
**Dharmsinh Desai University, Nadiad**
**Faculty of Technology**
**Department of Computer Engineering**


**B. Tech. CE Semester – 4**

**Subject - Software Project**

# Project Title

## Travelling Package Booking System


**Gajera Jenil V. (CE032)**
**Chovatiya Kaushal K.(CE021)**

**Guide by: Prof. Pinkal Chavda & prof.Brijesh Bhatt**
**Assistant professor**
**Dept. of Computer Engineering**

# INDEX

# 1.Introduction/Project Abstract

- Our project is about online bus ticket booking. In this Project admin can add place, price and description of that place. For this we have to create superuser of project. By logging into admin site he can add those things.

- Simple User can register and logging. After logging he/she can see places, prices and description added by admin. By selecting available place the user have to logging if he/she has not logging. After that payment card will open to make payment.

- We have also provided Mobile number at home page for contact.

- If any offer is available at any place it is also visible.

# 2. Software Requirement Specification

1. **create customer id:**

**R1.1: first enter the name and id**

Description: in first open the link and enter the name, id, number and other details

Input: enter the name and id

Output: proper name and id

**R1.2: confirm account**

Input: need data feel

Output: see your account and check correct

2. **process of ticket and route detail**

Description: after our system work to see detail of bus and route whatever they enter. Every available bus saw and seat is saw. seat reservation is like any different colour .

**R2.1: you decide where you go for ride**

Input: think types of bus and enter

Output: show available types of bus

**R2.2: check for your place's bus available**

Input: your first point place and last place for journey

Output: if bus available, saw all buses otherwise no buses.

3. **as per customer's like bus provide**

Description: when passenger select bus and enter the bus next select seat which you like you choose. You choose select 1,2 or many seat and next again saw your first point .And last point .

**R3.1: bus type seater, sleeper, both etc**

Input: your choice

Output: if available this type bus ,so see detail

**R3.2: again check your boarding and departure place and their time**

Input: only do continue

Output: your enter place are saw and once check

4. **advance booking**

Description: this advance booking one of the most benefit for your side. Some time in traveling whole bus fill and no place in bus. This time your booking is bless and you got a seat confirm

**R4.1: what is require advance booking**

Input: your booking seat

Output: you give a seat for journey confirm

**R4.2: benefit of advance booking**

Input: you enter the bus

Output: always your selected seat empty any case your seat is confirmed.

**R4.3: advance booking minor charge in ticket**

5. **payment mode**

Description: when you check your detail and continue next payment mode.in payment options you using any net banking ,upi and patym like anything.when success payment your ticket confirm booked and you got msg from our operator.

**R5.1: select next step payment option**

Input: your require payment mode

Output: your select mode approve and next

**R5.2: next enter otp or pin for payment**

Input: enter detail

Output: your payment succes done

6. **confirmation details**

**R6.1: confirm booking msg**

Input: succes payment done and do continue

Output: confirm booking msg

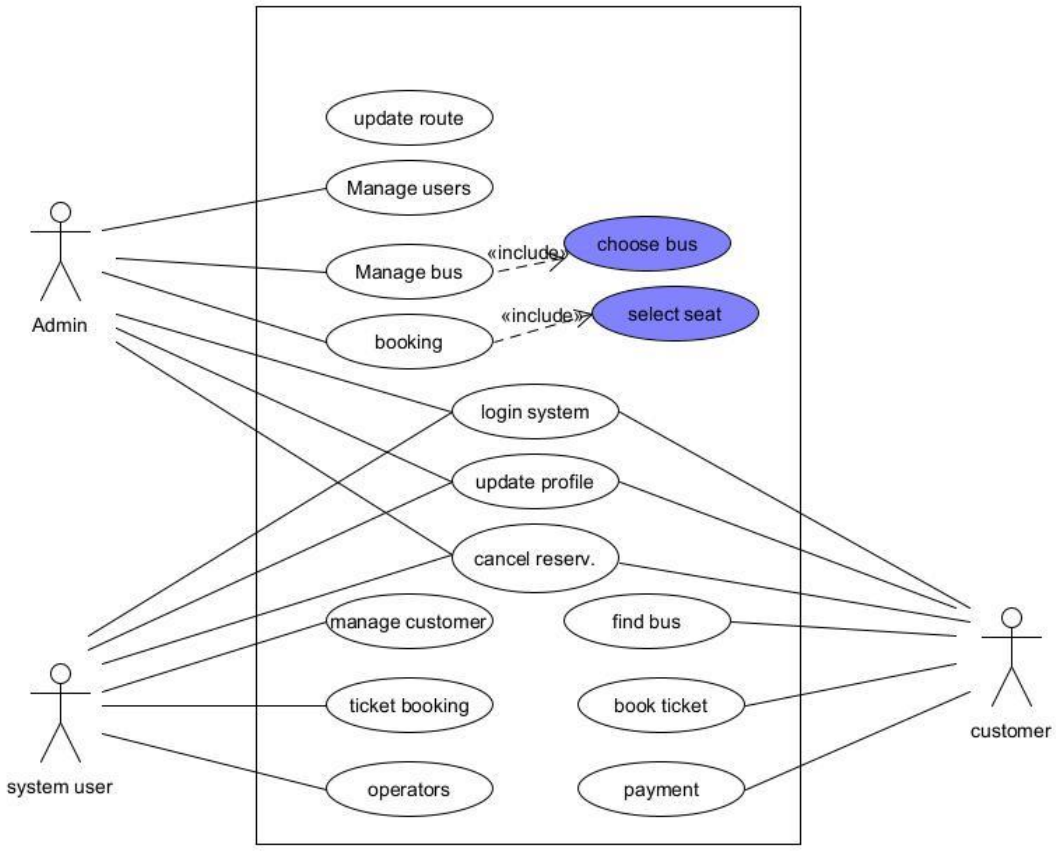
**R6.2: you got your ticket**

Input: next click continue

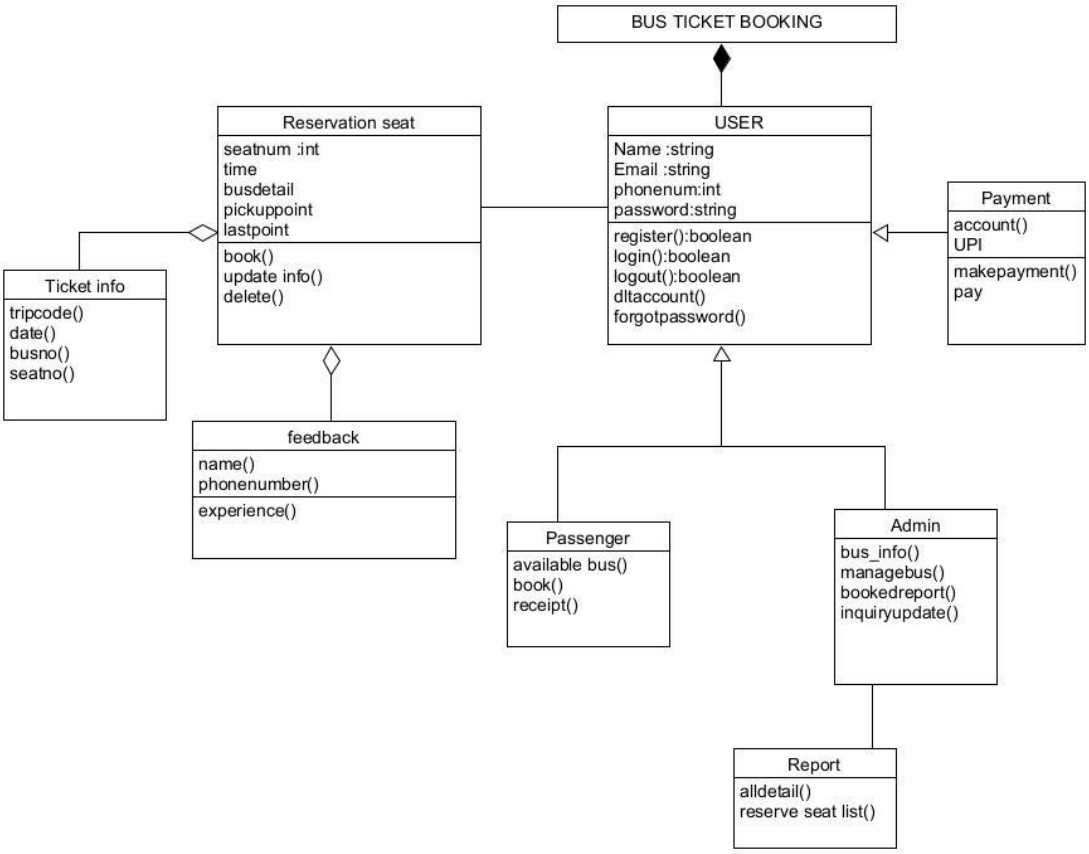
Output: via mail or msg you provide your ticket

