
76
77
78
79
80 AUTH_PASSWORD_VALIDATORS = [
81     {
82         'NAME': 'django.contrib.auth.password_validation.UserAttributesSimilarityValidator',
83     },
84     {
85         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
86     },
87     {
88         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
89     },
90     {
91         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
92     },
93 ]
94
95
96
97 LANGUAGE_CODE = 'en-us'
```

Conclusion

In comparison to previous experiences, when every activity related to the vehicle rental business was restricted to a physical place alone, the car rental industry has emerged with new delicacies. Even if the physical location has not been completely eliminated, the internet's power has altered the nature of functions and how these tasks are accomplished. Customers may now book vehicles online, rent automobiles online, and have the car delivered to their home if they are a registered member, or they can travel to the office to pick up the car.

Team Members

Gowtham Reddy S

Abhiram Patel U

Sai Vamsi T

Sai Pavan R