

CAR RENTAL SERVICES

Submitted to

Faculty of Information Technology

L.N. Mishra College of Business Management, Muzaffarpur

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF COMPUTER APPLICATION



Session : 2021-2024

Under the guidance of :-

**DR. VIJAY KUMAR
SINGH**

Asst. Professor
L.N.M.C.B.M, MUZAFFARPUR

Submitted by :-

PIYUSH RAJ

Roll No. : 215079

Reg. No. : 427201119407721

Session : 2021-2024

Course : BCA

Lalit Narayan Mishra College of Business Management

(Affiliated to University Grant Commission and Approved by All India Council for Technical Education)

An Autonomous (Constituent) College under B.R.A. Bihar University

Muzaffarpur



L.N. Mishra College of Business Management
 An Autonomous College under B.R. Ambedkar Bihar University
 Affiliated to UGC, Approved by AICTE & Department of Higher Education, Govt. of Bihar

DR.VIJAY KUMAR SINGH

Date:

(Asst. Professor)

GUIDE CERTIFICATE

This is to certify that the Project report entitled “**Car Rental Services**” has been prepared by **PIYUSH RAJ(215079)**, Session 2021-24 in partial fulfillment of the requirement for “BCA” Degree from **LNMCBM**, Muzaffarpur. This Certificate is under my supervision and guidance and I recommended that it may be accepted for evolution.

I wish him all success in his life.

DR.VIJAY KUMAR SINGH

(Asst. Professor)

Car Rental Service ;

BCA [2021-2024]



L.N. Mishra College of Business Management
An Autonomous College under B.R. Ambedkar Bihar University
Affiliated to UGC, Approved by AICTE & Department of Higher Education, Govt. of Bihar

**A
Project report
On**

Car Rental Services

Submitted

To

L.N MISHRA COLLEGE OF BUSINESS MANAGEMENT

Under B.R.A Bihar University, Muzaffarpur

For

The partial fulfillment of the requirement for the degree of

BACHELOR OF COMPUTER APPLICATION

Session:- 2021-2024



PIYUSH RAJ

ACKNOWLEDGEMENT

Apart from the efforts of team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge.

We are eternally grateful to our teacher **Dr. VIJAY KUMAR SINGH** for his even willingness to give us valuable advice and direction under which we executed this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

PIYUSH RAJ

Roll No.- 215079

Reg. No- **427201119407721**

Session- 2021-24

Course- BCA

CERTIFICATE

This is to certify that BCA project report entitled "Car Rental Service" is the work carried out **PIYUSH RAJ** , student of BCA, VI Semester of **L.N. MISHRA COLLEGE OF BUSINESS MANAGEMENT**, an autonomous college under B.R.Ambedkar Bihar University, under the supervision of **DR.VIJAY KUMAR SINGH** Professor,Department of BCA, **L.N. MISHRA COLLEGE OF BUSINESS MANAGEMENT**.

This report has not been submitted to any other organization/institution for the award any other degree/diploma.

Student Name – PIYUSH RAJ

Table of Contents

SNO	TOPIC	Page No
1.	PROBLEM STATEMENT	1
2.	PROCESS MODEL	2
3.	SOFTWARE REQUIREMENTS SPECIFICATION	3-6
4.	CONTEXT LEVEL DIAGRAM	7
5.	DFD LEVEL 0	8
6.	DFD LEVEL 1	9
7.	DFD LEVEL 2	10
8.	USE CASE DIAGRAM	11
9.	USE CASE DESCRIPTION	12-14
10.	DATA DICTIONARY	15
11.	ER DIAGRAM	16
12.	DATA DESIGN	17-18
13.	COMPONENT LEVEL DIAGRAM	19
14.	CODING	20-64
15.	SAMPLE SCREENSHOTS	65-68
16/	TEST CASES	69-70
17.	CONCLUSION	71
18.	LIMITATION	72-73
19.	REFERANCES	74

LIST OF FIGURES USED IN THE **PROJECT**

FIGURE NO	NAME	PAGE No
1	Main page	65
2	Login page	65
3	Registration page	66
4	All car page	66
5	Car details page	67
6	About page	68
7	Contact page	68
8	Checkout page	68
9	Payment page	69
10	Order Confirmation page	70

LIST OF TABLES USED IN THE PROJECT

TABLE NO	NAME	PAGE NO
1	Audi-Q71	
2	BMW-5	
3	Defender02	
4	Fortuner	
5	Renault-Kwid	
6	Scorpio	
7	Swift	
8	Thar	

PROBLEM STATEMENT

In the contemporary landscape of transportation, where convenience and accessibility are paramount, the traditional model of car rental services faces a myriad of challenges. Despite the proliferation of online platforms aimed at facilitating the process, a significant gap persists between consumer expectations and the actual user experience provided by car rental websites. This disparity is manifested in various inefficiencies and pain points, including but not limited to: complex and time-consuming booking procedures, limited vehicle availability especially during peak times, opacity in pricing structures leading to unexpected fees, lack of flexibility in rental terms, and inadequate customer support channels resulting in frustration and dissatisfaction among users.

Moreover, the emergence of alternative transportation options such as ridesharing and car-sharing services has intensified competition in the market, compelling traditional car rental companies to reassess their strategies and enhance their digital offerings. Failure to adapt to evolving consumer preferences and technological advancements not only risks alienating existing customers but also hampers the acquisition of new ones in an increasingly competitive landscape.

In light of these challenges, there is a critical imperative for car rental websites to undergo a paradigm shift in their approach, prioritizing user-centric design, seamless functionality, and transparent communication. By addressing these fundamental issues and leveraging emerging technologies such as artificial intelligence and data analytics, car rental services can not only meet but exceed the expectations of modern consumers, thereby securing their relevance and sustainability in an ever-evolving market.

PROCESS MODEL

Customer visit the “Car rental Services” website, browse through the selection of car and select car they want to rent

- During the booking process on the car rental website, users are presented with the option to choose cash on delivery as the payment method.
- Users proceed with their vehicle selection, rental dates, and other booking details as usual.

Cash Payment-

- Upon delivery of the vehicle, users make the payment in cash to the rental staff.

Rental Period –

- Users have full access to the rented vehicle for the duration of their reservation as agreed upon during the booking process.

Return of Vehicle-

- At the end of the rental period, users return the vehicle to the designated drop-off location.

Final Inspection and Closure –

- Rental staff inspect the returned vehicle for any damages or issues.
- Once the vehicle is deemed to be in satisfactory condition, the rental agreement is finalized, and any security deposits are refunded if applicable.

SOFTWARE REQUIREMENT SPECIFICATION

1) Introduction –

The purpose of this document is to outline the software requirements for a comprehensive car rental service system. This system will facilitate users in renting vehicles online, managing reservations, and administering the fleet of available cars.

2) Scope – The car rental service system will provide functionalities for:

- User registration and authentication.
- Browsing available vehicles. Making reservations and bookings
- Managing user profiles and preferences
- Administering vehicle inventory
- Processing payments
- Generating reports

3) Definitions & Abbreviations –

- SRS: Software Requirements Specification
- UI: User Interface
- API: Application Programming Interface

4) Product Functions –

The system will support the following functions:

- User registration and login
- Viewing available vehicles with details
- Selecting vehicles for rent based on preferences

- Booking vehicles for specified dates and times
- Administering user profiles and rental history
- Admin dashboard for managing vehicle inventory, pricing, and availability
- Integration with payment gateways for secure transactions
- Generating reports on bookings, revenues, and inventory status.

5) User Classes and Characteristics — The system will have the following user roles:

- Guest users: Can browse vehicles but cannot book without registration.
- Registered users: Can book vehicles, manage bookings, and view rental history.
- Admin users: Can manage the vehicle inventory, pricing and user accounts.

6) User Interfaces -

- User registration and login
- Vehicle browsing and selection
- Reservation and booking process
- User profile management
- Admin dashboard for managing vehicles and users

7) Software Interfaces -

- Database management system (MySQL) for data storage

8) Functional Requirements –

- User Registration

-Users can register with the system by providing required details.

- vehicle Searching

- Users can browse available vehicles with filters (e.g., by location, type, price).

- Vehicle Booking

- Users can select a vehicle, specify rental dates, and confirm the booking.

- Admin Dashboard

- Admin users can manage vehicle inventory, add/remove vehicles, set pricing, and view booking reports.

- Payment Processing

- Users can pay securely for bookings using integrated payment gateways.

- Reporting

- Generate reports for admin users on bookings, revenues, and inventory status.

9) Non-Functional Requirements –

- Performance

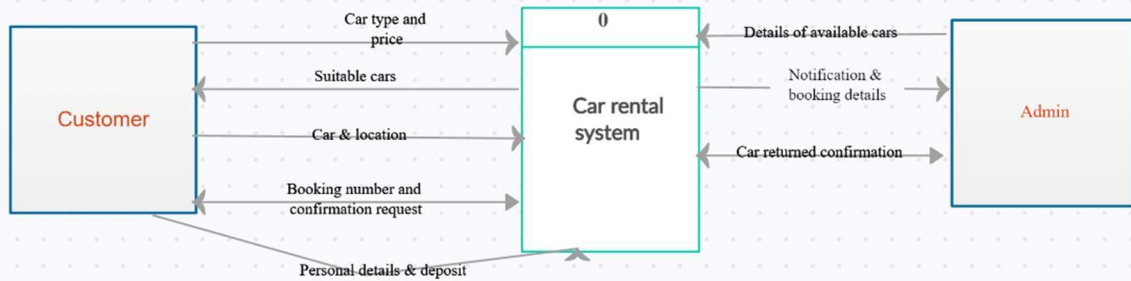
- The system should handle concurrent user requests efficiently.

- Security
 - Implement proper authentication mechanisms and data encryption for user privacy.
- Reliability
 - Ensure system stability and minimize down me.
- Usability
 - Provide a user-friendly interface for seamless navigation and booking.
 - Provide user manuals and system documentation for administrators and end-users.

Overview

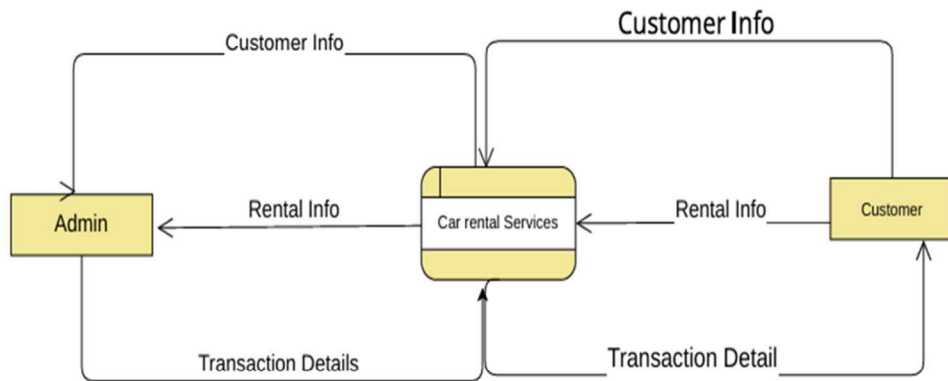
This Software Requirements Specification (SRS) outlines the key features and functionalities of a car rental service system. Detailed design and implementation specifications will follow based on these requirements.