# 3.Design Documents

- **Use case diagram**

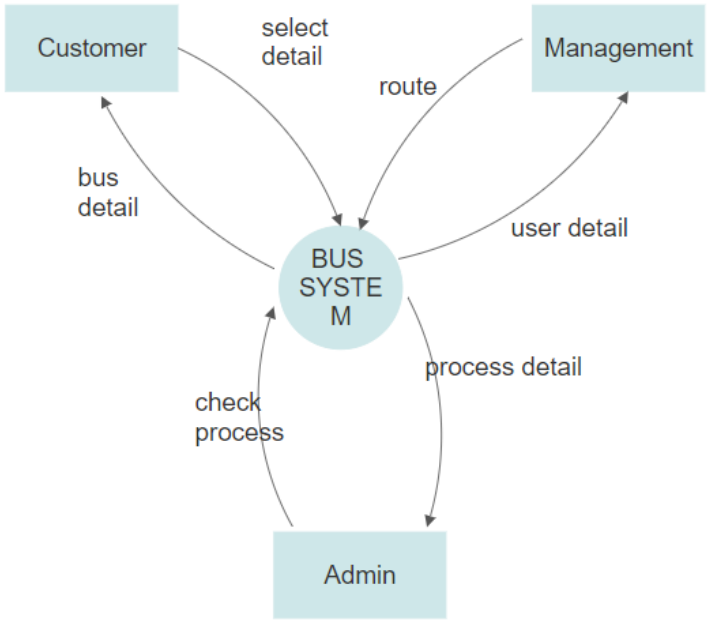



- **Class diagram**

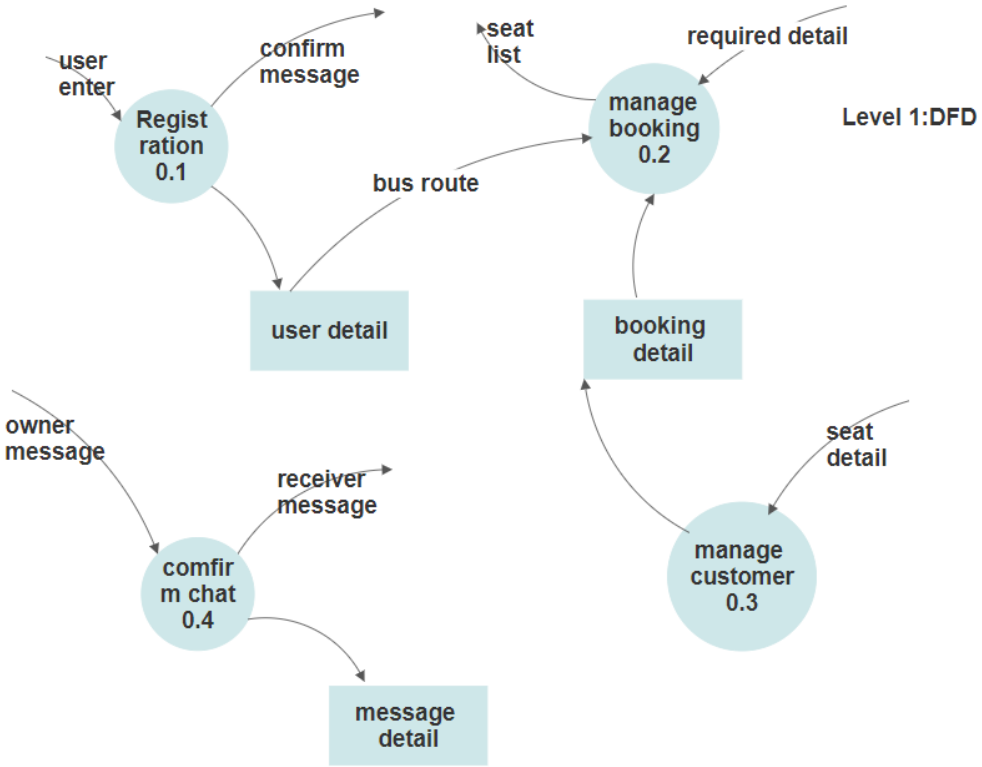

BUS TICKET BOOKING
Reservation seat
seatnum :int
time
busdetail
pickuppoint
lastpoint
book()
update info()
delete()
USER
Name :string
Email :string
phonenum:int
password:string
register():boolean
login():boolean
logout():boolean
dltaccount()
forgotpassword()
Payment
account()
UPI
makepayment()
pay
Ticket info
tripcode()
date()
busno()
seatno()
feedback
name()
phonenumber()
experience()
Passenger
available bus()
book()
receipt()
Admin
bus_info()
managebus()
bookedreport()
inquiryupdate()
Report
alldetail()
reserve seat list()

