

Python-Django Internship Report

Personal details

Name : Riya Sarkar

College Name : Shree Swaminarayan
Institute of Technology

Semester : 7th

Degree : B.E

GitHub URL :

https://github.com/RiyaSarkar-147/Python_15days_internship

Company Details

Company Name : Akash Techno labs

External Guide : Akash Techno labs

Training Duration : 26 May to 10 June 2021

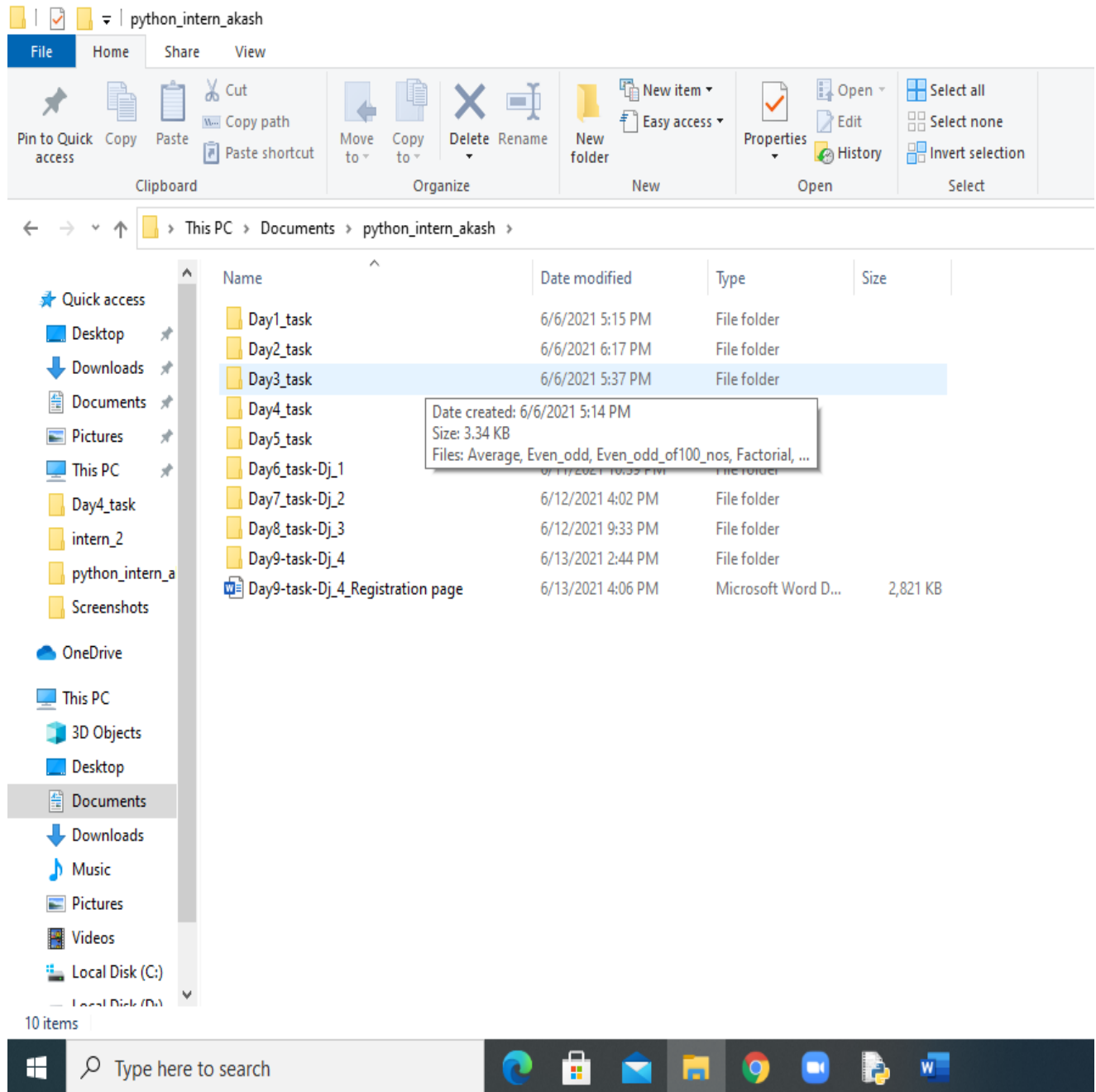
INDEX

Sr no	Day	Task	Pg. No
1	DAY 1	HTML PAGE	8
2	DAY 2	SUM OF TWO NUMBERS	9
		DATA AND ITS TYPES	9
3	DAY 3	THE GREATEST OF TWO NOS	10

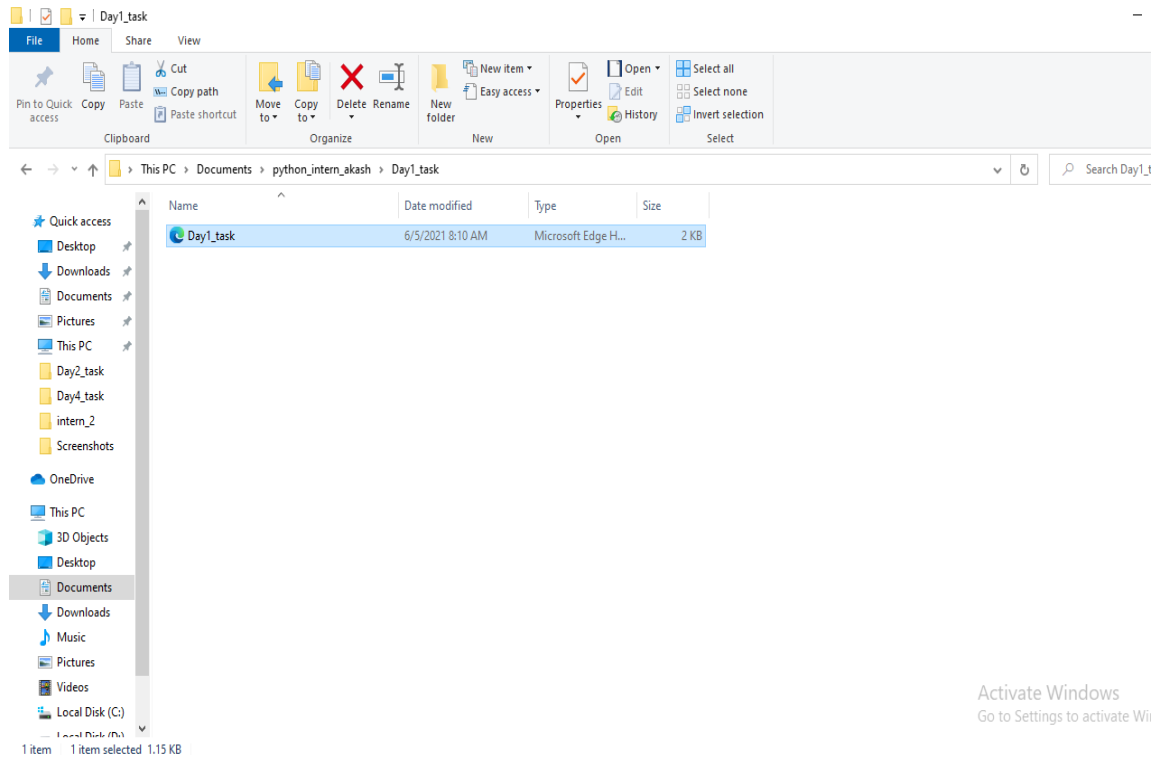
		FACTORIAL, AVG , SWAP OF TWO NOS	11
4	DAY 4	USE OF IF-ELIF AND FIBONACCI	13
		USE OF PASS AND NESTED FOR LOOP	13
5	DAY 5	AREA OF RECTANGLE AND SIMPLE INTEREST	15
		SCHEME CUSTOMER ANS INHERITANCE	15
6	DAY 6	DJANGO – INTRODUCTION AND INSTALLATION WORK	16-19
7	DAY 7	CREATION OF AN APP NAMED MYAPP	16-19
8	DAY 8	CREATION OF FORM	20-22