CONTEXT LEVEL DIAGRAM

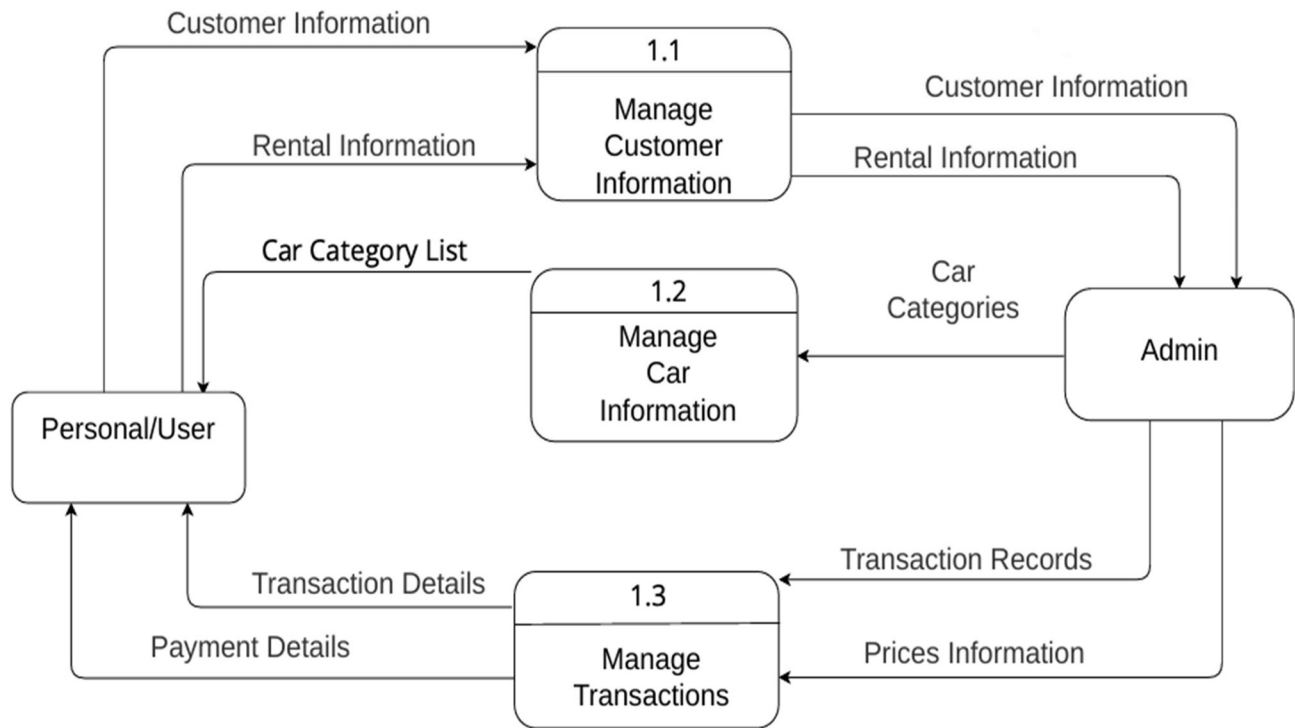


ZERO LEVEL DFD

0 LEVEL DFD

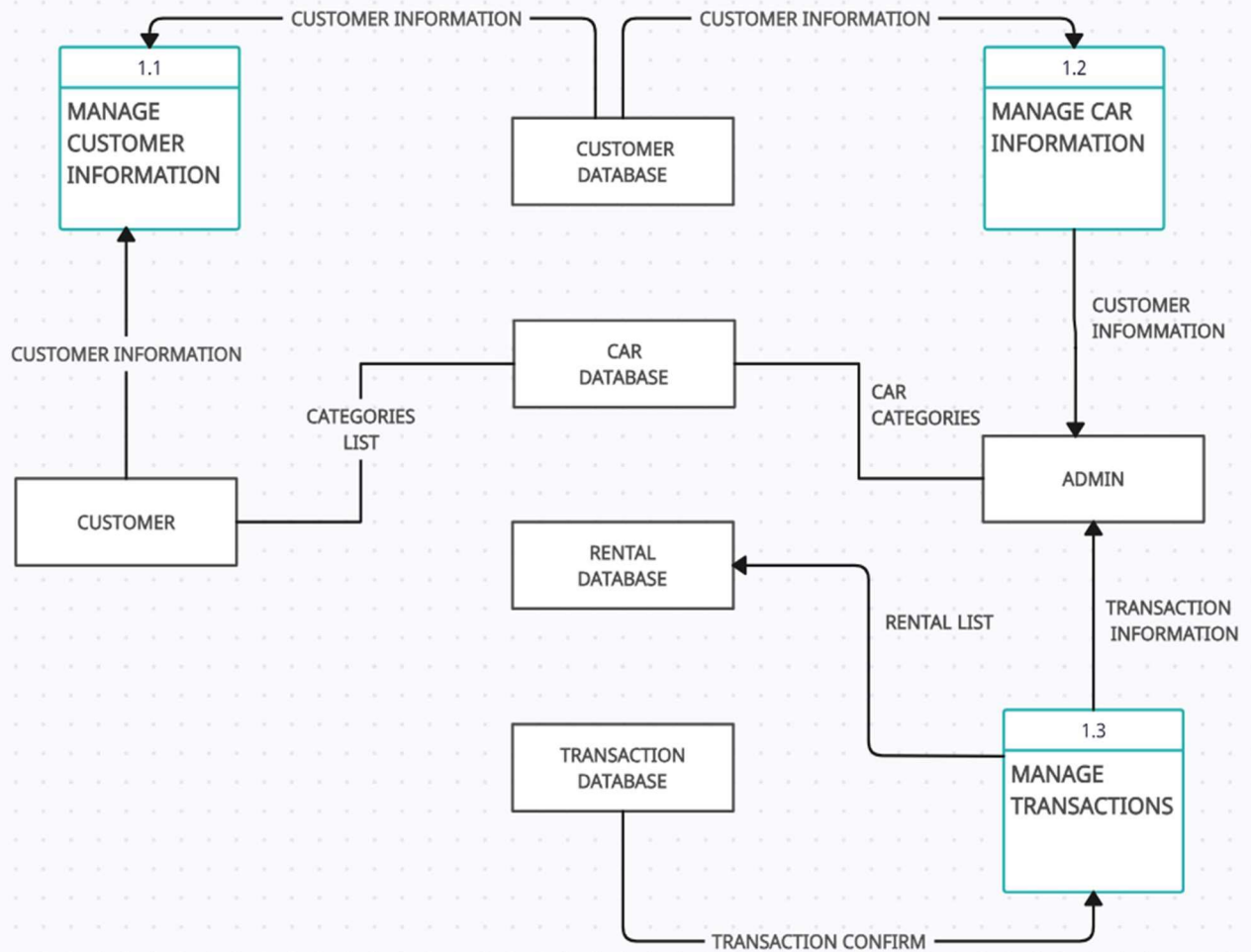


FIRST LEVEL DFD

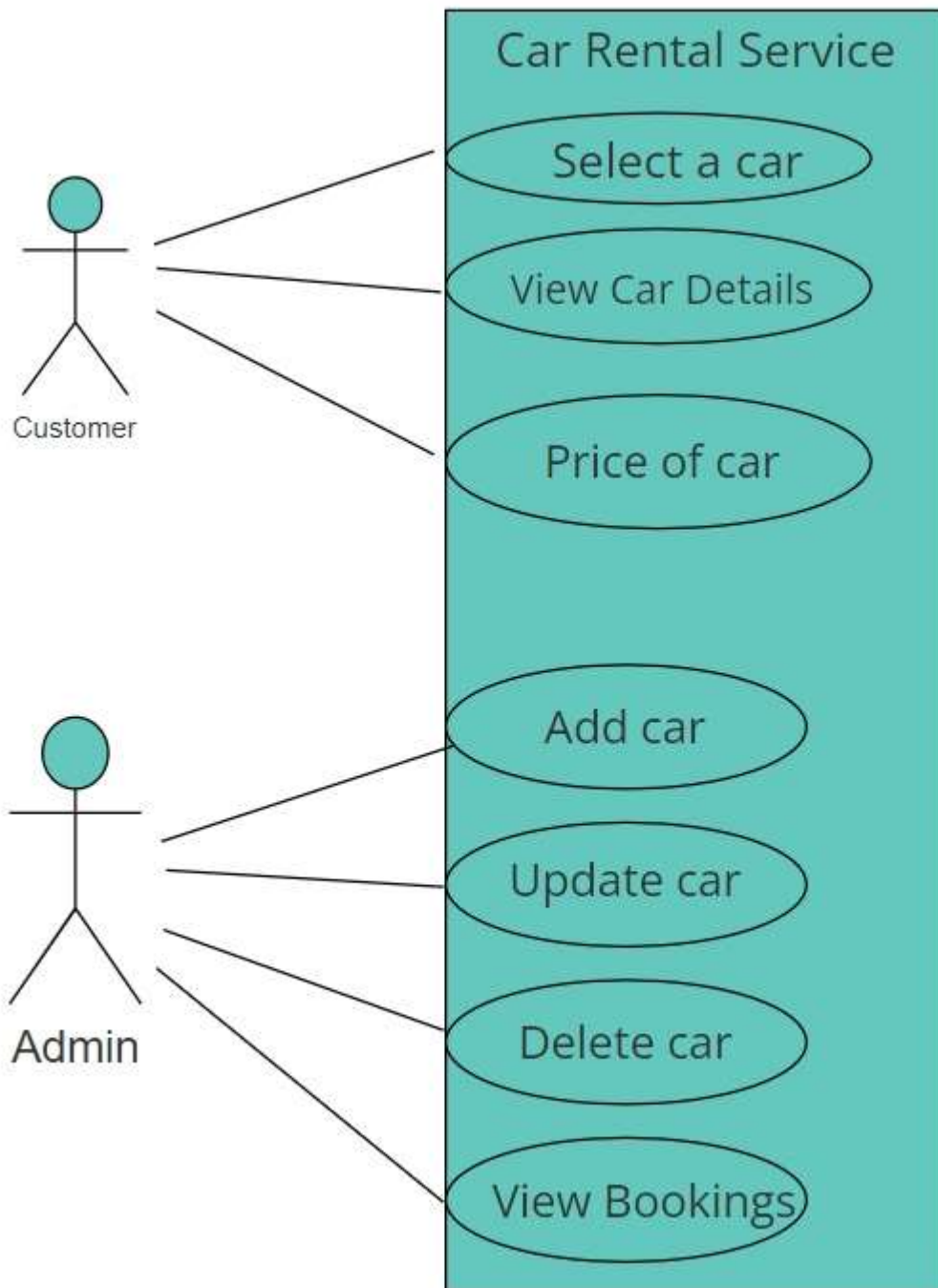


First Level DFD

SECOND LEVEL DFD



USE CASE DIAGRAM



USE CASE DESCRIPTION

Car Rental System

A Car Rental System is a software application designed to help customers rent cars from a rental company. The system includes features such as car inventory management and payment processing. The use case diagram outlines the tasks that can be performed within the system.