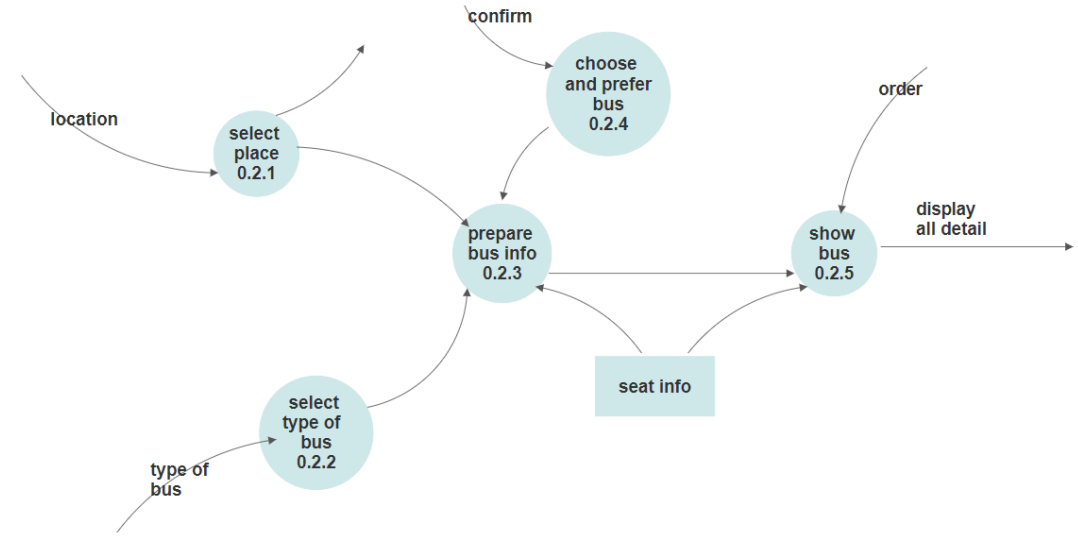
- **DFD Model**

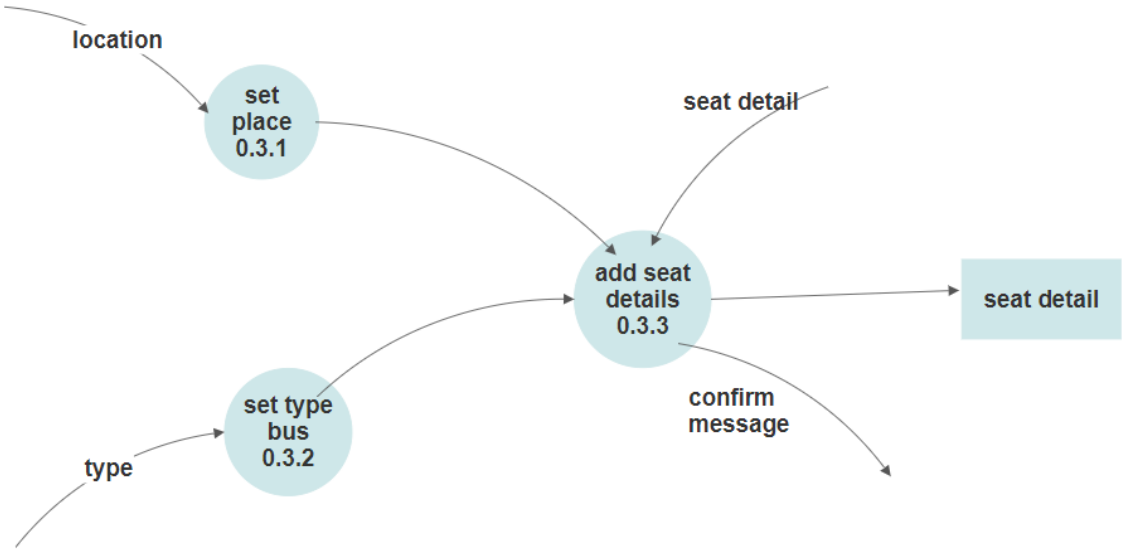
**Data flow diagram**




**Level 0: Context Diagram**

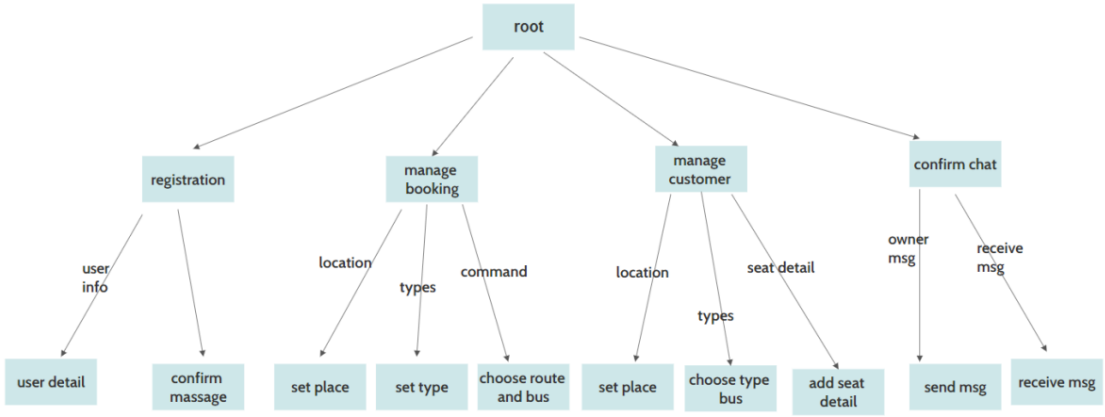




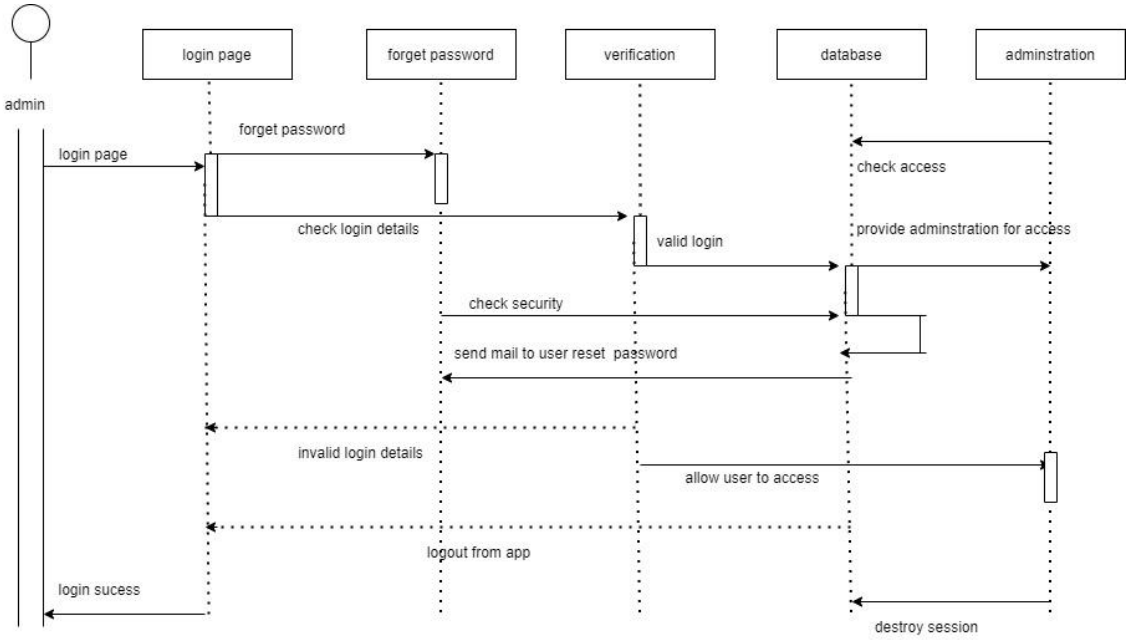




Level 2:DFD

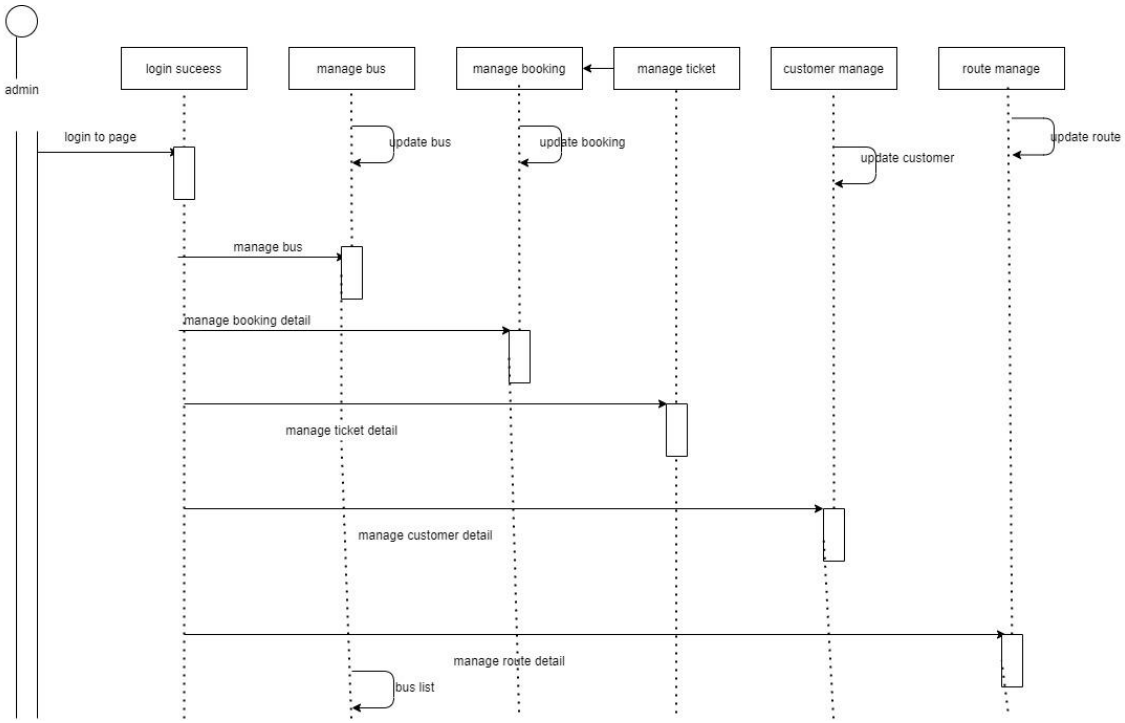
- **Structure chart**




- **Sequence diagram**