		ADDITION OF SUM	22-26
9	DAY 9	PRACTICE BY SELF – CREATION OF SIGN IN PAGE	26-31

THE IMAGE OF THE DOCUMENT FILE



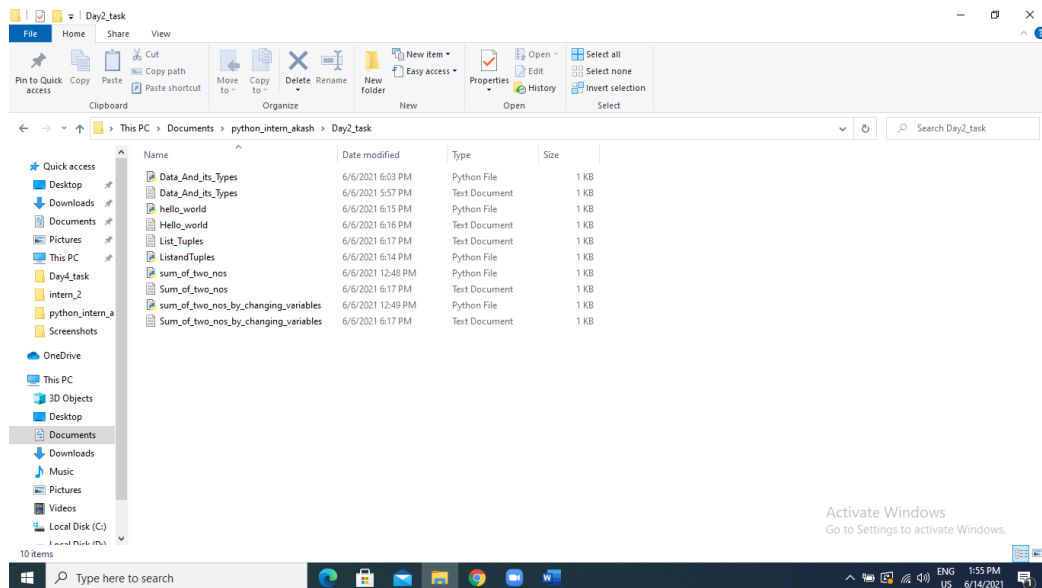
DAY 1 :



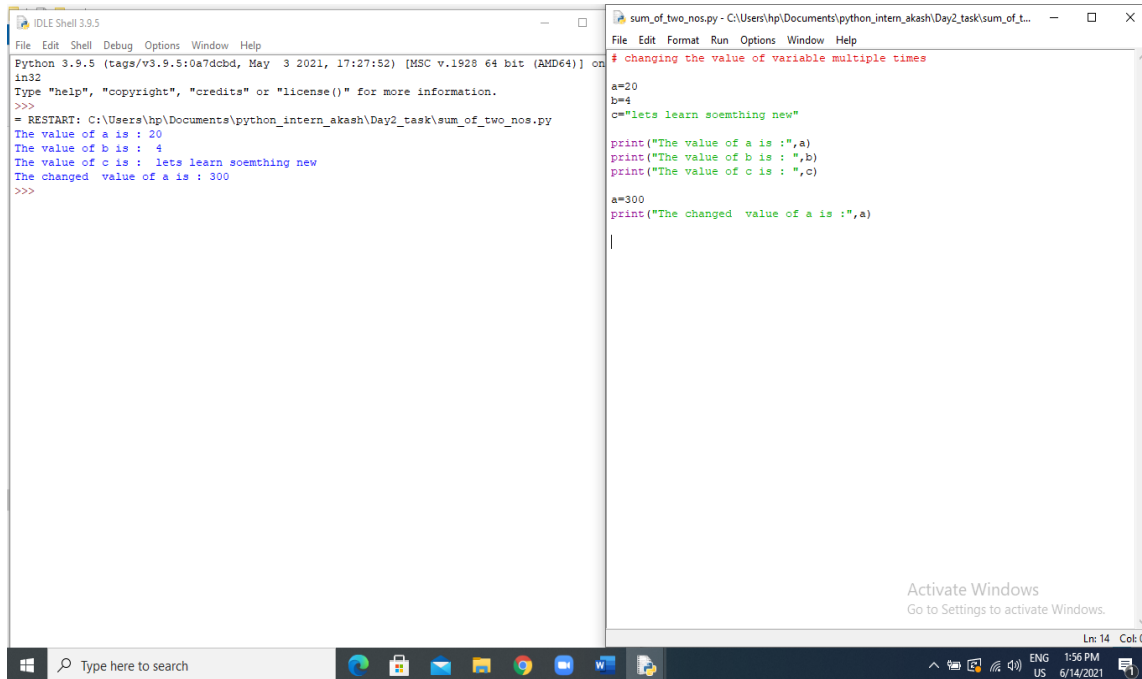
>> HTML PAGE

```
Day1task - Notepad
File Edit Format View Help
<html>
<form>
  <div class="container">
    <h1>
      Register Here
    </h1>
    <p>
      Please fill in the details to create an account with us.
    </p>
    <hr>
    <label for="email">
      <b>Enter Email : </b>
    </label>
    <input type="text" placeholder="Enter Email :" name="email">
    <p>
      <label for="pwd"><b>Password : </b></label>
      <input type="password" placeholder="Enter your Password :" name="pwd">
    </p>
    <p>
      <label for="confirm"><b>Confirm Password : </b></label>
      <input type="password" placeholder="Confirm Password :" name="confirm">
    </p>
    <hr>
    <p>>By creating an account you agree to our <a href="#"><p>Terms & Privacy</p></a></p>
    <button type="submit" class="registerbtn"><strong>Register</strong></button>
  </div>
  <div class="container signin">
    <p>>Already have an account? <a href="#"><p>Sign in</p></a></p></div>
</form>
</html>
```

DAY 2 :