Customers can search for available cars based on their preferences, view car details, and reserve a car for a specified rental period. This enables customers to easily find and rent a car that meets their needs and preferences, while also providing them with flexibility in case their plans change.

The system can also help our website to view and manage customer booking, including booking details such as rental period, pickup location, and rental price. This allows us to ensure that they have enough cars available for customers to rent and manage their car inventory effectively.

Overall, the Car Rental System use case diagram outlines the key functionalities of the system and demonstrates how it can be used to manage various aspects of car rental operations. The system can streamline the rental process, improve communication between rental companies and customers, and ensure that car rental transactions are executed successfully.

Pros of creating this use case diagram

Creating a use case diagram for a Car Rental System can provide several benefits. First, it helps to clarify the system's functionality and how it can be used to support our car rental website.

Second, the use case diagram can serve as a valuable communication tool between us, including developers, and customers. It can help ensure that everyone involved in the system's development and implementation is on the same page and has a shared understanding of the system's purpose and functionality. This can help prevent misunderstandings and ensure that the system meets the organization's needs and expectations.

Overall, a use case diagram can be a useful tool for planning and designing a Car Rental System, helping to ensure that the system is aligned with the goals and needs, and that it is capable of supporting various car rental activities. It can also help to facilitate communication and collaboration between stakeholders, improving the chances of a successful implementation and adoption of the system.

Benefits of creating this use case diagram

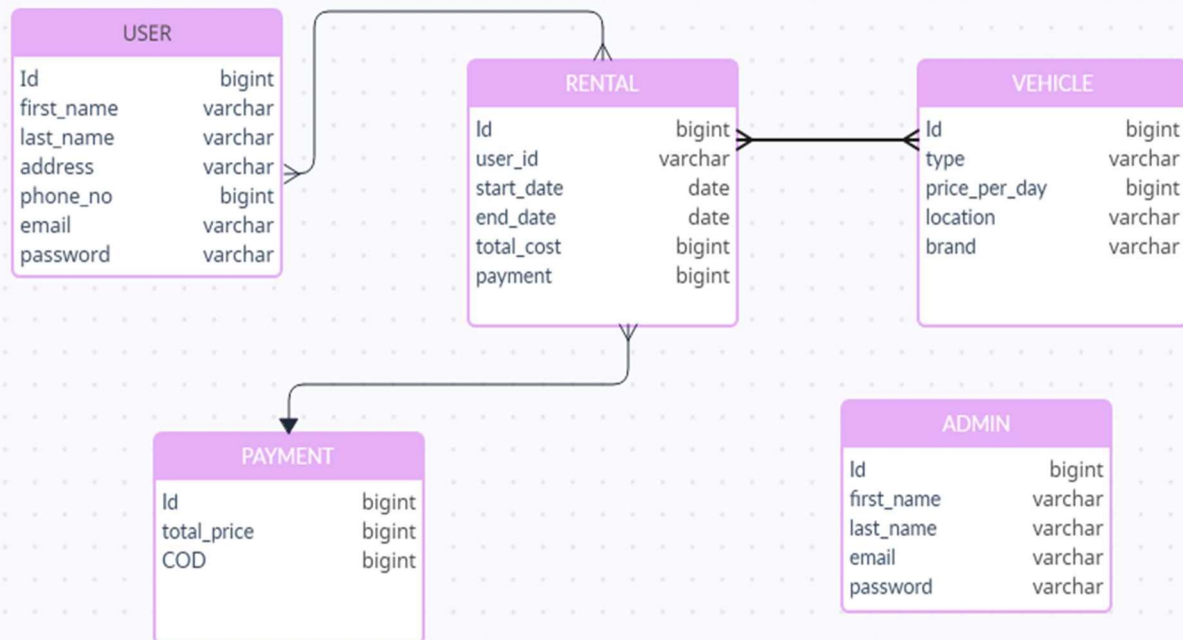
Creating a use case diagram for a Car Rental Website can provide several benefits. First, it can help to identify any gaps or areas of improvement in the system's functionality and ensure

that all necessary use cases have been accounted for. This can help to avoid potential issues or errors that could arise during the system's implementation and use.

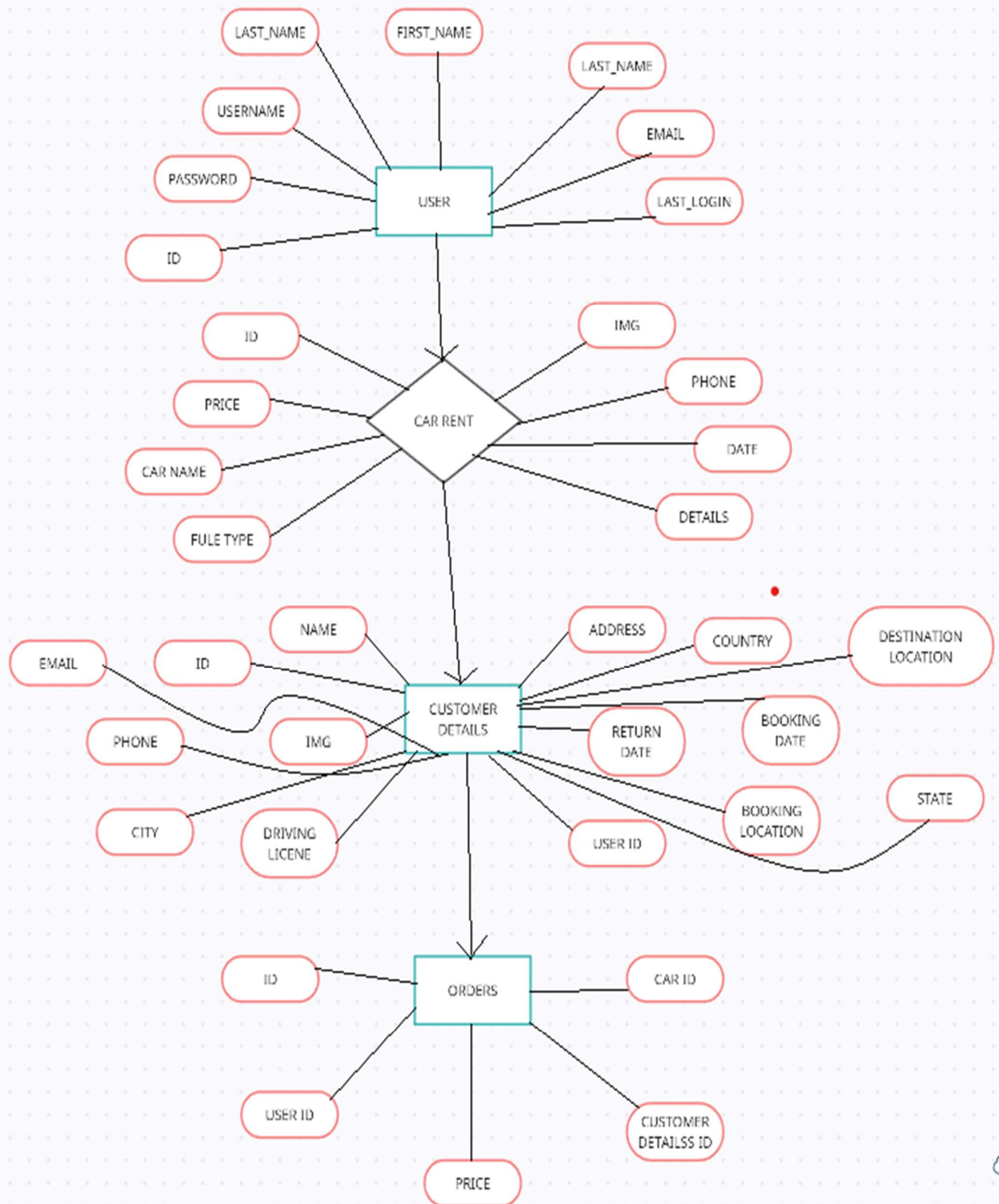
Second, the use case diagram can serve as a useful reference tool for all stakeholders involved in the system's development and implementation. It can help to ensure that everyone has a shared understanding of the system's purpose and functionality, and can help to facilitate communication and collaboration between us. This can help to ensure that the system meets the organization's needs and expectations, and that it is implemented successfully.

Overall, a use case diagram can be a valuable tool in the development and implementation of a Car Rental website, helping to ensure that the system is aligned with the organization's goals and needs, and that it is capable of supporting various car rental-related activities. It can also help to identify potential issues or gaps in the system's functionality, and can improve communication and collaboration between stakeholders, ultimately leading to a more successful implementation and adoption of the system.

DATA DICTIONARY



ER-DIAGRAM



DATA DESIGN

Data Design for Car Rental Website Services:

1. User Data :

- User ID
- Username
- Email
- Password (hashed)
- Name
- Contact Information
- Address
- Booking History

2. Vehicle Data :

- Model
- Year
- Type
- Fuel Type
- Availability Status
- Rental Rates
- Location