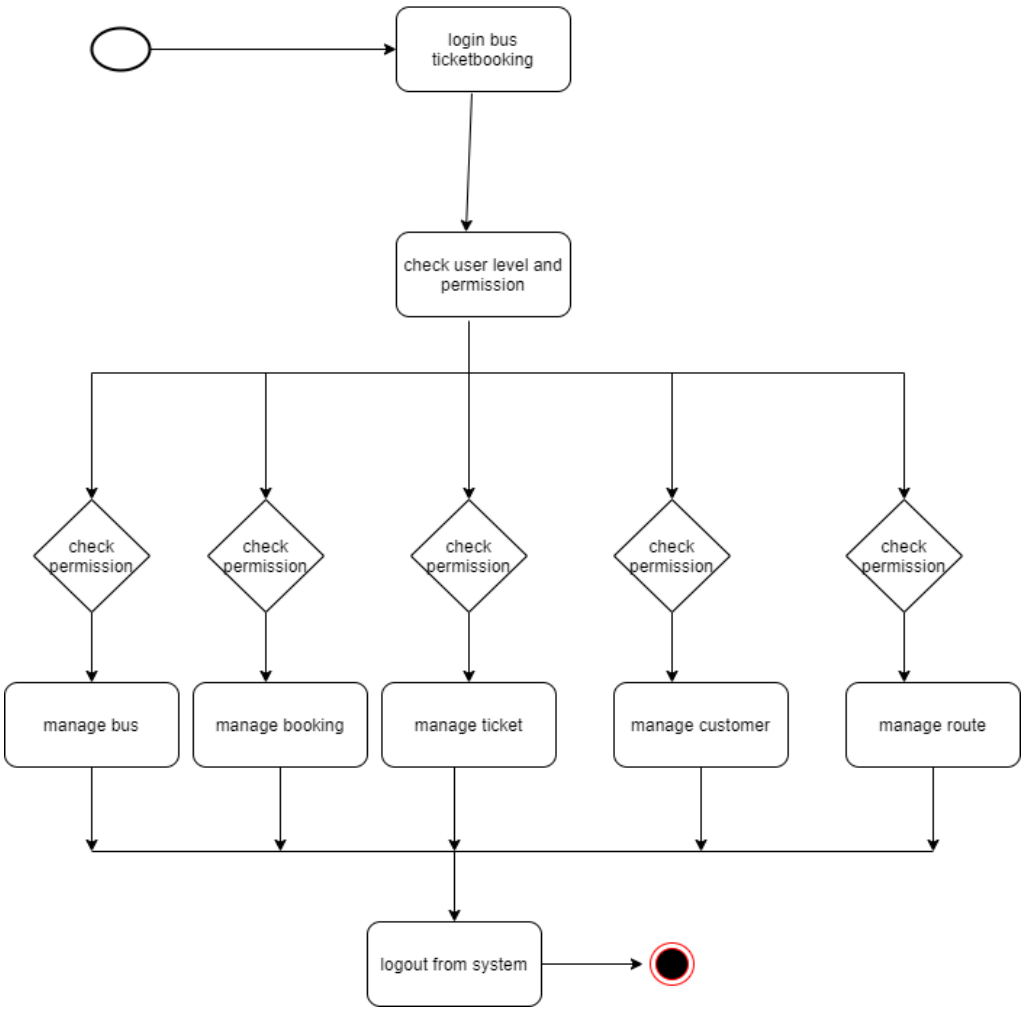


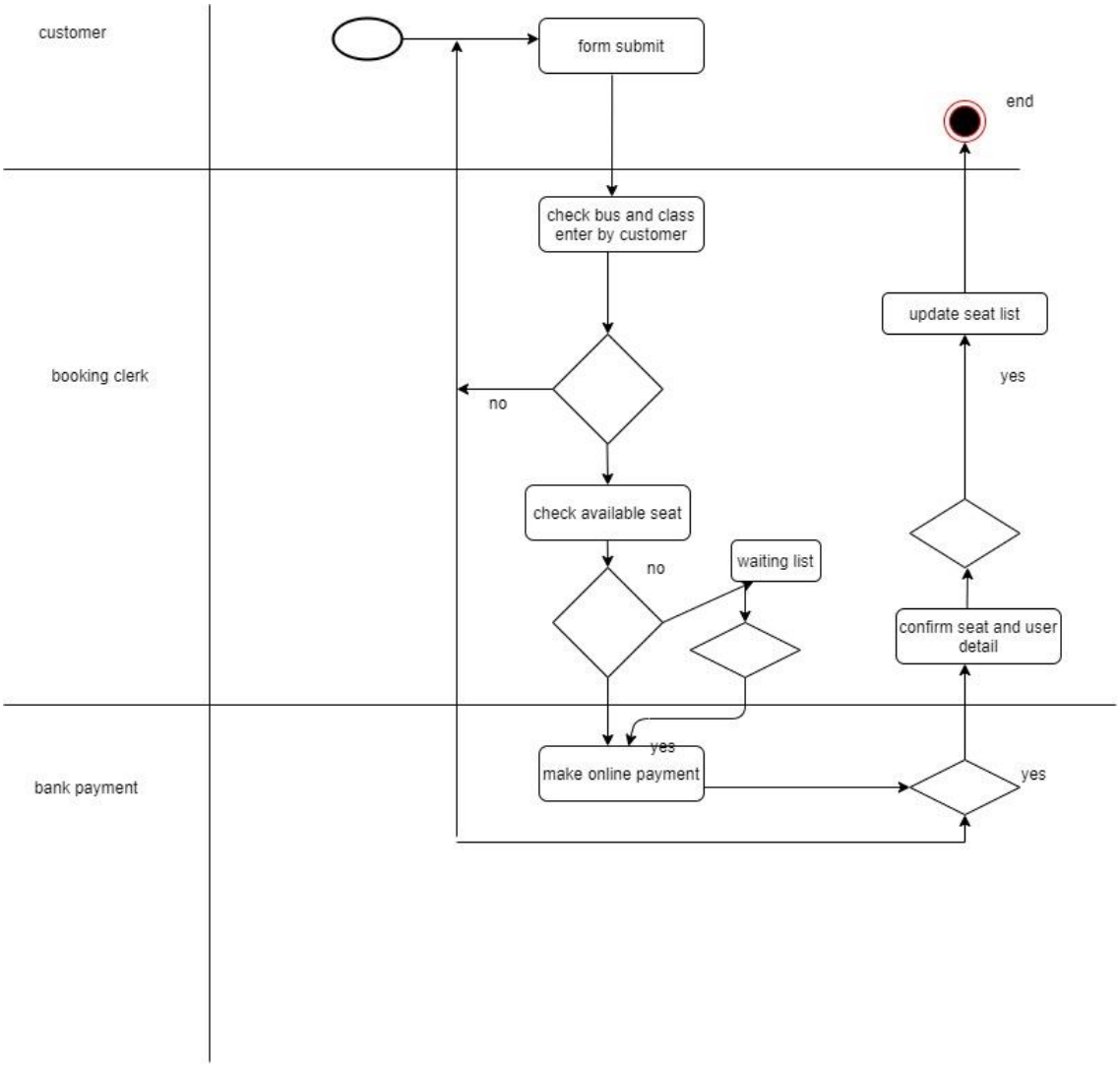

Sequence diagram




- **Activity Diagram**

Activity diagram






# 4.Implementation Details

- **Register Module:**
  In this module admin and customer can register themselves and can book ticket of their destination place and can use any other functionality.