>> SUM OF TWO NOS

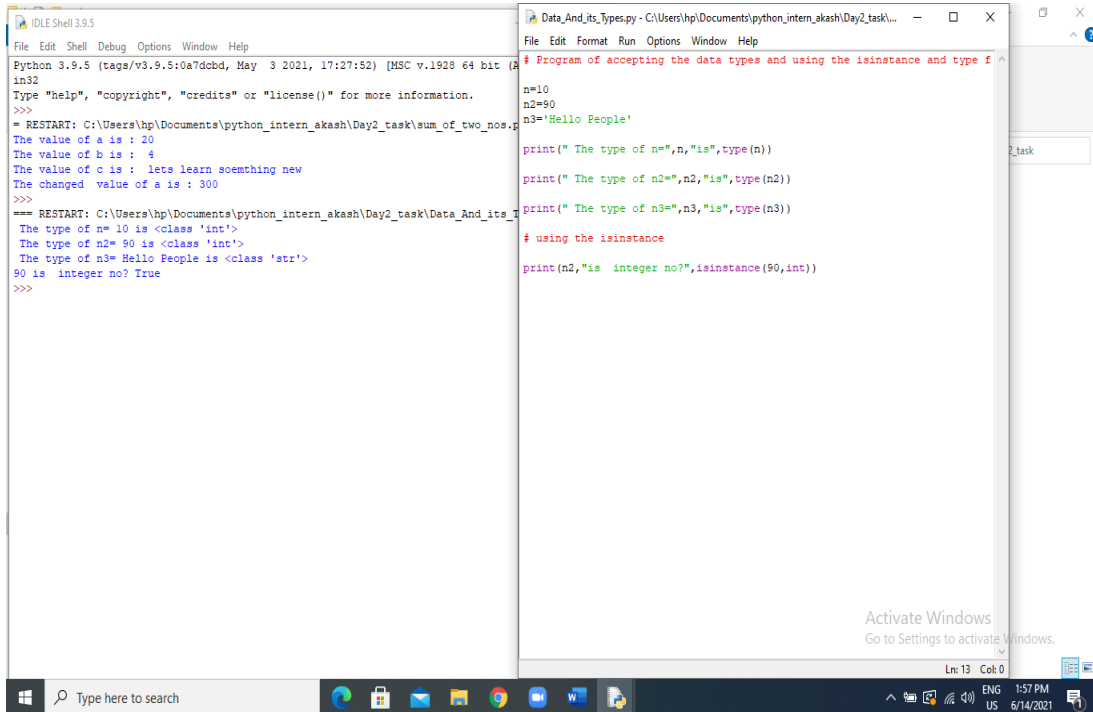


The screenshot shows two windows from the IDLE 3.9.5 environment. The left window is the IDLE Shell, displaying the execution of a Python script. The right window is the IDLE Editor, showing the source code of the script.

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\hp\Documents\python_intern_akash\Day2_task\sum_of_two_nos.py
The value of a is : 20
The value of b is : 4
The value of c is : lets learn soemthing new
The changed value of a is : 300
>>>
```

```
sum_of_two_nos.py - C:\Users\hp\Documents\python_intern_akash\Day2_task\sum_of_t...
File Edit Format Run Options Window Help
# changing the value of variable multiple times
a=20
b=4
c="lets learn soemthing new"
print("The value of a is :",a)
print("The value of b is : ",b)
print("The value of c is : ",c)
a=300
print("The changed value of a is :",a)
|
```

>> DATA AND ITS TYPES

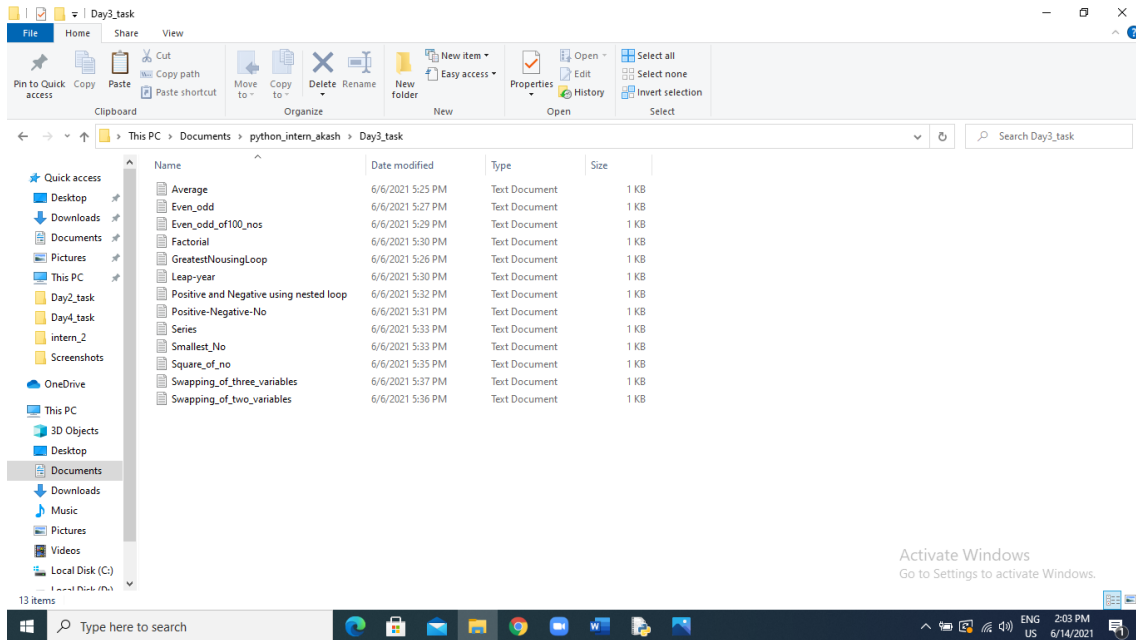


The screenshot shows two windows from the IDLE 3.9.5 environment. The left window is the IDLE Shell, displaying the execution of a Python script. The right window is the IDLE Editor, showing the source code of the script.

```
Python 3.9.5 (tags/v3.9.5:0a7dcdb, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\hp\Documents\python_intern_akash\Day2_task\sum_of_two_nos.py
The value of a is : 20
The value of b is : 4
The value of c is : lets learn soemthing new
The changed value of a is : 300
>>>
=== RESTART: C:\Users\hp\Documents\python_intern_akash\Day2_task\Data_And_its...
The type of n= 10 is <class 'int'>
The type of n2= 90 is <class 'int'>
The type of n3= Hello People is <class 'str'>
90 is integer no? True
>>>
```

```
Data_And_its_Types.py - C:\Users\hp\Documents\python_intern_akash\Day2_task\...
File Edit Format Run Options Window Help
# Program of accepting the data types and using the isinstance and type f...
n=10
n2=90
n3='Hello People'
print(" The type of n=",n,"is",type(n))
print(" The type of n2=",n2,"is",type(n2))
print(" The type of n3=",n3,"is",type(n3))
# using the isinstance
print(n2,"is integer no?",isinstance(90,int))
```

Day 3 :



>> GREATEST NO USING NESTED LOOP

The image shows two Notepad windows side-by-side. The left window, titled 'Positive and Negative using nested loop - Notepad', contains the following Python code:

```
# To calculate the Positive or negative using nested loop

a = int(input("Enter first integer: "))
b = int(input("Enter second integer: "))
c = int(input("Enter third integer: "))
if(a > b):
    if(a > c):
        print(a, " is greatest.")
    else:
        print(b, " is greatest.")
else:
    if(b > c):
        print(b, " is greatest.")
    else:
        print(c, " is greatest.")
```

The right window, titled 'GreatestNousingLoop - Notepad', contains the following Python code:

```
# Program to write the greatest no using the nested loop

n = int(input("Enter the number: "))
if(n != 0):
    if(n > 0):
        print(n, " is Positive")
    else:
        print(n, " is NEgative")
else:
    print("Number you entered is 0")
```

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date 6/14/2021 and time 2:06 PM.

>> FACTORIAL, TO SWAP TWO NOS, AVG

The image shows three Notepad windows side-by-side. The left window, titled 'Swapping_of_three_variables - Notepad', contains the following Python code:

```
# To swap three variables

a = int(input("Enter first integer: "))
b = int(input("Enter second integer: "))
print("Before swapping")
print("The value of first integer is: ", a)
print("The value of second integer is: ", b)
c = a
a = b
b = c
print("After swapping")
print("The value of first integer is: ", a)
print("The value of second integer is: ", b)
```

The middle window, titled 'Average - Notepad', contains the following Python code:

```
# program to calculate the average

sum = 0
for i in range(5):
    n = int(input("Enter an integer: "))
    sum = sum+n
    avg = sum/5
print("Average is: ", avg)
```

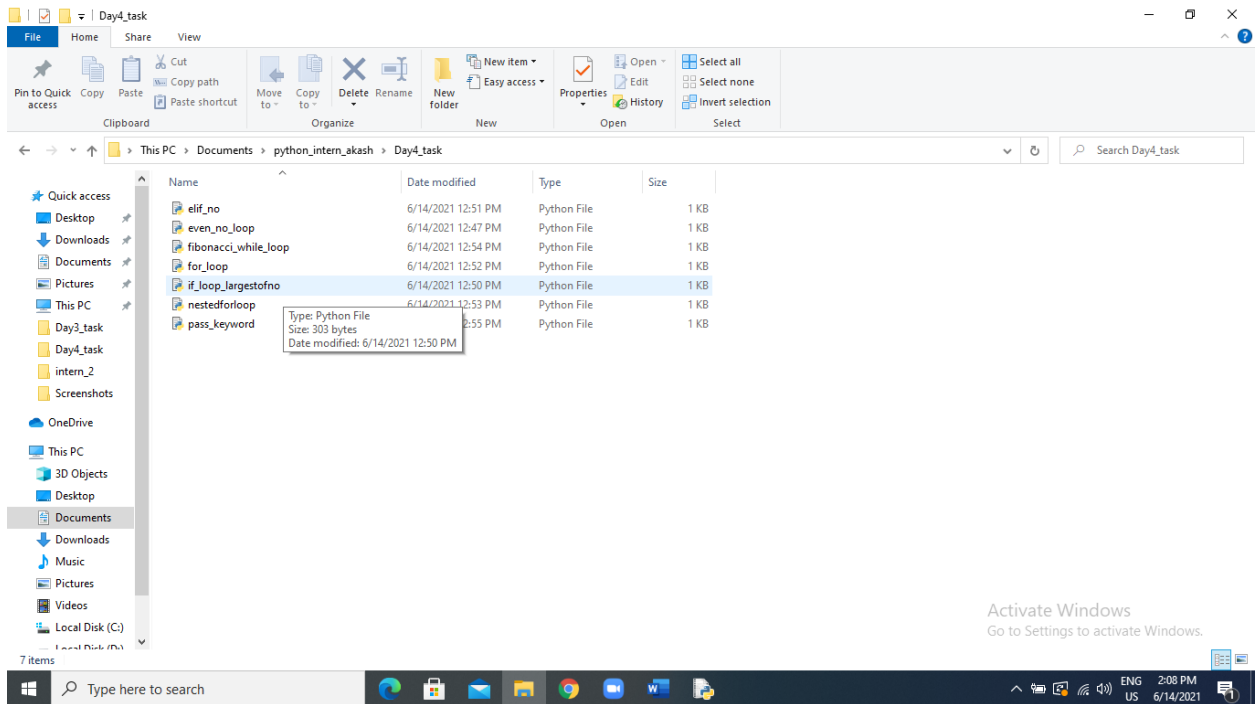
The right window, titled 'Factorial - Notepad', contains the following Python code:

```
# To calculate the factorial

n = int(input("Enter a positive integer number: "))
f = 1
if n == 0:
    print("Factorial of ", n, " is: "+str(f))
else:
    for i in range(1, n+1):
        f = f*i
    print("Factorial of ", n, " is: "+str(f))
```

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date 6/14/2021 and time 2:07 PM.

Day 4 :



>> FIBONACCI AND IF-ELIF

```
# fibonacci_while_loop.py
# Fibonacci series using while loop
terms = int(input("Enter the terms "))
# first two initial terms
a = 0
b = 1
count = 0

# check if the number of terms is Zero or negative
if (terms <= 0):
    print("Please enter a valid integer")
elif (terms == 1):
    print("Fibonacci sequence upto",limit,":")
    print(a)
else:
    print("Fibonacci sequence:")
    while (count < terms):
        print(a, end = ' ')
        c = a + b
        # updateing values
        a = b
        b = c
        count += 1

# if_loop_largestofno.py
# Use of elif
number = int(input("Enter the number?"))
if number==10:
    print("number is equals to 10")
elif number==50:
    print("number is equal to 50");
elif number==100:
    print("number is equal to 100");
else:
    print("number is not equal to 10, 50 or 100");

IDLE Shell 3.9.5
Python 3.9.5 (tags/v3.9.5:0a7dcb0, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\hp\Documents\python_intern_akash\Day4_task\if_loop_largestofno.py
Enter the number?
Traceback (most recent call last):
  File "C:\Users\hp\Documents\python_intern_akash\Day4_task\if_loop_largestofno.py", line 4, in <module>
    number = int(input("Enter the number?"))
ValueError: invalid literal for int() with base 10: ''
>>>
= RESTART: C:\Users\hp\Documents\python_intern_akash\Day4_task\if_loop_largestofno.py
Enter the number?6
number is not equal to 10, 50 or 100
>>>
```

>>NESTED FOR-LOOP AND USE OF PASS

```
# nestedforloop.py
# Use of nested for loop

# User input for number of rows
rows = int(input("Enter the rows:"))

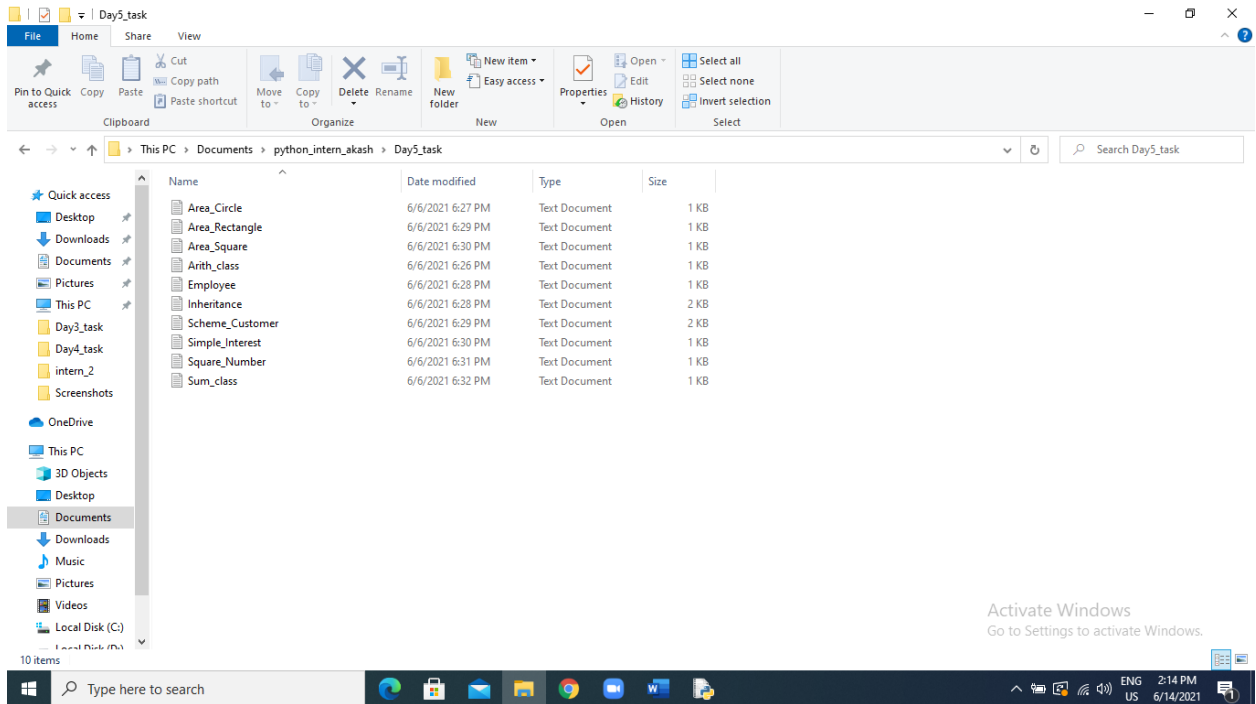
# Outer loop will print number of rows
for i in range(0,rows+1):