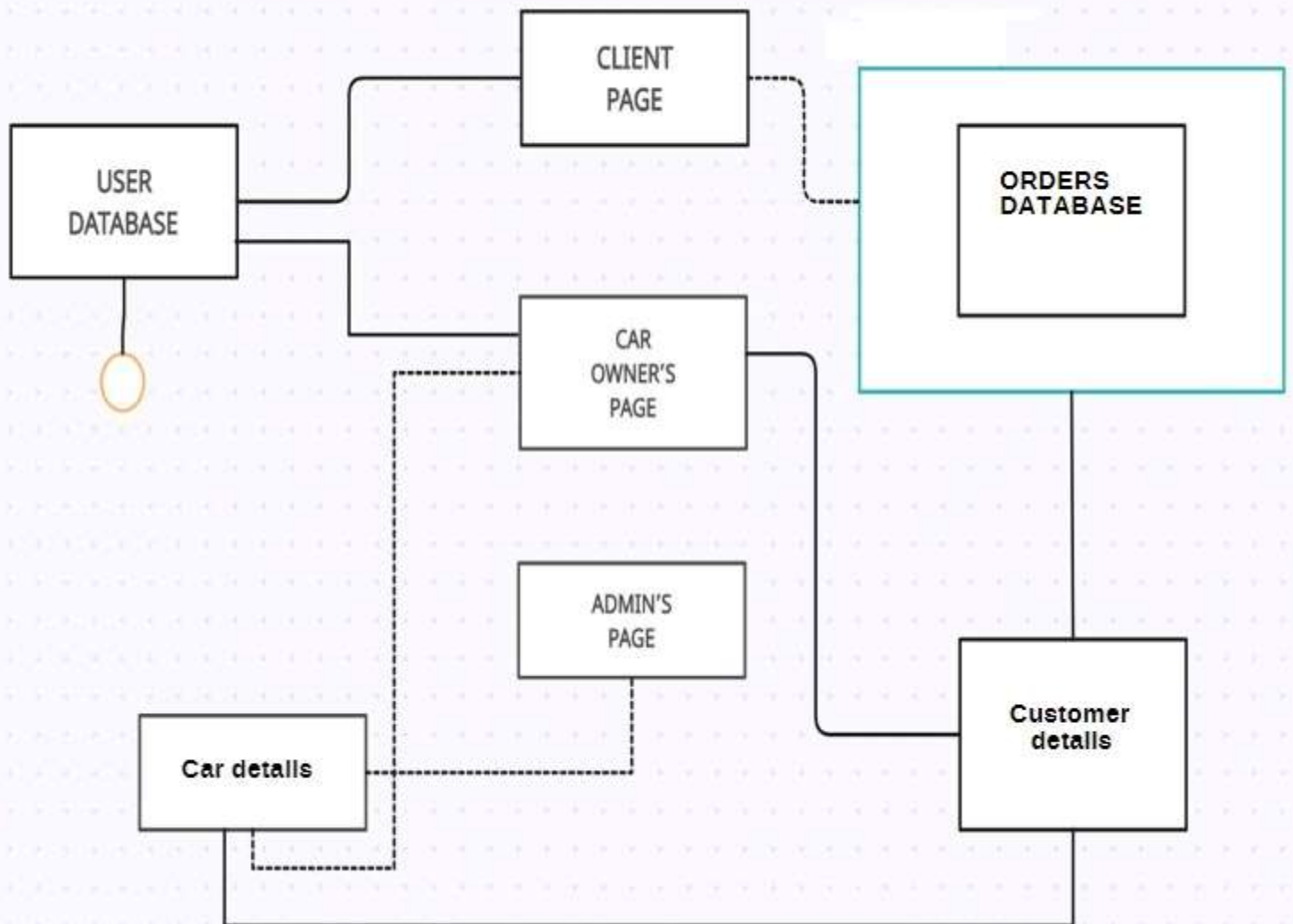
3. Booking Data :

- Vehicle ID
- Rental Start Date
- Rental End Date
- Total Cost
- Payment Method
- Payment Status
- Pickup Location

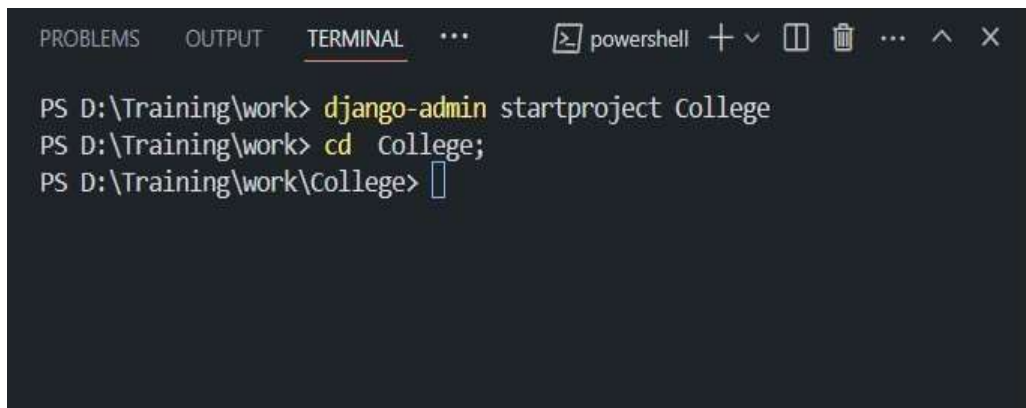
4. Payment Data :

- COD
- Payment Amount
- Payment Date
- Payment Status

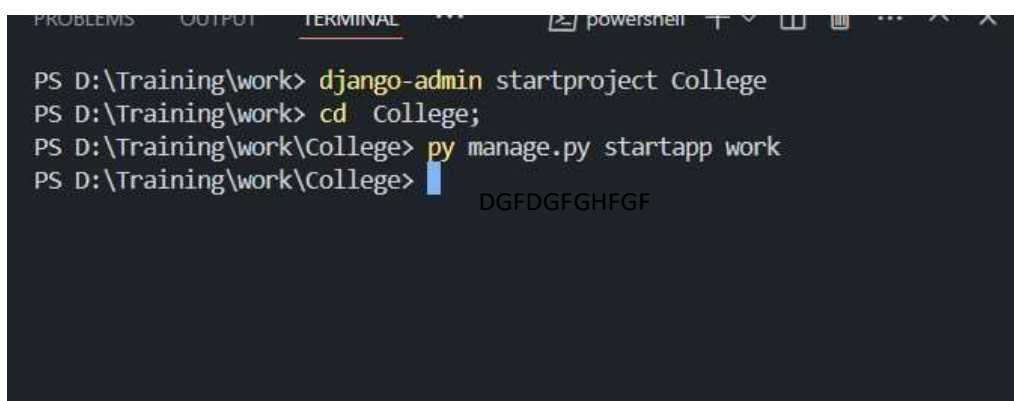
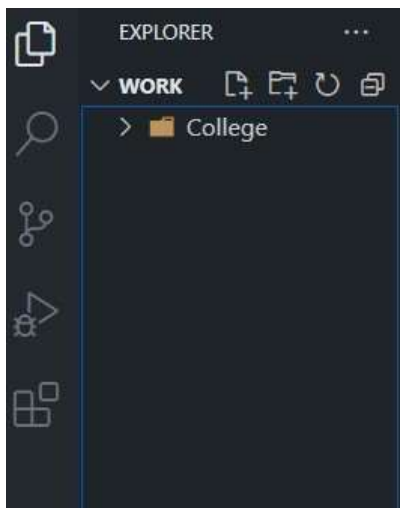
COMPONENT LEVEL DIAGRAM



CODING

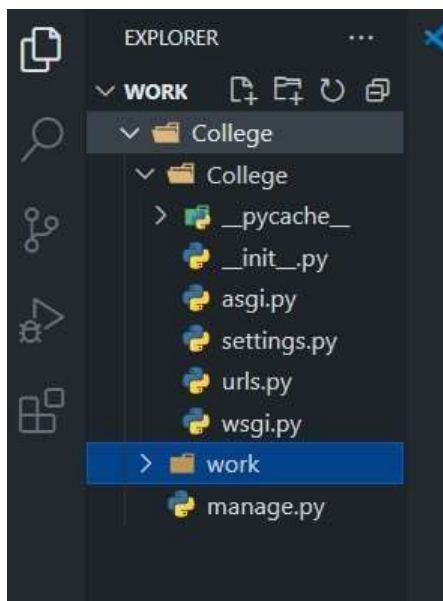
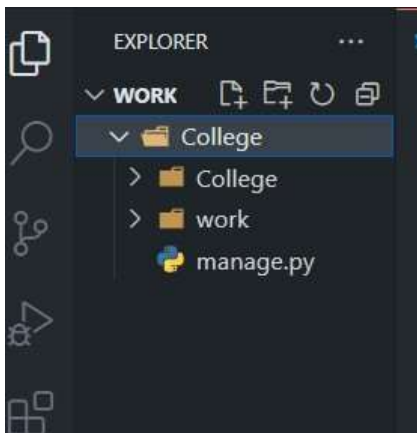


```
PROBLEMS OUTPUT TERMINAL ... powershell + - [ ] [ ] ... ^ X  
PS D:\Training\work> django-admin startproject College  
PS D:\Training\work> cd College;  
PS D:\Training\work\College> |
```



```
PROBLEMS OUTPUT TERMINAL ... powershell + - [ ] [ ] ... ^ X  
PS D:\Training\work> django-admin startproject College  
PS D:\Training\work> cd College;  
PS D:\Training\work\College> py manage.py startapp work  
PS D:\Training\work\College> |  
DGFDGFGHFGF
```

CODING



```
settings.py X
College > settings.py > ...

32
33 INSTALLED_APPS = [
34     'work',
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41 ]
42
```

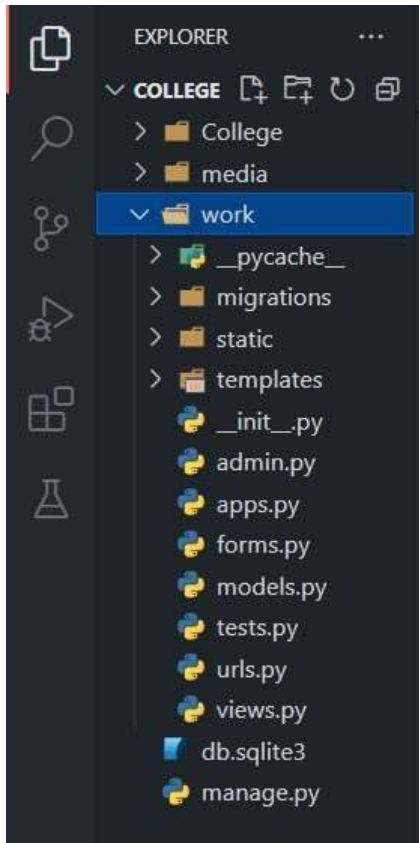
CODING

```
settings.py X
College > settings.py > ...
83
84
85 DATABASES = {
86     'default': {
87         'ENGINE': 'django.db.backends.mysql',
88         'NAME': 'college',|
89         'USER': 'root',
90         'PASSWORD': 'Piyush01',
91         'HOST': 'localhost', # Or your MySQL host
92         'PORT': '3306',      # Or your MySQL port
93         'OPTIONS': {
94             'charset': 'utf8mb4', # Use utf8mb4 for emoji support
95         },
96     }
97 }
98
```

```
settings.py X
College > settings.py > ...
135
136 import os
137 MEDIA_ROOT = os.path.join(BASE_DIR, "media")
138 MEDIA_URL = "/media/"
139
140 # Default primary key field type
141 # https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field
142
143 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
144
```

```
urls.py X
College > urls.py > ...
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19 from django.conf import settings
20 from django.conf.urls.static import static
21
22 urlpatterns = [
23     path('', include('work.urls')),
24     path('admin/', a (function) MEDIA_URL: Any
25 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
26
27
28 admin.site.site_header="Work Administration"
29 admin.site.site_title="Piyush"
30 admin.site.index_title="Welcome"
```

CODING



Database

Car Rent

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
Carname	varchar(150)	NO		NULL	
Fueltype	varchar(50)	NO		NULL	
Price	varchar(500)	NO		NULL	
img	varchar(100)	NO		NULL	
phone	varchar(15)	NO		NULL	
Date	date	YES		NULL	
Details	varchar(1000)	NO		NULL	

8 rows in set (0.01 sec)