register.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Registeration</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
    <script src='main.js'></script>
</head>
<body style="background-color:powderblue;" align="center">
    <form action="register" method="post" align="center">
        {% csrf_token %}
        <h1 style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Register</h1>
        <br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">First Name: </lable>  &
        emsp;  
        <input type="text" name="first_name" placeholder="First Name" style="font-size:150%;background-color:powderblue;color:#0033cc;
            font-family:verdana;"><br><br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Last Name: </lable>   
        ;  
        <input type="text" name="last_name" placeholder="Last Name" style="font-size:150%;background-color:powderblue;color:#0033cc;font
            -family:verdana;"><br><br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Username: </lable>   
            
        <input type="text" name="username" placeholder="Username" style="font-size:150%;background-color:powderblue;color:#0033cc;font
            -family:verdana;"><br><br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Email: </lable>   &
        emsp;      
        <input type="email" name="email" placeholder="email" style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:
            verdana;"><br><br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Password: </lable>   
           
        <input type="password" name="password1" placeholder="Password" style="font-size:150%;background-color:powderblue;color:#0033cc;
            font-family:verdana;"><br><br>
        <lable style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">Confirm Password: </lable>
        <input type="password" name="password2" placeholder="Confirm Password" style="font-size:150%;background-color:powderblue;color:#
            0033cc;font-family:verdana;"><br><br>
        <br>
        <input type="Submit" value="Register" style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;"><br><
        br>

    </form>

    <div align="center" style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">
        {% for message in messages %}
        <h3> {{message}}  </h3>
        {% endfor %}
    </div>

</body>
</html>
```

- Login Module
  In this module customer or admin can login and can do their own work.

login.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Login</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
    <script src='main.js'></script>
</head>
<body style="background-color:powderblue;">
    <form action="login" method="post" align="center">
        {% csrf_token %}
        <br><br>
        <lable style="font-size:150%;color:#0033cc;font-family:verdana;">Username: </lable><br>
        <input type="text" name="username" placeholder="Username" style="font-size:150%;"><br><br>
        <lable style="font-size:150%;color:#0033cc;font-family:verdana;">Password: </lable><br>
        <input type="password" name="password" placeholder="Password" style="font-size:150%;"><br><
        br>
        <input type="Submit" value="Login" style="font-size:150%;background-color:powderblue;color:#
            0033cc;font-family:verdana;"><br><br>
    </form>


    <div align="center" style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:
        verdana;">
            {% for message in messages %}
            <h3> {{message}}  </h3>
            {% endfor %}
    </div>
</body>
</html>
```

- Payment Module

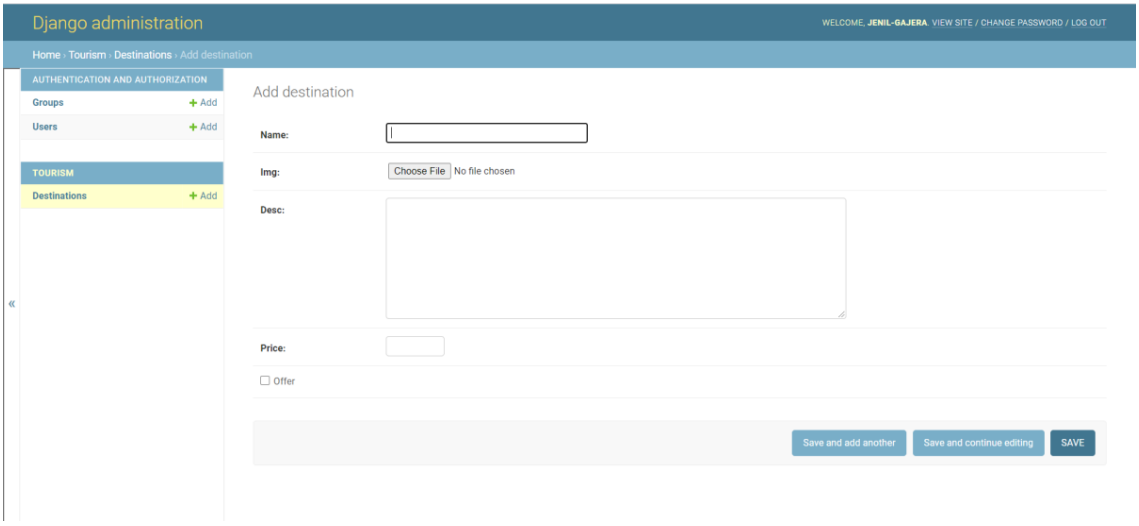
  In this module customer have to payment for ticket booking.

payment.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Payment</title>
</head>
<body style="font-size:150%;background-color:powderblue;color:#0033cc;font-family:verdana;">

    {% if user.is_authenticated %}
        <h1 align="center" style="font-size:150%;background-color:powderblue;color:#0033cc;font
            -family:verdana;">Hello, {{user.first_name}}</h1>
        <br>
        <div class="sign-up-form">
        <form action="paymentsuccessful" method="post" align="center" >
         {% csrf_token %}
            Card Number<br>
            <input type="Number" name="card_num" style="font-size:90%;" min="100000000000" max="
            999999999999" placeholder="Enter 12 digit card number" required><br><br>
            Expiry Date<br>
            <select name="month" style="font-size:90%;" required>
                <option value="">Month</option>
                <option value="Jan(01)">Jan(01)</option>
                <option value="Feb(02)">Feb(02)</option>
                <option value="Mar(03)">Mar(03)</option>
                <option value="Apr(04)">Apr(04)</option>
                <option value="May(05)">May(05)</option>
                <option value="Jun(06)">June(06)</option>
                <option value="Jul(07)">July(07)</option>
                <option value="Aug(08)">Aug(08)</option>
                <option value="Sept(09)">Sep(09)</option>
                <option value="Oct(10)">Oct(10)</option>
                <option value="Nov(11)">Nov(11)</option>
                <option value="Dec(12)">Dec(12)</option>
            </select>
            <select name="year" style="font-size:90%;" required>
                <option value="">Year</option>
                <option value="2021">2021</option>
```

```
                    <option value="2022">2022</option>
                    <option value="2023">2023</option>
                    <option value="2024">2024</option>
                    <option value="2025">2025</option>
                    <option value="2026">2026</option>
                    <option value="2027">2027</option>
                    <option value="2028">2028</option>
                    <option value="2029">2029</option>
                    <option value="2030">2030</option>
                </select>
                <br><br>
                CVV<br>
                <input type="Number" name="cvv"  style="font-size:90%;" max="999" min="100"
                placeholder="Enter 3 Digit CVV" required> <br><br>
                <input type="submit" style="font-size:90%;background-color:powderblue;color:#0033cc;
                    font-family:verdana;" value="Make Payment">
                <input type="button" style="font-size:90%;background-color:powderblue;color:#0033cc;
                    font-family:verdana;" onclick="location.href='/' "value="Cancel">
            </form>
            </div>