# Inner loop will print number of Astrisk
    for j in range(i):
        print("",end = ' ')
        print()

# pass_keyword.py
# Use of pass
for i in [1,2,3,4,5]:
    if(i==4):
        pass
        print("This is pass block",i)
    print(i)

IDLE Shell 3.9.5
Python 3.9.5 (tags/v3.9.5:0a7dcb0, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in formation.
>>>
= RESTART: C:\Users\hp\Documents\python_intern_akash\Day4_task\pass_keyword.py =
1
2
3
This is pass block 4
4
5
>>> |
```

Day 5 :



>>SIMPLE INTEREST AND AREA OF REC-

The image shows two Notepad windows side-by-side. The left window, titled 'Area_Rectangle - Notepad', contains Python code for a class 'cal5' that calculates the area of a rectangle. It includes an __init__ method, a calArea method, and a display method. Below the class definition, there is a main block that takes user input for length and width, creates an object, and prints the area. The right window, titled 'Simple_Interest - Notepad', contains Python code for a class 'cal3' that calculates simple interest. It includes an __init__ method, a calInterest method, and a display method. Below the class definition, there is a main block that takes user input for principle amount, rate of interest, and number of months, creates an object, and prints the simple interest. Both windows have a status bar at the bottom showing 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'. The Windows taskbar is visible at the bottom of the image.

```
Area_Rectangle - Notepad
File Edit Format View Help
# Create a class cal5 that will calculate area of a rectangle.
# Create constructor method which has two parameters .
# Create calArea() method that will calculate area of a rectangle.
# Create display() method that will display area of a rectangle.
class cal5:

    def __init__(self, l, w):
        self.l = l
        self.w = w

    def calArea(self):
        return((self.l*self.w))

    def display(self):
        return(self.calArea())

l = float(input("Enter length: "))
w = float(input("Enter width: "))
obj = cal5(l, w)
print("Area of Rectangle is:", obj.display())

Simple_Interest - Notepad
File Edit Format View Help
# Create a class cal3 that will calculate simple interest.
# Create constructor method which has three parameters .
# Create calInterest() method that will calculate Interest .
# Create display() method that will display Interest.
class cal3:

    def __init__(self, p, r, n):
        self.p = p
        self.r = r
        self.n = n

    def calInterest(self):
        return((self.p*self.r*self.n)/100)

    def display(self):
        return(self.calInterest())

p = float(input("Enter Principle Amount: "))
r = float(input("Enter rate of interst: "))
n = int(input("Enter the number of months: "))
obj = cal3(p, r, n)
print("Simple Interest is:", obj.display())
```

>> SCHEME CUSTOMER ANS INHERITANCE

The image shows two Notepad windows side-by-side. The left window, titled 'Inheritance - Notepad', contains Python code demonstrating inheritance. It defines a base class 'publisher' with an __init__ method and a gettitle method. Then, it defines two subclasses: 'book' which inherits from 'publisher' and adds a pno attribute and a getpno method, and 'tape' which inherits from 'book' and adds a time attribute and a gettime method. At the bottom, there is a main block that takes user input for title, pno, and time, creates a 'tape' object, and prints its attributes. The right window, titled 'Scheme_Customer - Notepad', contains Python code for a class 'customer' that inherits from a 'scheme' class. The 'scheme' class has an __init__ method and attributes for scheme_id, scheme_name, outgoing_rate, and message_charge. The 'customer' class has an __init__ method that takes all these attributes plus cust_id, name, and mobile_no. Below the class definitions, there is a main block that takes user input for all these values, creates a 'customer' object, and prints its attributes. Both windows have a status bar at the bottom showing 'Ln 29, Col 43' and 'Ln 1, Col 1' respectively, '100%', 'Windows (CRLF)', and 'UTF-8'. The Windows taskbar is visible at the bottom of the image.

```
Inheritance - Notepad
File Edit Format View Help
# Write a program with use of inheritance: Define a class publisher ti
class publisher():
    def __init__(self, title):
        self.title = title

    def gettitle(self):
        return self.title

class book(publisher):
    def __init__(self, title, pno):
        publisher.__init__(self, title)
        self.pno = pno

    def getpno(self):
        return self.pno

class tape(book):
    def __init__(self, title, pno, time):
        book.__init__(self, title, pno)
        self.time = time

    def gettime(self):
        return self.time

title = input("Enter the name of title: ")
pno = int(input("Enter Page Number: "))
time = input("Enter Time: ")
obj = tape(title, pno, time)
print("Title Name: ", obj.gettitle())
print("Page Number: ", obj.getpno())
print("Time: ", obj.gettime())

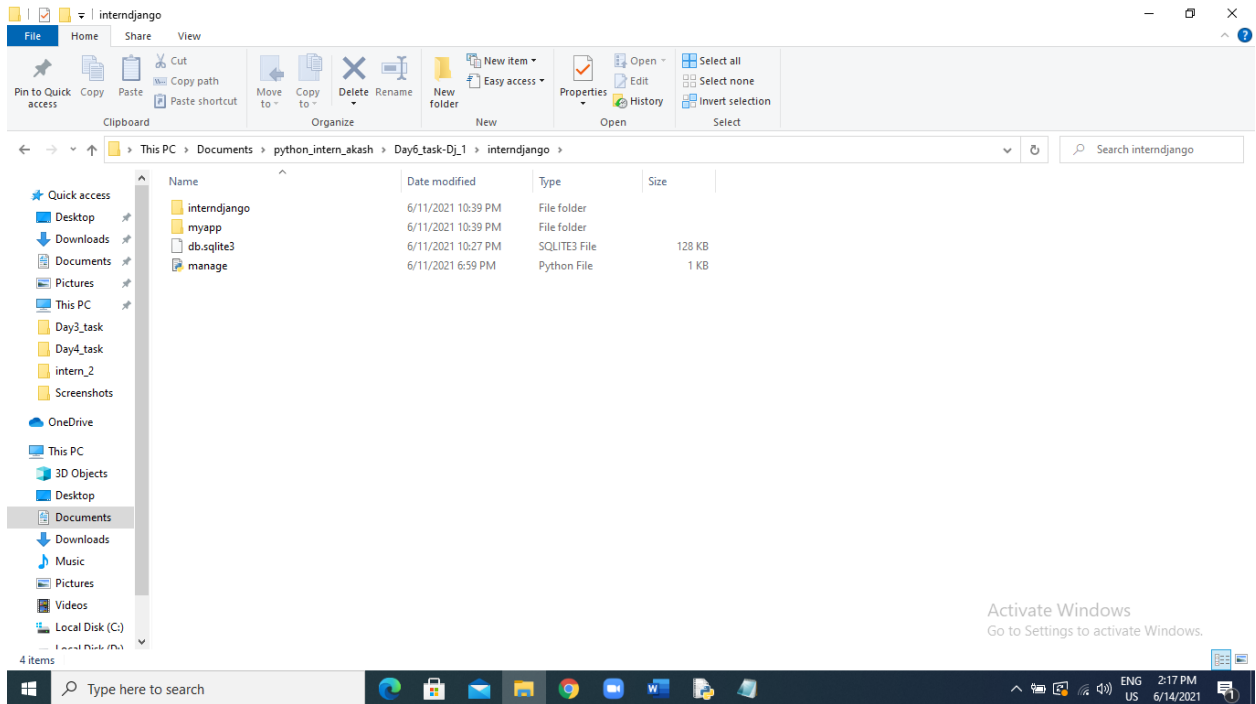
Scheme_Customer - Notepad
File Edit Format View Help
# Create a class called scheme with scheme_id, scheme_name,outgoing_rate, and message_charge.
# Derive customer class form scheme and include cust_id, name and mobile_no data.
# Define necessary functions to read and display data