Database

Customer details

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
Name	varchar(100)	NO		NULL	
Email	varchar(50)	NO		NULL	
phone	int	YES		NULL	
DrivingLicene	varchar(100)	NO		NULL	
Address	varchar(500)	NO		NULL	
Bookingdate	datetime(6)	YES		NULL	
Returndate	datetime(6)	YES		NULL	
Bookinglocation	varchar(255)	NO		NULL	
Destinationlocation	varchar(255)	NO		NULL	
City	varchar(35)	NO		NULL	
State	varchar(50)	NO		NULL	
Pincode	int	YES		NULL	
Country	varchar(50)	NO		NULL	
user_id	int	NO	MUL	NULL	
img	varchar(100)	NO		NULL	

16 rows in set (0.00 sec)

Users

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
password	varchar(128)	NO		NULL	
last_login	datetime(6)	YES		NULL	
is_superuser	tinyint(1)	NO		NULL	
username	varchar(150)	NO	UNI	NULL	
first_name	varchar(150)	NO		NULL	
last_name	varchar(150)	NO		NULL	
email	varchar(254)	NO		NULL	
is_staff	tinyint(1)	NO		NULL	
is_active	tinyint(1)	NO		NULL	
date_joined	datetime(6)	NO		NULL	

11 rows in set (0.01 sec)

Orders

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
carid	varchar(100)	NO		NULL	
price	varchar(100)	NO		NULL	
customerdetails_id	bigint	NO	MUL	NULL	
user_id	int	NO	MUL	NULL	

5 rows in set (0.18 sec)

CODING

Urls.py

```
from django.contrib import admin
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('home/', views.home, name='home'),
    path('login/', views.login_page, name='login_page'),
    path('logout/', views.logout_page,
name='logout_page'),
    path('cardetails/', views.cardetails, name='cardetails
'),
    path('checkout1/', views.checkout1, name='checkout1'),
    path('ridepayment/', views.ridepayment, name='ridepaym
ent'),
    path('about_us/', views.about_us, name='about_us'),
    path('order1/', views.order1, name='order1'),
    path('register/', views.register_page, name='register_
page'),
]
```

CODING

models.py

```
from django.db import models
from django.contrib.auth.models import User

# Create your models here.
class CarRent(models.Model):
    Carname = models.CharField(max_length=150)
    Fueltype = models.CharField(max_length=50)
    Price = models.CharField(max_length=500)
    img = models.ImageField(default='')
    phone = models.CharField(max_length=15)
    Date = models.DateField(null=True)
    Details = models.CharField(max_length=1000)

    def __str__(self):
        return f"{self.Carname} {self.Fueltype} {self.Price} {self.img} {self.phone} {self.Date} {self.Details}"

class Customerdetails(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    Name = models.CharField(max_length=100)
    Email = models.CharField(max_length=50)
    phone = models.IntegerField(null=True)
    DrivingLicene = models.CharField(max_length=100)
    #img = models.ImageField(default='')
    Address = models.CharField(max_length=500)
    Bookingdate = models.DateTimeField(null=True)
    Returndate = models.DateTimeField(null=True)
    Bookinglocation = models.CharField(max_length=255)
    Destinationlocation = models.CharField(max_length=255)
    City=models.CharField(max_length=35)
```

CODING

```
State=models.CharField(max_length=50)
Pincode=models.IntegerField(null=True)
Country=models.CharField(max_length=50)

def __str__(self):
    return f"{self.Name}"

class Orders(models.Model):
    user =
models.ForeignKey(User,on_delete=models.CASCADE)
    customerdetails =
models.ForeignKey(Customerdetails,on_delete=models.CASCADE)

    carid = models.CharField(max_length=100)
    price = models.CharField(max_length=100)

    def __str__(self):
        return f"{self.user} {self.customerdetails}
{self.carid} {self.price}"
```

admin.py

```
from django.contrib import admin
from .models import CarRent
from .models import Customerdetails,Orders

# Register your models here.

class CarRentAdmin(admin.ModelAdmin):
    list_display=("Carname", "Fueltype", "Price", "img",
"phone", "Date", "Details",)
```

CODING

```
admin.site.register(CarRent, CarRentAdmin)
admin.site.register(Customerdetails)

class OrdersAdmin(admin.ModelAdmin):
    list_display=("user", "customerdetails", "carid",
"price",)
admin.site.register(Orders, OrdersAdmin)
```

views.py

```
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth import authenticate, login ,
logout
from django.contrib.auth.decorators import
login_required
from django.contrib.auth.models import User
from datetime import datetime
from dateutil.relativedelta import relativedelta
from .models import *
from . models import CarRent
from . models import Customerdetails

# Create your views here.
def home(request):
    rent = CarRent.objects.all()
    return render(request, 'index.html', {'rent':rent})

def login_page(request):
```

CODING

```
# Check if the HTTP request method is POST (form
submission)
if request.method == "POST":
    username = request.POST.get('username')
    password = request.POST.get('password')

    # Check if a user with the provided username
exists
    if not
User.objects.filter(username=username).exists():
        # Display an error message if the username
does not exist
        messages.error(request, 'Invalid Username')
        return redirect('login_page')

    # Authenticate the user with the provided
username and password
    user = authenticate(username=username,
password=password)

    if user is None:
        # Display an error message if authentication
fails (invalid password)
        messages.error(request, "Invalid Password")
        return redirect('login_page')
    else:
        # Log in the user and redirect to the home
page upon successful login
        login(request, user)
        return redirect('home')

# Render the login page template (GET request)
return render(request, 'login.html')

def register_page(request):
```

CODING

```
# Check if the HTTP request method is POST (form
submission)
if request.method == 'POST':
    first_name = request.POST.get('first_name')
    last_name = request.POST.get('last_name')
    username = request.POST.get('username')
    password = request.POST.get('password')

    # Check if a user with the provided username
already exists
    user = User.objects.filter(username=username)

    if user.exists():
        # Display an information message if the
username is taken
        messages.info(request, "Username already
taken!")
        return redirect('register_page')

    # Create a new User object with the provided
information
    user = User.objects.create_user(
        first_name=first_name,
        last_name=last_name,
        username=username
    )

    # Set the user's password and save the user
object
    user.set_password(password)
    user.save()

    # Display an information message indicating
successful account creation
```

CODING

```
        messages.info(request, "Account created  
Successfully!")  
        return redirect('register_page')  
  
    # Render the registration page template (GET  
request)  
    return render(request, 'Register.html')  
  
def logout_page(request):  
    logout(request)  
    return redirect('login_page')  
  
def cardetails(request):  
    carid = request.GET.get('carid','')  
    cardata = CarRent.objects.filter(id=carid)  
    return  
render(request, 'cardetails.html',{'car':cardata})  
  
def checkout1(request):  
    carprice = request.GET.get('carprice','')  
    carid = request.GET.get('carbook','')  
    #print(carprice)  
    if request.method == 'POST':  
        name = request.POST.get('name','')  
        cprice = request.POST.get('price','')  
        carid = request.POST.get('carid','')  
        email = request.POST.get('email','')  
        phone = request.POST.get('phone','')  
        licence = request.POST.get('licence','')  
        images = request.POST.get('images','')  
        address = request.POST.get('Address','')  
        bookingdate = request.POST.get('bookingdate','')  
        bookinglocation =  
request.POST.get('bookinglocation','')  
        returndate = request.POST.get('returndate','')
```

CODING

```
destinationlocation =
request.POST.get('destinationlocation','')
city = request.POST.get('city','')
state = request.POST.get('State','')
pincode = request.POST.get('pincode','')
country = request.POST.get('country','')
Customerdetails.objects.filter()
if request.user.is_authenticated:
    Customerdetails.objects.create(
        user = request.user,
        Name = name ,
        Email = email ,
        phone = phone ,
        DrivingLicene = licence ,
        img = images ,
        Address = address ,
        Bookingdate = bookingdate ,
        Returndate = returndate ,
        Bookinglocation = bookinglocation ,
        Destinationlocation =
destinationlocation ,
        City = city ,
        State = state ,
        Pincode = pincode ,
        Country = country

    )
    bookingdate = datetime.strptime(bookingdate,
'%Y-%m-%d')
    returndate = datetime.strptime(returndate, '%Y-
%m-%d')
    difference = relativedelta(returndate,
bookingdate)
    number_of_days = difference.days
    print(number_of_days)
```


CODING

```
        print("hi")
        print(cprice)
        #print(number_of_days*carprice)
        amount=int(number_of_days)* int(cprice)
        print(amount)
        return
render(request, 'payment.html', {'days':number_of_days, 'price':cprice,
'amount':amount, 'carid':carid, 'licence':licence})

        return
render(request, 'checkout.html', {"price":carprice, 'carid':carid})

def ridepayment(request):

    return render(request, 'payment.html')

def about_us(request):
    return render(request, 'AboutUs.html')

def order1(request):
    carid = request.GET.get('carid', '')
    days = request.GET.get('days', '')
    price = request.GET.get('price', '')
    amount = request.GET.get('amount', '')
    licence = request.GET.get('licence', '')
    cardata = CarRent.objects.filter(id = carid)
    customerdata =
Customerdetails.objects.get(DrivingLicene=licence)
    # id = 0
    # for i in customerdata:
    #     id +=i.id

    if request.user.is_authenticated:
```

CODING

```
Orders.objects.create(
    user = request.user ,
    customerdetails = customerdata,
    carid = carid,
    price = amount

)

return
render(request, 'order.html', {'cardata': cardata, 'days': days, 'price': price, 'amount': amount})
```

main.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Car rental Service</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/
css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6h
W+ALEwIH" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
```

CODING

```
awesome/6.5.2/css/all.min.css" integrity="sha512-
SnH5WK+bZxgPHs44uWIX+LLJAJ9/2PkPKZ5QiAj6Ta86w+fsb2TkcmfR
yVX3pBnMFCv7oQPJkl9QevSCWr3W6A=="
crossorigin="anonymous" referrerpolicy="no-referrer" />
    {% block style %}
    {% endblock %}

</head>
<body>
    <nav class="navbar navbar-expand-lg bg-body-
tertiary" id="header">
        <div class="container-fluid" style="background-
color: rgba(76, 192, 153, 0.5);">
            {% load static %}
            <a class="navbar-brand" href="#"></a>
            <button class="navbar-toggler" type="button"
data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link active" aria-
current="page" href="/"><i class="fa-solid fa-
house"></i> Home</a>
                    </li>
                    <li class="nav-item">
```

CODING

```
        <a class="nav-link active" href="{% url
'about_us' %}"><i class="fa-solid fa-circle-info"></i>
About Us</a>
    </li>
    <li class="nav-item">
        <a class="nav-link active" href="#"><i
class="fa-solid fa-comments"></i> Feedback</a>
    </li>
    <li class="nav-item">
        <a class="nav-link active" href="{% url
'about_us' %}"><i class="fa-regular fa-address-
book"></i> Contact Us</a>
    </li>

</ul>
<div class="header-button">
    {% if user.is_authenticated %}
        <a href="#" class="sign-in btn btn-primary
"><i class="fa-solid fa-user"></i>{{user.username}}</a>
        <a href="{% url 'logout_page' %}"
class="sign-in btn btn-primary "><i class="fa-solid fa-
user"></i>Logout</a>
    {% else %}
        <a href="{% url 'login_page' %}"
class="sign-in btn btn-primary "><i class="fa-solid fa-
user"></i> SignIn</a>
        <a href="{% url 'register_page' %}"
class="sign-up btn btn-primary"><i class="fa-solid fa-
user"></i> SignUp</a>
    {% endif %}
</div>
</div>
</div>
</nav>
{% block body %}
```

CODING