        {% else %}
            <h1 align="center" style="font-size:90%;background-color:powderblue;color:#0033cc;font
                -family:verdana;">Please Login or Register</h1><br>
            <li align="center" style="font-size:90%;background-color:powderblue;color:#0033cc;font
                -family:verdana;"><a href="reglog/register">Register</a></li>
            <li align="center" style="font-size:90%;background-color:powderblue;color:#0033cc;font
                -family:verdana;"><a href="reglog/login">Login</a></li>
        {% endif %}


    </body>
    </html>
```

- Add Destination
  This Module is for admin. Here admin can add destination, Photo of place, description and price and choose offer if he wants.



# 5.Testing

Testing is created for registration and login module.

**Test.py file**

```
from django.test import TestCase
from selenium import webdriver
from django.contrib.staticfiles.testing import StaticLiveServerTestCase
from django.urls import reverse
import time

class TestProjectListPage(StaticLiveServerTestCase):
    def setUp(self):
        self.browser = webdriver.Chrome('chromedriver.exe')

    def tearDown(self):
        self.browser.close()
```

**login_test**

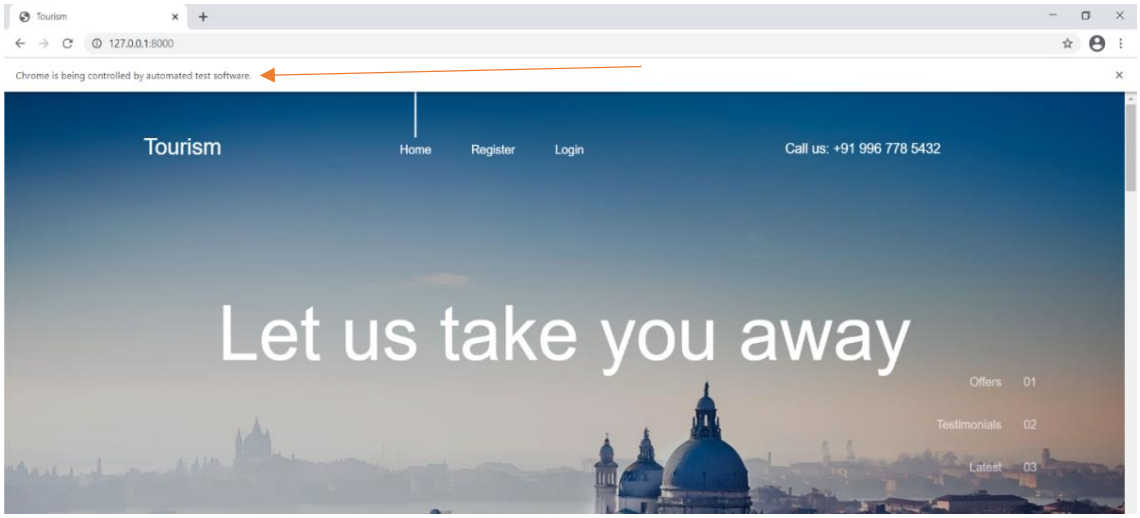
```
    def login_test(self):
        self.browser.get('http://127.0.0.1:8000/')
        self.browser.maximize_window()

        time.sleep(1)
        link = self.browser.find_element_by_link_text('Login')
        link.click()

        self.browser.find_element_by_name("username").send_keys("jenil1")
        time.sleep(1)
        self.browser.find_element_by_name("password").send_keys("123456")
        time.sleep(1)
        self.browser.find_element_by_name("submit").click()
        time.sleep(5)

```

**Registration_test**

```python
    def register_test(self):
        self.browser.get('http://127.0.0.1:8000/')
        self.browser.maximize_window()

        time.sleep(1)
        link = self.browser.find_element_by_link_text('Register')
        link.click()

        self.browser.find_element_by_name("first_name").send_keys("jenil")
        time.sleep(1)
        self.browser.find_element_by_name("last_name").send_keys("gajera")
        time.sleep(1)
        self.browser.find_element_by_name("username").send_keys("jenil2")
        time.sleep(1)
        self.browser.find_element_by_name("email").send_keys("j1@gmail.com")
        time.sleep(1)
        self.browser.find_element_by_name("password1").send_keys("123456")
        time.sleep(1)
        self.browser.find_element_by_name("password2").send_keys("123456")
        time.sleep(1)
        self.browser.find_element_by_name("submit").click()
        time.sleep(5)
```

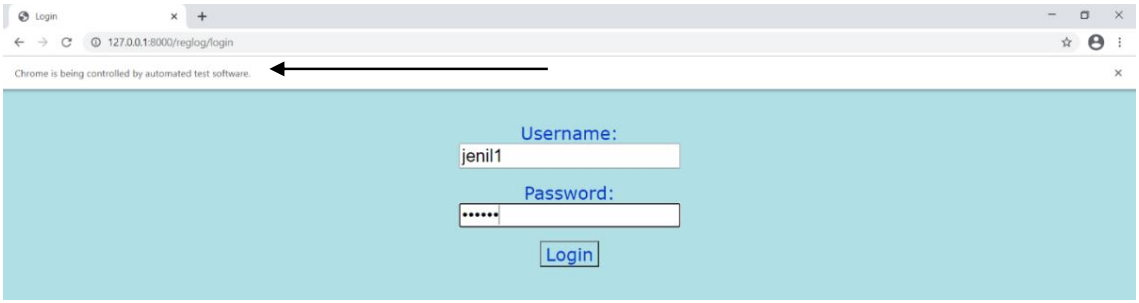
Testing is done for registration and login module and both test are properly working.

**Screen Shots** of running:



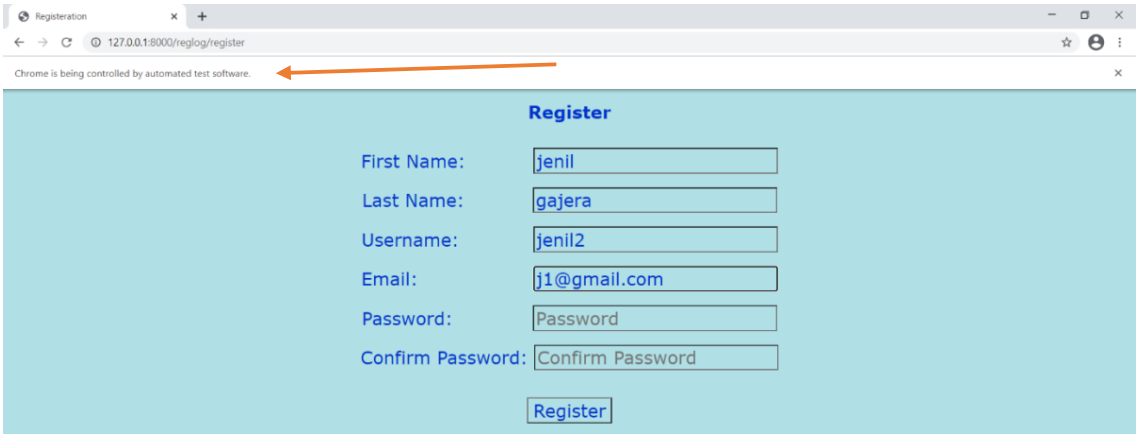
## Login Module:

```
C:\Users\Lenovo\Dairy-Management-System\SP Project\travellapp>python manage.py test tourism.tests.TestProjectListPage.login_test
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (mysql.W002) MariaDB Strict Mode is not set for database connection 'default'
        HINT: MariaDB's Strict Mode fixes many data integrity problems in MariaDB, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: htt
ps://docs.djangoproject.com/en/3.1/ref/databases/#mysql-sql-mode

System check identified 1 issue (0 silenced).