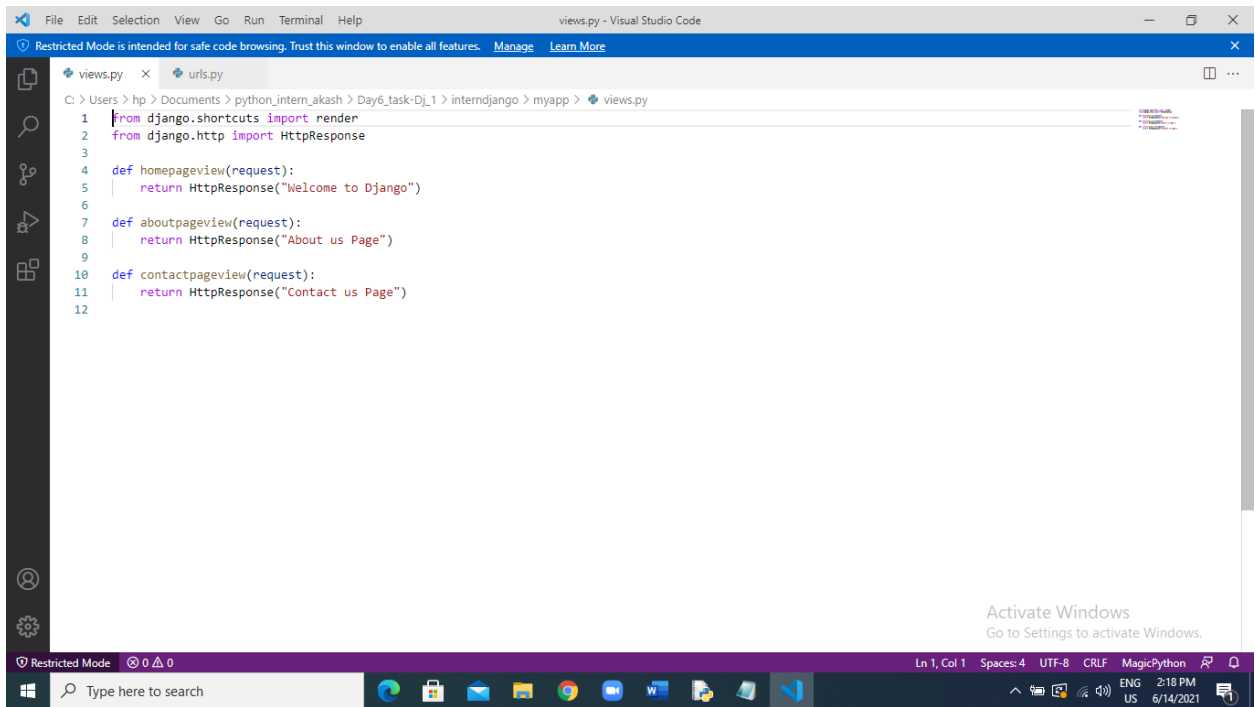
class scheme():
    def __init__(self, scheme_id, scheme_name, outgoing_rate, message_charge):
        self.scheme_id = scheme_id
        self.scheme_name = scheme_name
        self.outgoing_rate = outgoing_rate
        self.message_charge = message_charge

class customer(scheme):
    def __init__(self, scheme_id, scheme_name, outgoing_rate, message_charge, cust_id, name, m
        scheme.__init__(self, scheme_id, scheme_name, outgoing_rate, message_charge)
        self.cust_id = cust_id
        self.name = name
        self.mobile_no = mobile_no
        print(self.scheme_id, self.scheme_name, self.outgoing_rate,
              self.message_charge, self.cust_id, self.name, self.mobile_no)

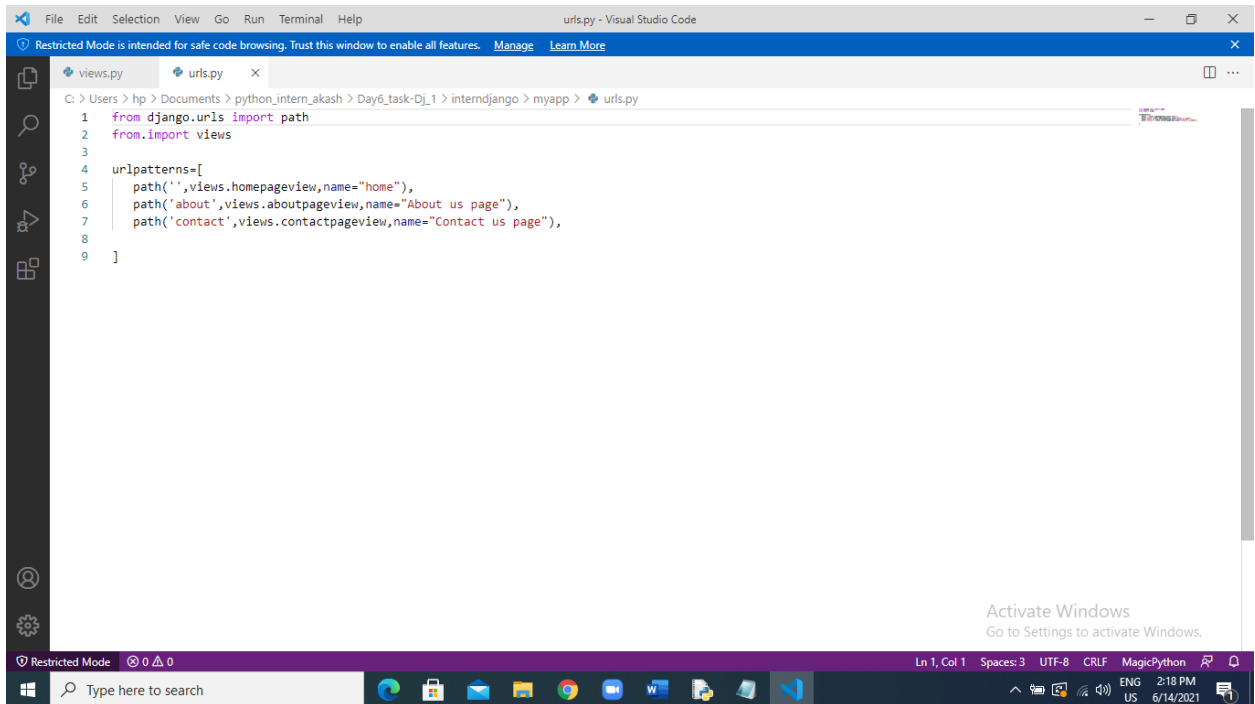
scheme_id = int(input("Enter Scheme id: "))
scheme_name = input("Enter Scheme Name: ")
outgoing_rate = float(input("Enter Outgoing Rate: "))
message_charge = float(input("Enter Message Charge: "))
cust_id = int(input("Enter Customer id: "))
name = input("Enter Customer Name: ")
mobile_no = input("Enter mobile no: ")
obj = customer(scheme_id, scheme_name, outgoing_rate, message_charge, cust_id, name, mobile_no)
```

Day 6 : Django works --- Introduction and Installation done

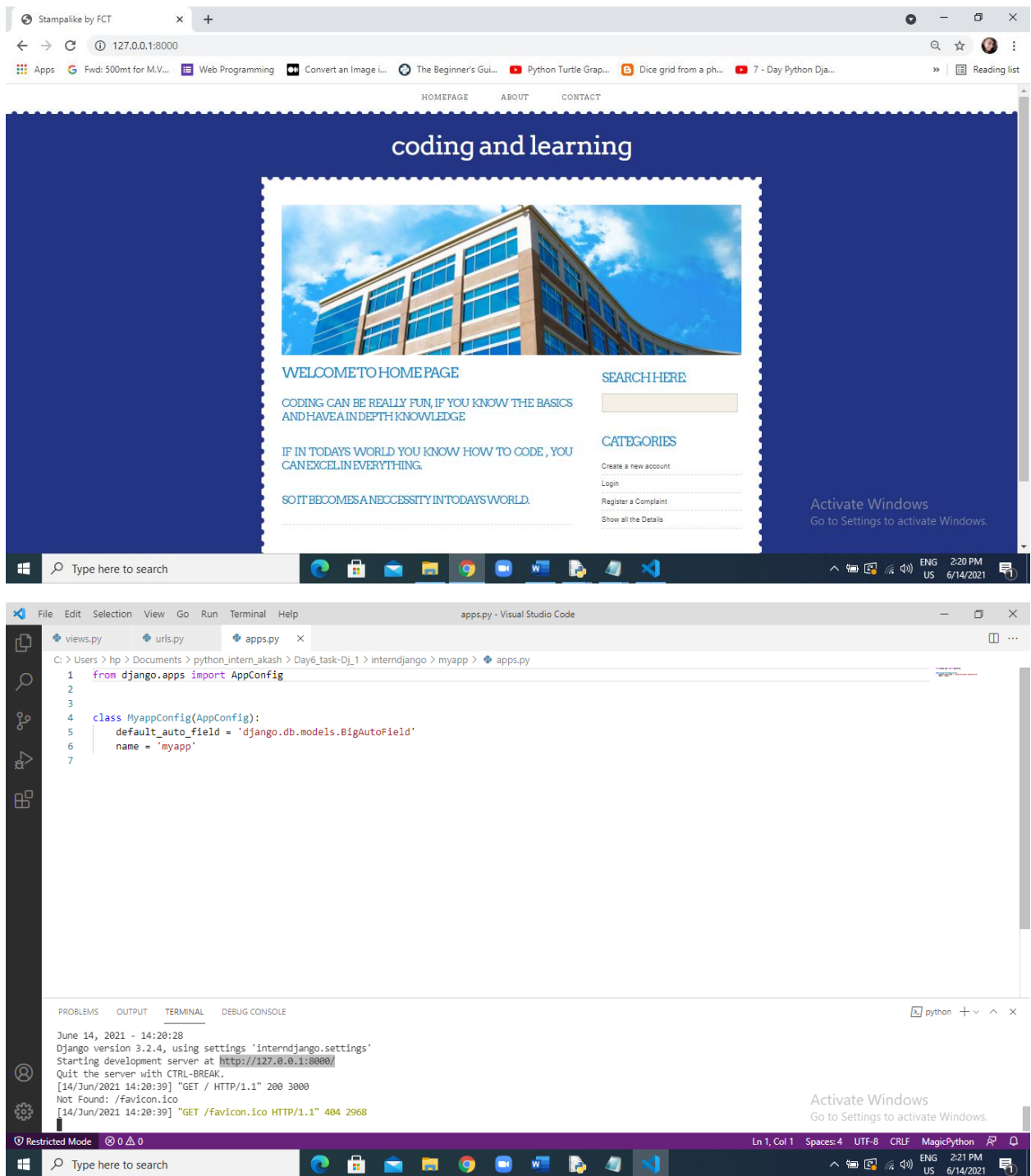