```
{% endblock %}
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/j
s/bootstrap.bundle.min.js" integrity="sha384-
YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN
7N6jIeHz" crossorigin="anonymous"></script>
{% block script %}

{% endblock %}
</body>
</html>
```

index.html

```
{% extends 'main.html'%}

{% block style %}
{% load static %}

<link rel="stylesheet" href="{% static 'mystyle.css'%}">

{% endblock %}

{% block body %}

    <section class="view" id="view">
        <div class="text">
            <h2><span>Easy</span> & Fast <br>Way to Rent
<br>a Car.</h2>
```

CODING

```
        <p><i>big</i></p><br>
    memories</i></p><br>
</div>
```

```
</section>
```

```
<section class="travel" id="travel">
    <div class="top-view">
        <span>How It's Work.</span>
        <h2>Rent with 4 Easy Steps.</h2>
    </div>
```

```
        <div class="travel-container">
            <div class="box">
                <i class="fa-solid fa-location-dot"></i>
                <h2>Choose Location</h2>
                <p>Experience the road like never
before.</p>
            </div>
```

```
            <div class="box">
                <i class="fa-solid fa-calendar-
days"></i>
                <h2>Book a Car</h2>
                <p>Rent a car that suits your every
need..</p>
            </div>
```

```
            <div class="box">
                <i class="fa-solid fa-calendar-
check"></i>
                <h2>Pick-Up Date</h2>
                <p>Unforgettable journeys start
here.</p>
            </div>
```

CODING

```
<div class="box">
  <i class="fa-solid fa-calendar-
check"></i>
  <h2>Return Date</h2>
  <p>Thanks for travel with us.</p>
</div>
</div>
</section>

<section class="service" id="service">
  <div class="top-view">
    <span>Best Services</span>
    <h2>Explore top Deals.</h2>
  </div>
  <div class="service-container">
    {% for i in rent %}
    <div class="box">
      <div class="box-img">
        
      </div>
      <br>
      <p>{{i.Fueltype}}</p>
      <h3>{{i.Carname}}</h3>
      <h2>Rs{{i.Price}}<span>/Day</span></h2>
      <a href="{% url
'cardetails'%}?carid={{i.id}}" class="button"> Rent
Now</a>
    </div>
    {% endfor %}
  </div>
```

CODING

```
</section>

<section class="feedback" id="feedback">
  <div class="top-view">
    <span>Feedback</span>
    <h1>Best Customer Experience</h1>
  </div>
  <div class="feedback-container">
    <div class="feedback-img">
      
    </div>
    <div class="feedback-text">
      <span>Feedback</span>
      <p>I just wanted to compliment you on
your rental car service as we really appreciated the
seamless collect and return arrangements. We look
forward to 'traveling' with you in the spring</p>
      <p>I want you to know how delighted I am
with the car! It is really a pleasure to drive; I feel
very comfortable in it and find that it handles very
well. The extras like the heaters and leather seats make
a difference. I am so pleased</p>
      <a href="#" class="button">Read More</a>
    </div>
  </div>
</section>

<footer>
  <div class="foot-panel1">
    <li><a href="#header">Back to Home</a></li>
  </div>
  <div class="foot-panel2">
```


CODING

```
<ul>
  <p>Get to Know Us</p>
  <a>Careers</a>
  <a>Blog</a>
  <a>About DriveMeNow</a>
</ul>
<ul>
  <p>Make Money with Us</p>
  <a>Rent Car on DriveMeNow</a>
  <a>Advertise Your Products</a>
  <a>Self-Publish with Us</a>
</ul>

<ul>
  <p>Let Us Help You</p>
  <a>Your Account</a>
  <a>Your Orders</a>
  <a>24*7 Assistant</a>
  <a>Help</a>
</ul>
</div>
</footer>

<footer id="copyright">
  <h5>Made by Team B1(BCA)</h5><br>
  &#169; DRIVEMeNow, Inc. or its affiliates
</footer>
{% endblock %}
<!-- Link to Java script -->
<script src="script.js"></script>
```

CODING

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Login</title>
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/cs
s/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <!-- Login form -->
    <form class="col-6 mx-auto card p-3 shadow-lg"
method="post" enctype="multipart/form-data">
      <h1 style="text-align: center;"><span
style="color: rgb(25, 185, 145);">Please login
carefully</span></h1>

      {% csrf_token %} <!-- CSRF token for
security -->

      <!-- Login heading -->
      <h3>Login</h3>
      <hr>

      <!-- Display error/success messages -->
      {% if messages %}
      <div class="alert alert-primary"
role="alert">
        {% for message in messages %}
        {{ message }}
        {% endfor %}
```

CODING

```
</div>
{% endif %}

<!-- Username input field -->
<div class="form-group">
    <label
for="exampleInputEmail1">Username</label>
    <input type="text" class="form-control"
name="username" id="exampleInputEmail1" aria-
describedby="emailHelp"
placeholder="Enter username"
required>
</div>

<!-- Password input field -->
<div class="form-group">
    <label
for="exampleInputPassword1">Password</label>
    <input type="password" name="password"
class="form-control" id="exampleInputPassword1"
placeholder="Password" required>
</div>

<!-- Link to registration page -->
<p>Don't have an account <a href="{% url
'register_page' %}">Register</a> </p>

<!-- Submit button -->
<button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
</body>
</html>
```

CODING

Register.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <link
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/cs
s/bootstrap.min.css" rel="stylesheet">
  <title>Registration Form</title>
</head>
<body>
  <div class="container mt-5">
    <!-- Registration form -->
    <form class='{% url "register_page" %}'
method="POST" enctype="multipart/form-data">
      {% csrf_token %} <!-- CSRF token for
security -->

      <!-- Registration form heading -->
      <h1 style="text-align: center;"><span
style="color: rgb(29, 188, 182);">Please Sign-
Up</span></h1>
      <h3>Register</h3>
      <hr>

      <!-- Display error/success messages -->
      {% if messages %}
      <div class="alert alert-primary"
role="alert">
        {% for message in messages %}
        {{ message }}
```

CODING

```
        {% endfor %}
    </div>
{% endif %}

<!-- First Name input field -->
<div class="form-group">
    <label for="exampleInputEmail1">First
name</label>
    <input type="text" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp"
placeholder="Enter First name"
name="first_name" required>
</div>

<!-- Last Name input field -->
<div class="form-group">
    <label for="exampleInputEmail1">Last
name</label>
    <input type="text" name="last_name"
class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp"
placeholder="Enter Last name"required>
</div>

<!-- Username input field -->
<div class="form-group">
    <label
for="exampleInputEmail1">Username</label>
    <input type="text" class="form-control"
name="username" id="exampleInputEmail1"
aria-describedby="emailHelp"
placeholder="Enter username" required>
</div>

<!-- Password input field -->
```

CODING

```
<div class="form-group">
  <label
for="exampleInputPassword1">Password</label>
  <input type="password" class="form-
control" name="password" id="exampleInputPassword1"
placeholder="Password" required>
</div>

<div class="form-group">
  <label
for="exampleInputPassword2">Confirm-Password</label>
  <input type="password" class="form-
control" name="confirm_password"
id="exampleInputPassword2"
placeholder="Password" required>
</div>

<!-- Link to login page for users who
already have an account -->
<p>Already have an account <a href="{% url
'login_page' %}">Login</a> </p>

<!-- Submit button -->
<button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
</body>
</html>
```

CODING

Cardetails.html

```
{% extends 'main.html'%}
```

```
{% block style %}
```

```
<style>
```

```
    .cardetailsbox{
        /* background-color: rgb(72, 80, 80); */
        display: flex;
        padding: 2rem;
        background-color: rgba(9, 224, 191, 0.671);
        /* justify-content: space-between; */
    }
    .cardetailsleft img{
        width: 35rem;
    }
    .carddetailsright{
        margin-left: 3rem;
        border-radius: 5%;
    }
}
```

```
</style>
```

```
{% endblock %}
```

```
{% block body %}
```

```
<div id="cardetailsbody">
```

```
    {% for i in car %}
    <div class="cardetailsbox">
```

CODING

```
<div class="cardetailsleft">
    
</div>
<div class="carddetailsright">
    <h2 class="card-title">{{i.Carname}}</h2>
    <h5>Fuel type: {{i.Fueltype}}</h5>
    <h5>RS: {{i.Price}} Per Day</h5>
    <!-- <h5>Contact_no :{{i.phone}}</h5> -->
    <h5>Details :</h5>
    <p>{{i.Details}}</p>
    <a href="{% url 'home'%}?carbook={{i.id}}"
class="btn btn-secondary">Back</a>
    <a href="{% url
'checkout1'%}?carbook={{i.id}}&carprice={{i.Price}}"
class="btn btn-primary">Rent Now</a>
</div>
</div>
{% endfor%}
</div>

{% endblock %}
```

Checkout.html

```
{% extends 'main.html'%}
```

```
{% block style %}
```

```
<style>
```


CODING

```
/* General styles */
#checkoutbody {
  width: 60%;
  margin: 50px auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  background-color: rgba(90, 216, 210, 0.8);
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

/* Style for headings */
h1 {
  font-size: 24px;
  color: #333;
}

/* Style for form labels and inputs */
form div {
  margin-bottom: 15px;
}

label {
  font-weight: bold;
}

input[type="text"],
input[type="date"],
textarea,
select {
  width: 100%;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}
```

CODING

```
}

textarea {
  height: 100px;
  resize: vertical; /* Allow vertical resize */
}

/* Style for buttons */
.btn {
  display: inline-block;
  padding: 8px 16px;
  font-size: 16px;
  text-align: center;
  text-decoration: none;
  border-radius: 4px;
  cursor: pointer;
}

.btn-secondary {
  background-color: #ccc;
  color: black;
  margin-right: 10px;
}

.btn-primary {
  background-color: #007bff;
  color: #fff;
}

.btn-primary:hover {
  background-color: #0056b3;
}

</style>
```

CODING

```
{% endblock %}
```

```
{% block body %}
```

```
<div id="checkoutbody">
  <div id="heading1">
    <h1>Checkout</h1>
  </div>
  <div id="form1">
    <form action="{% url 'checkout1' %}"
method="POST">
      {% csrf_token %}
      <input type="hidden" name="price"
value="{{price}}">
      <input type="hidden" name="carid"
value="{{carid}}">
      <div>
        <label for="name">Name:</label>
        <input type="text" id="name"
name="name">
      </div>
      <div>
        <label for="email">Email:</label>
        <input type="text" id="email"
name="email">
      </div>
      <div>
        <label for="number">Phone:</label>
        <input type="text" id="phone"
name="phone">
      </div>
      <div>
        <label for="text">Driving Licence
No:</label>
```

CODING