DevTools listening on ws://127.0.0.1:55248/devtools/browser/a4425c3a-38b1-483c-8bff-bec1d0c85a0f
[10004:11236:0402/123222.077:ERROR:device_event_log_impl.cc(214)] [12:32:22.077] USB: usb_device_handle_win.cc:1056 Failed to read descriptor from node connection: A device attached to the system is not function
ing. (0x1F)
[10004:11236:0402/123222.087:ERROR:device_event_log_impl.cc(214)] [12:32:22.087] USB: usb_device_handle_win.cc:1056 Failed to read descriptor from node connection: A device attached to the system is not function
ing. (0x1F)
.
----------------------------------------------------------------------
Ran 1 test in 11.145s

OK
```



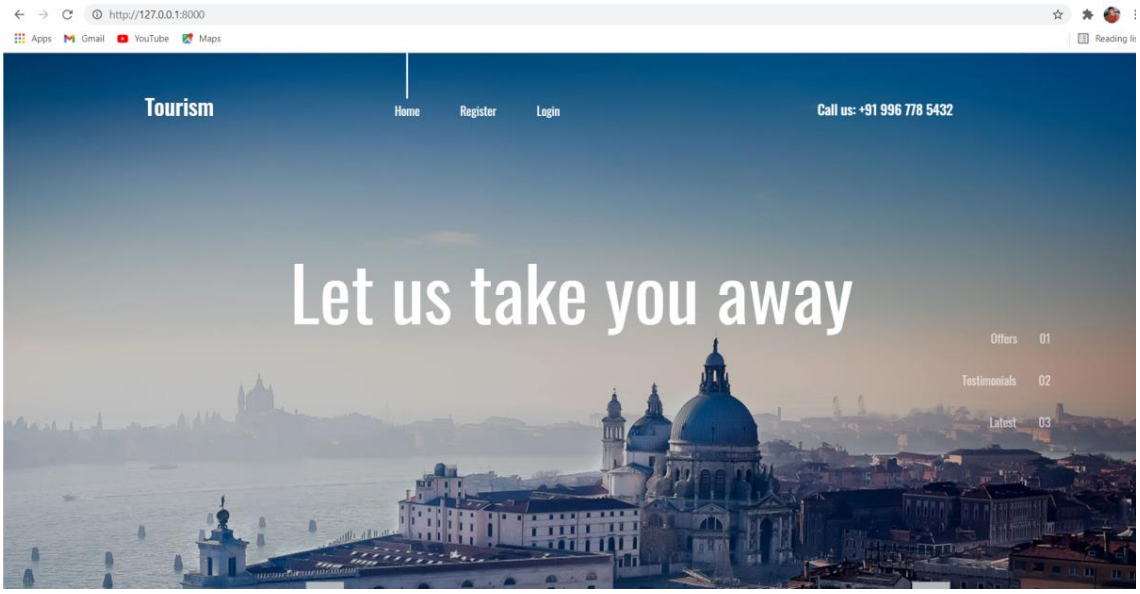

## Registration Module:

```
C:\Users\Lenovo\Dairy-Management-System\SP Project\travellapp>python manage.py test tourism.tests.TestProjectListPage.register_test
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (mysql.W002) MariaDB Strict Mode is not set for database connection 'default'
        HINT: MariaDB's Strict Mode fixes many data integrity problems in MariaDB, such as data truncation upon insertion, by escalating warnings into errors. It is strongly recommended you activate it. See: htt
ps://docs.djangoproject.com/en/3.1/ref/databases/#mysql-sql-mode

System check identified 1 issue (0 silenced).

DevTools listening on ws://127.0.0.1:55315/devtools/browser/0d48b899-7441-4010-98ab-7d89e9bca28b
[18352:17196:0402/125114.380:ERROR:device_event_log_impl.cc(214)] [12:51:14.380] USB: usb_device_handle_win.cc:1056 Failed to read descriptor from node connection: A device attached to the system is not function
ing. (0x1F)
[18352:17196:0402/125114.382:ERROR:device_event_log_impl.cc(214)] [12:51:14.382] USB: usb_device_handle_win.cc:1056 Failed to read descriptor from node connection: A device attached to the system is not function
ing. (0x1F)
[18352:17196:0402/125114.382:ERROR:device_event_log_impl.cc(214)] [12:51:14.382] USB: usb_device_handle_win.cc:1056 Failed to read descriptor from node connection: A device attached to the system is not function
ing. (0x1F)
.
----------------------------------------------------------------------
Ran 1 test in 15.205s

OK
Destroying test database for alias 'default'...
```

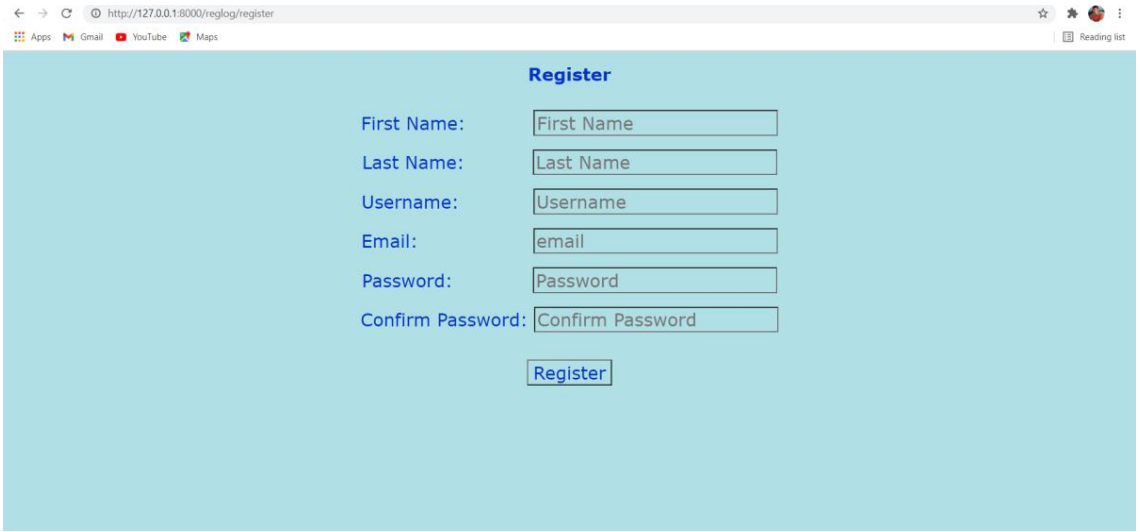


# 6.Work flow/Layout

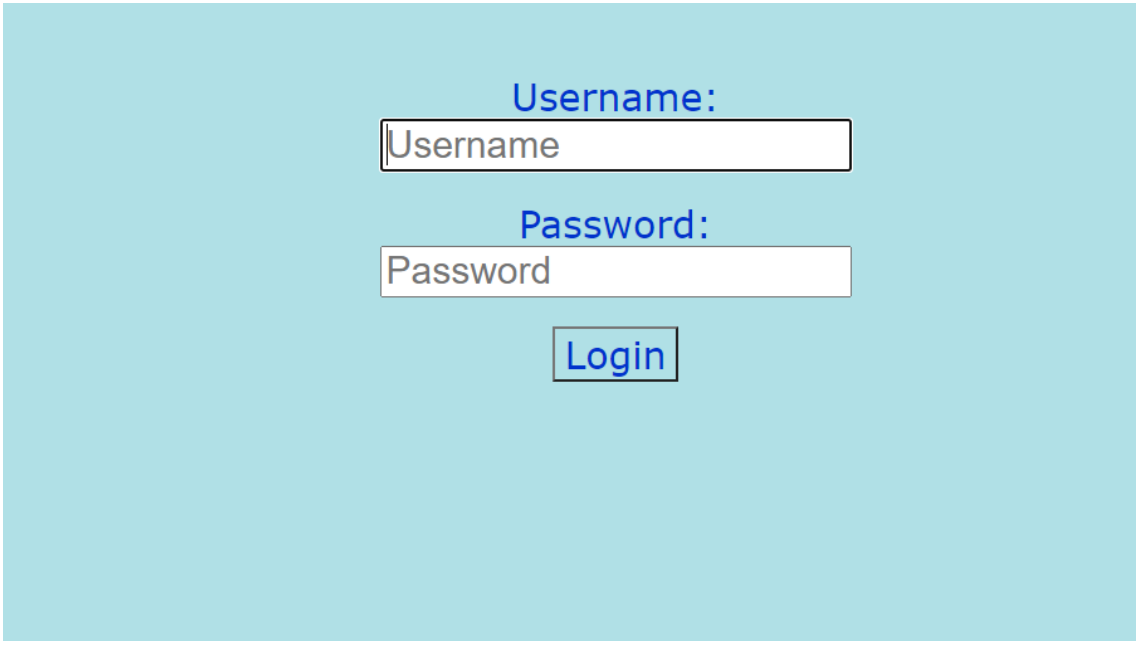
First home page is displayed there is option like register and login.



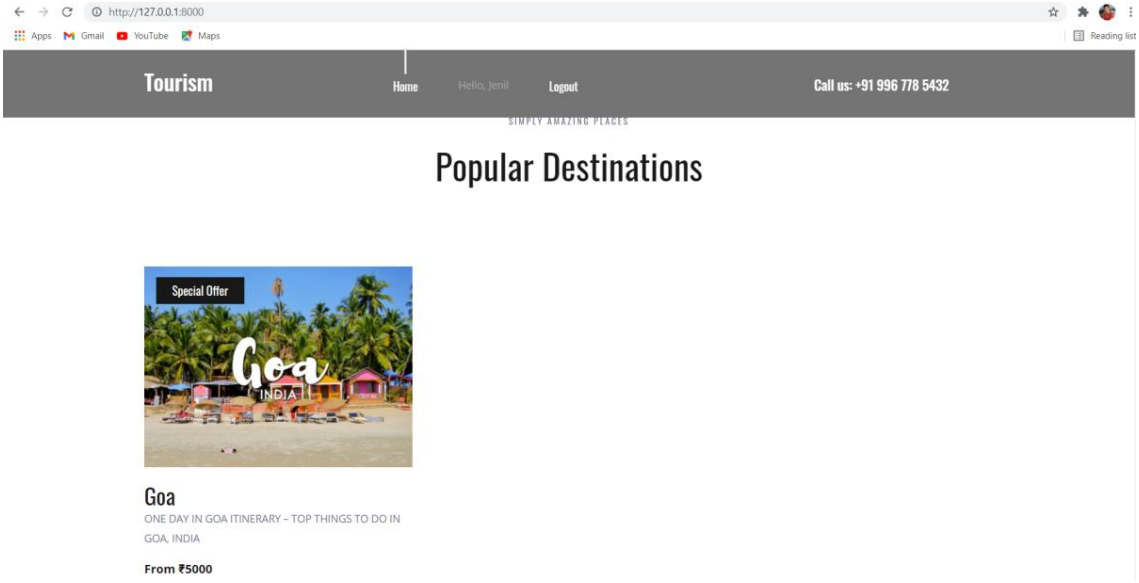

Register Module is for new user. For register we have to provide all details.



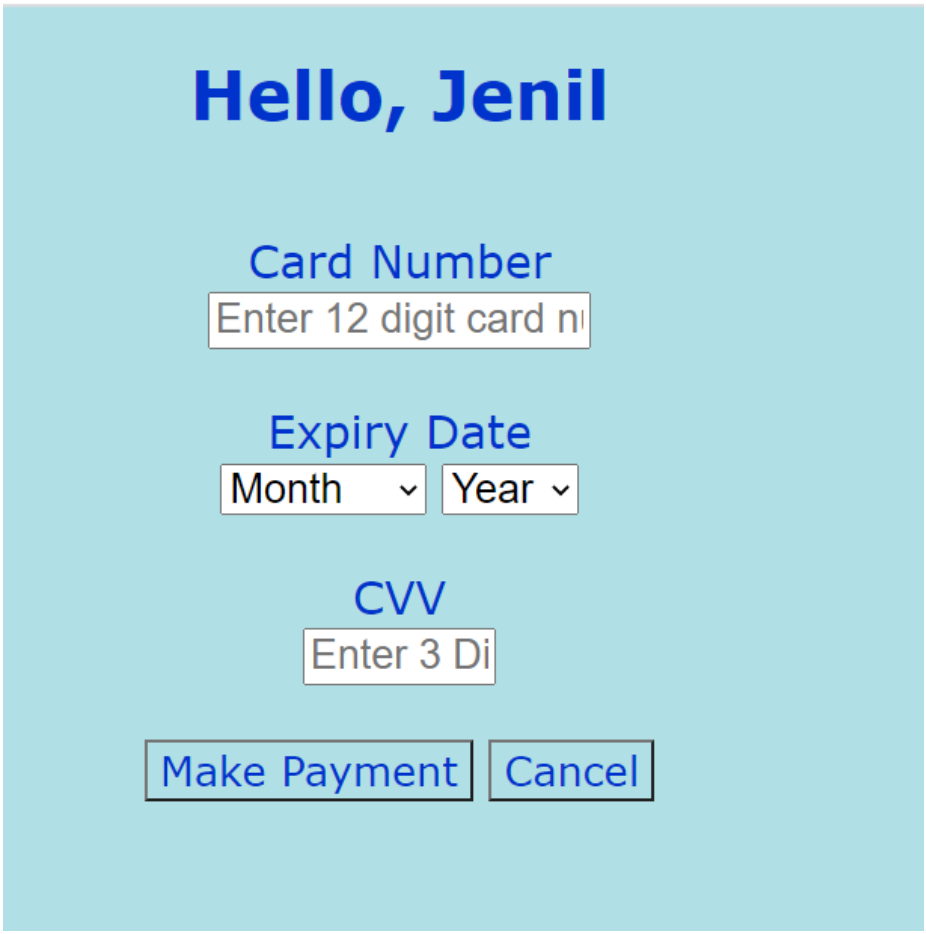

If we are not new user then we can directly login.



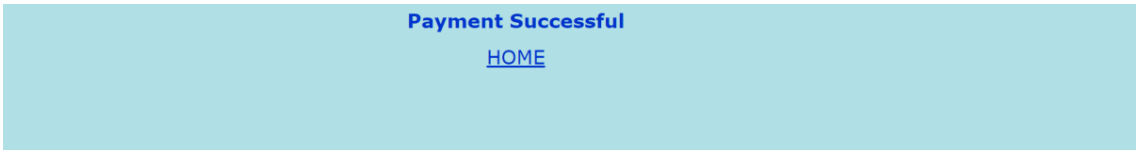

This is place and price added by admin.




We have to click on goa and we redirect to payment page.




After entering all data and click on Make Payment button.
We will redirect to payment successful page.
And by clicking on HOME we redirect to home page.



# 7.Conclusion

By using this application we can easily book ticket at home. We Don’t have to go outside to book ticket.

By making this project we conclude that how actual project works and how data behind different page are flows using Django framework. Also how css can be used for proper indentation of pages.

# 8.Limitations and Future Extensions

- **Limitations**

  User can not select seat of bus.

  User cannot change his data.

  If user forget his password he cannot retrive it.

- **Future Extensions**

  Update user details future can be added.

  Select seat in bus and see different buses of particular root can be added.

  Forgot password functionality can be added.

# 9.Bibliography

This project is fully self maded.

Some css code for proper indentation of pages are taken from internet. Theme is taken from internet.

(www.google.com)
(https://colorlib.com/)