```
        <input type="text" id="licence"
name="licence">
    </div>
    <div>
        <label for="address">Address:</label>
        <textarea name="Address" id="Address"
rows="3" cols="12">Please Enter your address</textarea>
    </div>
    <div>
        <label for="booking">Booking
Date:</label>
        <input type="date" id="Booking-date"
name="bookingdate">
    </div>
    <div>
        <label for="return">Booking
Location:</label>
        <input type="text" id="Booking-location"
name="bookinglocation">
    </div>
    <div>
        <label for="return">Return Date:</label>
        <input type="date" id="Return-date"
name="returndate">
    </div>
    <div>
        <label for="return">Destination
Location:</label>
        <input type="text" id="destination-
location" name="destinationlocation">
    </div>
    <div>
        <label for="text">City:</label>
        <input type="text" id="city"
name="city">
```

CODING

```
</div>

<div>
  <label for="text">State:</label>
  <select name="State" id="State">
    <option value="Bihar">Bihar</option>
    <option
value="Kolkata">Kolkata</option>
    <option
value="Chennai">Chennai</option>
    <option value="New Delhi">New
Delhi</option>
    <option
value="Mumbai">Mumbai</option>
  </select>
</div>
<div>
  <label for="number">Pincode:</label>
  <input type="text" id="pincode"
name="pincode">
</div>
<div>
  <label for="text">Country:</label>
  <input type="text" id="country"
name="country">
</div>
<a href="{% url
'cardetails'%}?carbook={{i.id}}" class="btn btn-
secondary">Back to Previous Page</a>
  <button type="submit" class="btn btn-
primary">Submit</button>
</form>
</div>
</div>
```

CODING

```
{% endblock %}
```

Payment.html

```
{% extends 'main.html' %}
```

```
{% block style %}
```

```
<style>
```

```
    body {
```

```
        font-family: Arial, sans-serif;
```

```
        line-height: 1.6;
```

```
        background-color: #f9f9f9;
```

```
        color: #333;
```

```
        padding: 20px;
```

```
    }
```

```
    .container {
```

```
        max-width: 800px;
```

```
        margin: 0 auto;
```

```
        background-color: #fff;
```

```
        padding: 20px;
```

```
        border-radius: 5px;
```

```
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
    }
```

```
    h2 {
```

```
        text-align: center;
```

```
        color: #007bff;
```

```
    }
```

```
    .payment-details {
```

```
        margin-top: 20px;
```

CODING

```
}

.payment-details h1 {
  font-size: 24px;
  margin-bottom: 10px;
}

.payment-details h1,
.payment-details h2 {
  text-align: center;
  color: #333;
}

.btn-container {
  text-align: center;
  margin-top: 30px;
}

.btn {
  display: inline-block;
  padding: 10px 20px;
  margin-right: 10px;
  text-decoration: none;
  color: #fff;
  background-color: #007bff;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.btn-secondary {
  background-color: #6c757d;
}
```

CODING

```
.btn-primary:hover,
.btn-secondary:hover {
    background-color: #0056b3;
}
</style>
{% endblock %}

{% block body %}
<div class="container">
    <h2>Welcome to Payment Page</h2>

    <div class="payment-details">
        <h1>Number of Days: {{ days }}</h1>
        <h1>Price per Day: Rs: {{ price }}</h1>
        <h1>Total Amount: Rs: {{ amount }}</h1>
        <h1><strong>Payment Method:</strong> Cash on
Delivery</h1>
    </div>

    <div class="btn-container">
        <!-- <a href="{% url 'checkout1' %}" class="btn
btn-secondary">Back to Previous Page</a> -->
        <a href="{% url 'order1'
%}?carid={{carid}}&days={{days}}&price={{price}}&amount=
{{amount}}" class="btn btn-primary">Confirm Payment</a>
    </div>
</div>

{% endblock %}
```

Order.html

```
{% extends 'main.html' %}
```


CODING

```
{% block style %}  
<style>  
  body {  
    font-family: Arial, sans-serif;  
    background-color: #f9f9f9;  
    color: #333;  
    box-shadow: aquamarine;  
    padding: 20px;  
  }  
  
  .container {  
    max-width: 600px;  
    margin: 0 auto;  
    background-color: #fff;  
    padding: 30px;  
    border-radius: 8px;  
    box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);  
    text-align: center;  
  }  
  
  h2 {  
    color: #007bff;  
    margin-bottom: 20px;  
  }  
  
  .billing-details {  
    margin-bottom: 30px;  
  }  
  
  .billing-details h3 {  
    font-size: 24px;  
    margin-bottom: 10px;  
    color: #333;  
  }  
</style>  
{% endblock %}
```

CODING

```
.billing-details p {
    font-size: 18px;
    margin-bottom: 10px;
}

.btn-container {
    text-align: center;
}

.btn {
    display: inline-block;
    padding: 12px 24px;
    margin-right: 10px;
    text-decoration: none;
    color: #fff;
    background-color: #007bff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

.btn:hover {
    background-color: #0056b3;
}
</style>
{% endblock %}

{% block body %}
<div class="container">
    <h2>Booking Confirmed!</h2>

    <div class="billing-details">
        <h3>Booking Details</h3>
```

CODING

```
{% for i in cardata %}
<p><strong>Car Name:</strong> {{ i.Carname
}}</p>
{% endfor %}
<p><strong>Number of Days:</strong> {{ days
}}</p>
<p><strong>Price per Day:</strong> Rs: {{ price
}}</p>
<p><strong>Total Amount:</strong> Rs: {{ amount
}}</p>
<p><h2>Your car has been successfully
booked!</h2></p>
<p><h3>Please pay the amount in cash upon
delivery.</h3></p>
</div>

<div class="btn-container">
    <a href="{% url 'home' %}" class="btn btn-
primary">Return to Home</a>
</div>
</div>
{% endblock %}
```

mystyle.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    scroll-padding-top: 2rem;
    scroll-behavior: smooth;
    list-style: none;
    text-decoration: none;
    font-family: 'Poppins', sans-serif;
```

CODING

```
}

section {
  padding: 50px 100px;
}

.view{
  width: 100%;
  min-height: 100vh;
  position: relative;
  background-image: url(images/Defender02.jpg);
  background-color: rgb(195, 222, 227);
  background-repeat: no-repeat;
  background-position:center right;
  display: grid;
  grid-template-columns: repeat(2, 1fr);
}

.text h2 {
  font-size: 3.4rem;
  letter-spacing: 2px;
}

.text span {
  color: rgb(21, 166, 214);
  text-shadow: 2px 2px 5px black;
}

.text p {
  font-size: 1.2rem;
  margin: 00.5rem 0 1rem;
}

.top-view {
  text-align: center;
```

CODING

```
}  
.top-view span {  
  font-weight: 500;  
  text-transform: uppercase;  
}  
  
.top-view h2 {  
  font-size: 2.0rem;  
}  
  
.travel-container {  
  display: grid;  
  align-items: center;  
  grid-template-columns: repeat(auto-fit,minmax(250px,  
auto));  
  gap: 1.1rem;  
  margin-top: 2.5rem;  
}  
  
.travel-container .box {  
  text-align: center;  
  padding: 20px;  
}  
  
.travel-container .box .fa-solid {  
  font-size: 35px;  
  padding: 10px;  
  background: white;  
  border-radius: 0.8rem;  
  color: #0dacc9;  
}  
  
.travel-container .box h2 {  
  font-size: 1.5rem;  
  font-weight: 500;  
  margin: 1.4rem 0 0.5rem;  
}
```

CODING

```
.service-container {  
  display: grid;  
  grid-template-columns: repeat(auto-fit,minmax(280px,  
auto));  
  gap: 1.8rem;  
  margin-top: 2rem;  
}
```

```
.service-container .box {  
  padding: 10px;  
  border-radius: 1.5rem;  
  box-shadow: 1px 4px 44px rgba(26, 81, 147, 0.662);  
}
```

```
.service-container .box .box-img {  
  width: 100%;  
  height:200px;  
}
```

```
.service-container .box .box-img img {  
  width: 100%;  
  height: 100%;  
  border-radius: 1.5rem;  
  object-fit: cover;  
  object-position: center;  
}
```

```
.service-container .box p {  
  padding: 0 10px;  
  border: 1px solid #0dacc9;  
  width: 60px;  
  border-radius: 0.8rem;  
  margin: 1rem 0.5 1.5rem;  
}
```

CODING

```
.service-container .box h3 {
  font-weight: 500;
}

.service-container .box h2 {
  font-size: 1.5rem;
  font-weight: 600;
  color: rgb(42, 79, 56);
  margin: 0.5rem 0 0.5rem;
}

.service-container .box h2 span {
  font-size: 1.0rem;
  font-weight: 500;
  color:rgb(42, 79 , 56);
}

.service-container .box .button {
  display: flex;
  justify-content: center;
  background: #474fa0;
  color: white;
  padding: 10px;
  border-radius: 0.8rem;
}

.feedback-container {
  display: grid;
  grid-template-columns: repeat(3,2fr);
  margin-top: 2rem;
  align-items: center;
  gap: 1rem;
}

.feedback-img {
```

CODING

```
    width: 100%;
}

.feedback-text span {
    font-weight: 500;
    color: rgb(245, 59, 46);
    text-transform: uppercase;
}

.feedback-text p {
    margin: 0.5rem 0 1.5rem ;
}

.feedback-text .button {
    padding: 10px 18px;
    background: #474fa0;
    color: white;
    border-radius: 0.8rem;
}

footer {
    margin-top: 5px;
}

.foot-panel1 {
    background-color: rgb(232, 177, 177);
    color: white;
    height: 50px;
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 0.85rem;
}

.foot-panel1 a {
    color: black;
    font-weight: bold;
}
```


CODING

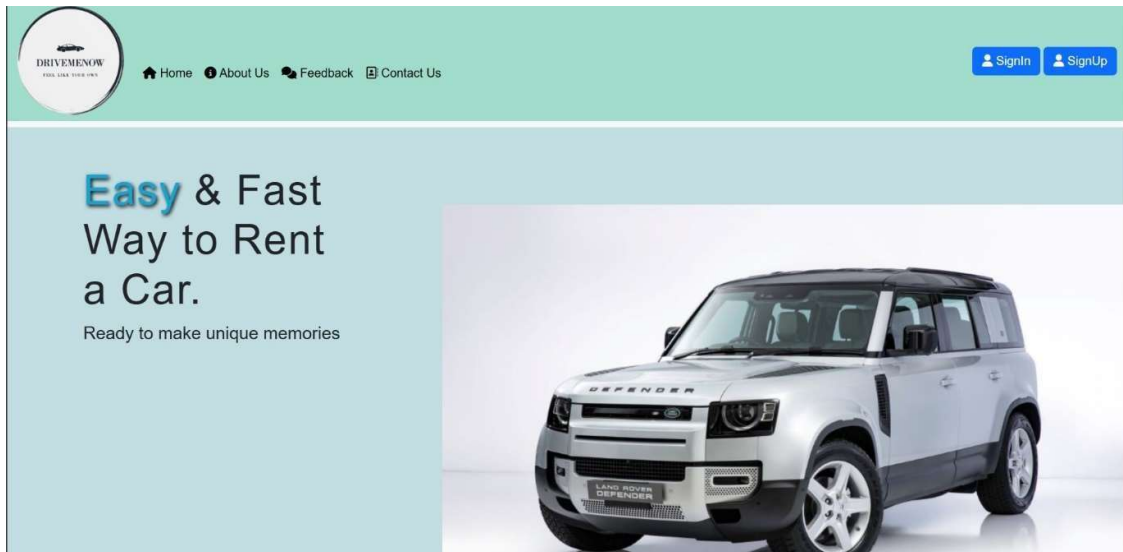
```
.foot-panel2 {
  background-color: rgb(164, 231, 237);
  color: black;
  height: 250px;
  display: flex;
  font-weight: bold;
  justify-content: space-around;
}
ul {
  margin-top: 22px;
}

ul a {
  display: block;
  font-size: 0.85rem;
  margin-top: 10px;
  color: black;
}

#copyright {
  background-color: rgba(26, 81, 147, 0.662);
  color: white;
  text-align: center;
  border-radius: 8px;
  border-color: aqua;
}
```

SAMPLE SCREENSHOT

Main Page



Login Page

Please login carefully

Login

Username

Password

Don't have an account

[Register](#)

Submit

SAMPLE SCREENSHOT

Registration Page

Please Sign-Up

Register

First name

Enter First name

Last name

Enter Last name

Username

Enter username

Password

Password

Confirm-Password

Password

Already have an account [Login](#)

Submit

All Cars Page

BEST SERVICES
Explore top Deals.



Diesel

Maruti-Suzuki Swift

Rs1800/Day

Rent Now



Petrol

Renault-Kwid

Rs2000/Day

Rent Now



Diesel

Mahindra-Thar

Rs3000/Day

Rent Now



Diesel

Toyota-Fortuner

Rs8000/Day

Rent Now



Diesel

Mahindra Scorpio N

Rs5000/Day

Rent Now



Petrol

Land-Rover Defender

Rs12000/Day

Rent Now



Diesel

Audi Q7

Rs15000/Day

Rent Now



Petrol

BMW 5 Series

Rs16000/Day

Rent Now

CAR DETAILS PAGE



[Home](#) [About Us](#) [Feedback](#) [Contact Us](#)

[SignIn](#)

[SignUp](#)



BMW 5 Series

Fuel type: Petrol

RS: 16000 Per Day

Details :

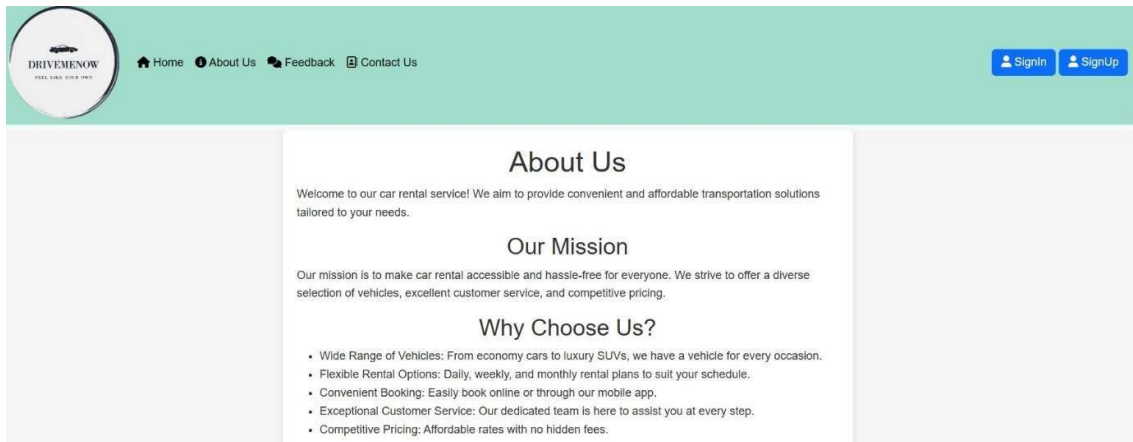
The BMW 5 Series is a midsize luxury sedan that can seat up to five passengers. It is a sporty and comfortable business sedan that combines modern design, technological innovations, and cThe 5 Series has a 2.0-liter, four-cylinder petrol and diesel engine, as well as a 3.0-liter, six-cylinder diesel engine. It has an 8-speed automatic transmission, adaptive suspension, six airbags, ABS with BA, DTC, and many other safety features. The 5 Series is 4,963 mm long, 1,868 mm wide, and has a wheelbase of 2,975 mm. It has an average claimed mileage of 14.8-20.3 kmpl, depending on the fuel.omfort features.

[Back](#)

[Rent Now](#)

SAMPLE SCREENSHOT

About Us Page



Contact page



Checkout Page



Checkout

Name:

Email:

Phone:

Driving Licence No:

Address:
Please Enter your address

Booking Date:
dd-mm-yyyy

Booking Location:

Return Date:
dd-mm-yyyy

Destination Location:

City:

State:

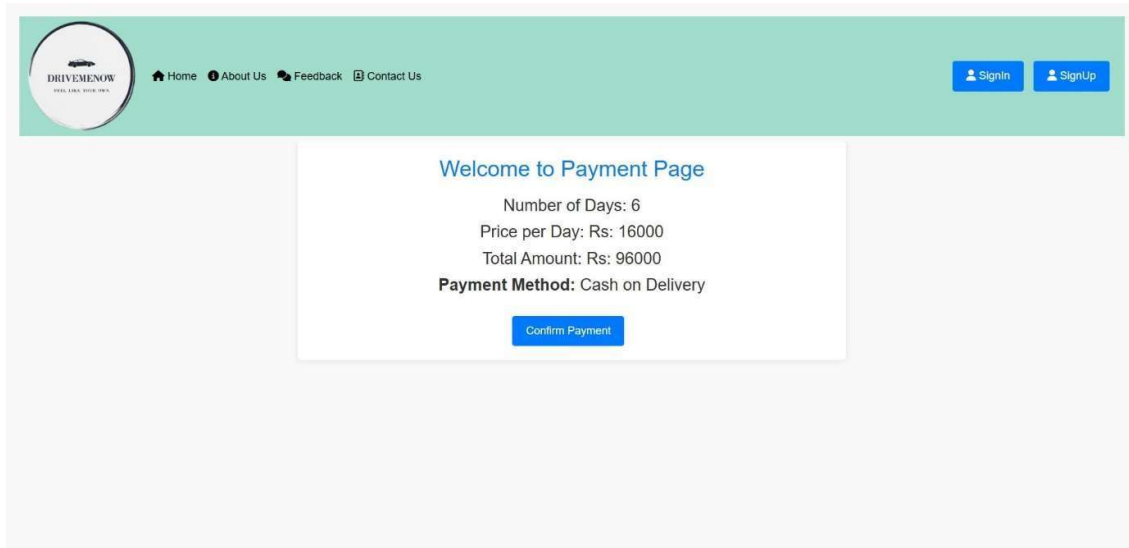
Pincode:

Country:

[Back to Previous Page](#) [Submit](#)

SAMPLE SCREENSHOT

Payment Page



TEST CASES

1. User Registration :

- Test user registration with valid credentials.
- Test user registration with invalid credentials (e.g., incomplete form, invalid email format).
- Verify that passwords meet security requirements (e.g., minimum length, special characters).

3. Booking Process :

- Test booking a car with valid details.
- Test booking a car with invalid details (e.g., incomplete form, invalid payment information).
- Verify that the booking confirmation page displays correct information.

4. User Account Management :

- Test updating user profile information.
- Verify that users can view their booking history.

5. Payment Processing :

- METHOD CASH ON DELIVERY.
- Verify that users receive a confirmation email after successful payment.
- Verify that bookings are not confirmed if payment fails.

6. Admin Panel :

- Test login to the admin panel with valid credentials.
- Test login with invalid credentials.
- Verify that admin can add new cars to the inventory.
- Verify that admin can edit existing car details.
- Verify that admin can view booking details and modify or cancel bookings if necessary.

8. Error Handling :

- Test error messages for various scenarios (e.g., server errors, invalid input).

- Verify that error messages are displayed appropriately and help users understand how to resolve issues.

9. Performance Testing :

- Test the website's performance under different load conditions (e.g., simultaneous searches, bookings).
- Verify that the website responds quickly and efficiently.

10. Compatibility Testing :

- Test the website on different browsers (e.g., Chrome, Firefox, Edge).
- Test the website on different devices (e.g., desktop, tablet, mobile).
- Verify that the website is responsive and functions correctly across various platforms.

CONCLUSION

In conclusion, the test cases for a car rental website are crucial for ensuring its functionality, usability, and reliability. By thoroughly testing various aspects of the website, including user registration, booking process, user account management, payment processing, admin panel functionality, error handling, performance, and compatibility, we can ensure that the website meets the needs of both users and administrators.

A comprehensive set of test cases helps identify and address any issues or bugs in the system, leading to a smoother user experience and increased customer satisfaction. Additionally, proper testing ensures that the website performs efficiently under different load conditions and across various platforms, enhancing its accessibility and usability.

By conducting thorough testing and addressing any identified issues, the car rental services website can maintain its reputation for reliability, security, and user-friendliness, ultimately leading to the success and growth of the business.

LIMITATION

1. **Geographical Restrictions** : Car rental services may have limitations on where vehicles can be picked up or dropped off. This can restrict users who need cars in areas where the service is unavailable.

2. **Damage and Liability** : Users may face challenges in understanding the liability and insurance coverage in case of accidents or damages to the rented vehicle. Unclear terms and conditions can lead to disputes and dissatisfaction.

3. **Costs and Fees** : Hidden costs such as insurance fees, fuel charges, additional driver fees, and late return fees can significantly increase the overall rental cost. Users may find it challenging to estimate the total expense accurately.

6. **Dependency on Internet Connectivity**: The reliance on internet connectivity for booking, payment processing, and accessing reservation details can be a limitation, especially in areas with poor network coverage or during network outages.

7. **Security Concerns** : Storing personal and financial information on the website poses security risks, such as data

breaches or unauthorized access. Users may hesitate to provide sensitive information due to security concerns.

10. Competition and Pricing : Intense competition among car rental services can lead to pricing wars and aggressive marketing tactics. Users may face difficulty in navigating through numerous options and finding the best value for their money.

Despite these limitations, car rental services continue to evolve and improve, addressing user concerns and enhancing the overall experience through innovations in technology, customer service, and sustainability initiatives.

REFERENCES

The Following sites where referred during the analysis and execution phase of this project.

SITE REFERRED :

1. [W3school.com](https://www.w3school.com/)
2. [Stackoverflow.com](https://stackoverflow.com/)
3. [Bootstrap.com](https://getbootstrap.com/)
4. [Youtube.com](https://www.youtube.com/)
5. [Geeksforgeeks.org](https://www.geeksforgeeks.org/)