```
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3
4 def homepageview(request):
5     return HttpResponseRedirect("Welcome to Django")
6
7 def aboutpageview(request):
8     return HttpResponseRedirect("About us Page")
9
10 def contactpageview(request):
11     return HttpResponseRedirect("Contact us Page")
12
```



```
1 from django.urls import path
2 from . import views
3
4 urlpatterns=[
5     path('',views.homepageview,name="home"),
6     path('about',views.aboutpageview,name="About us page"),
7     path('contact',views.contactpageview,name="Contact us page"),
8 ]
9
```

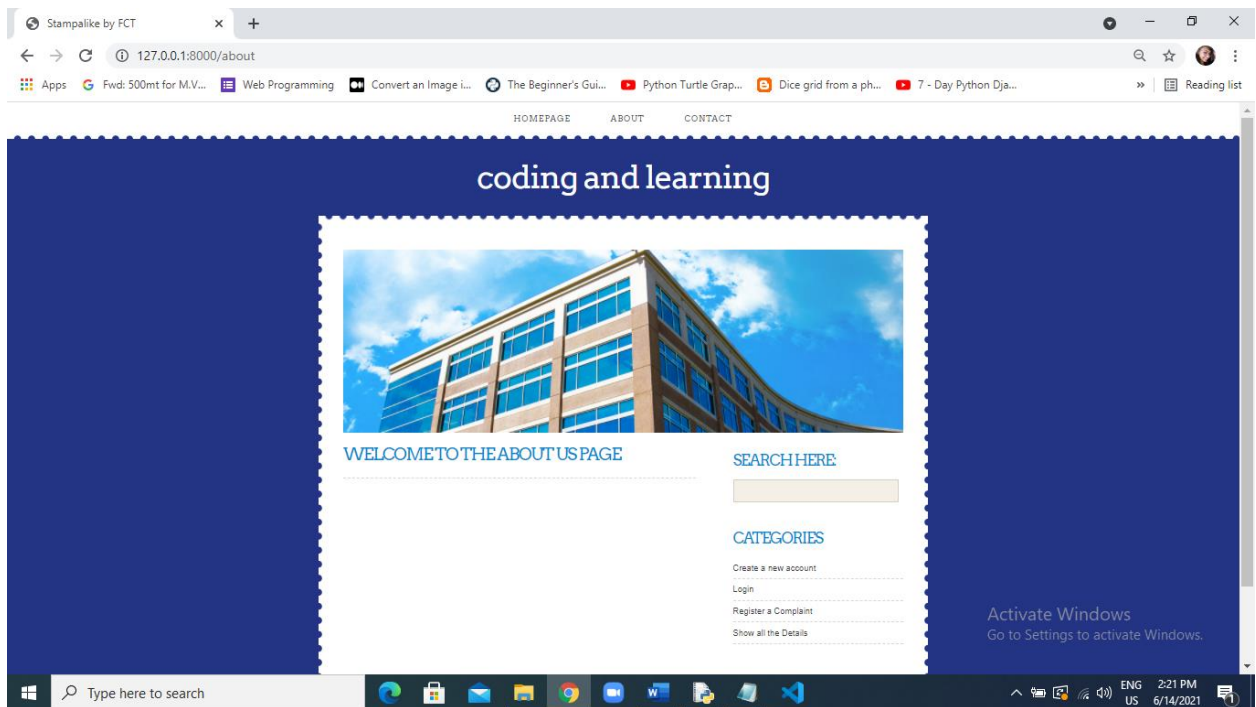


18

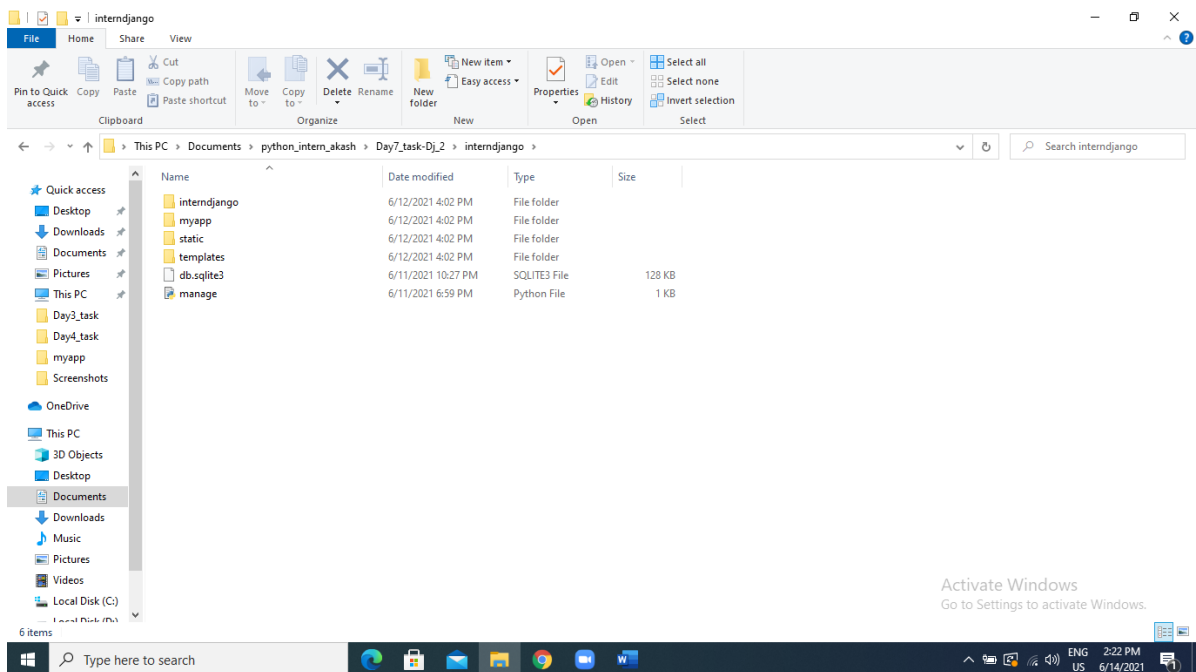
SUBMITTED BY -- RIYA SARKAR

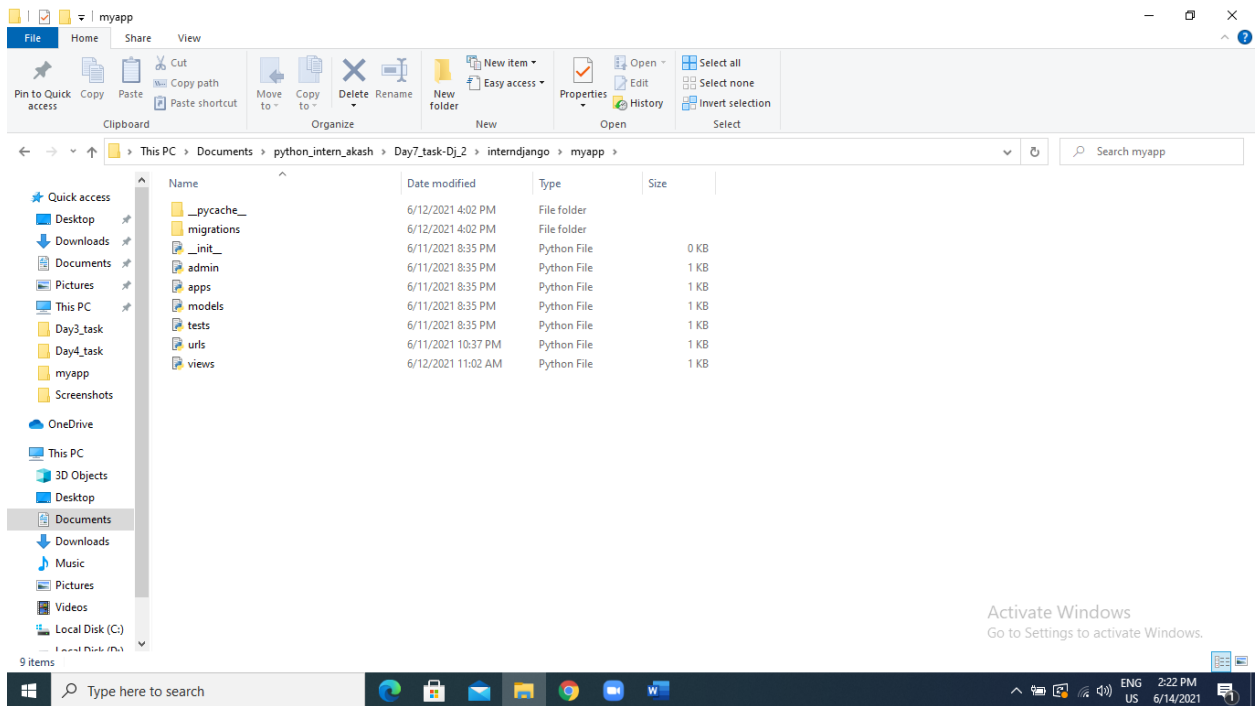
EN NO -- 181250107049

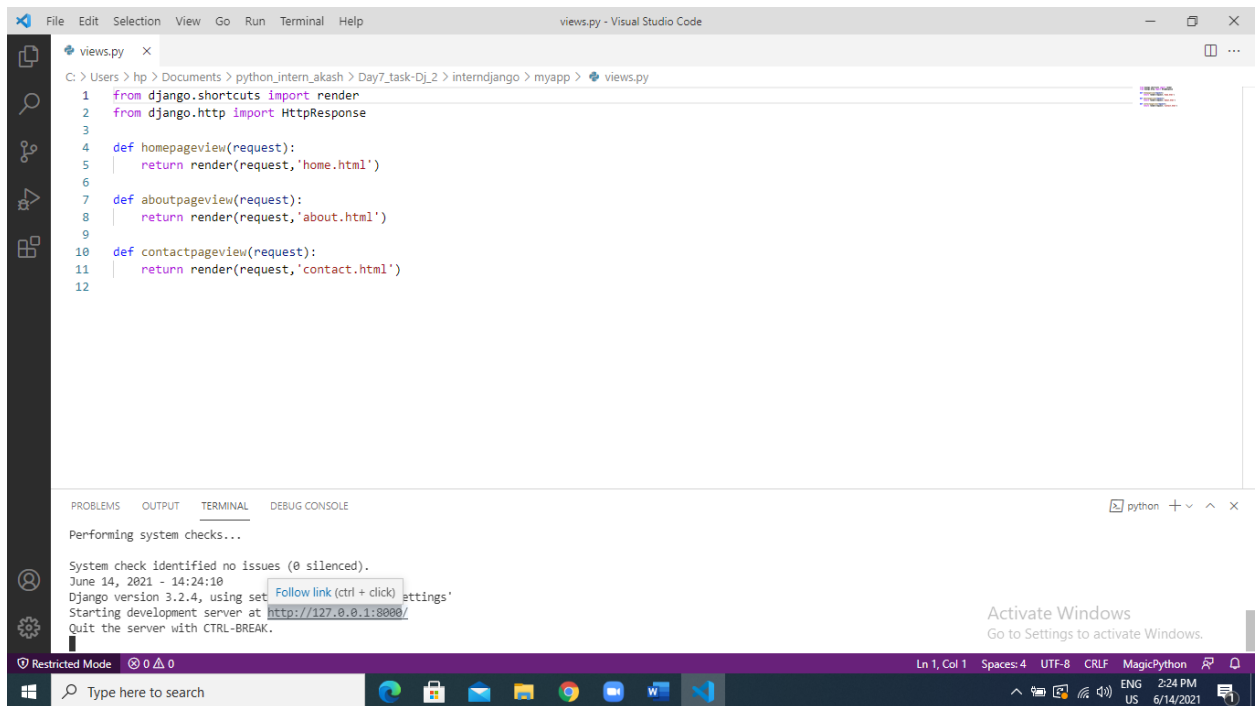
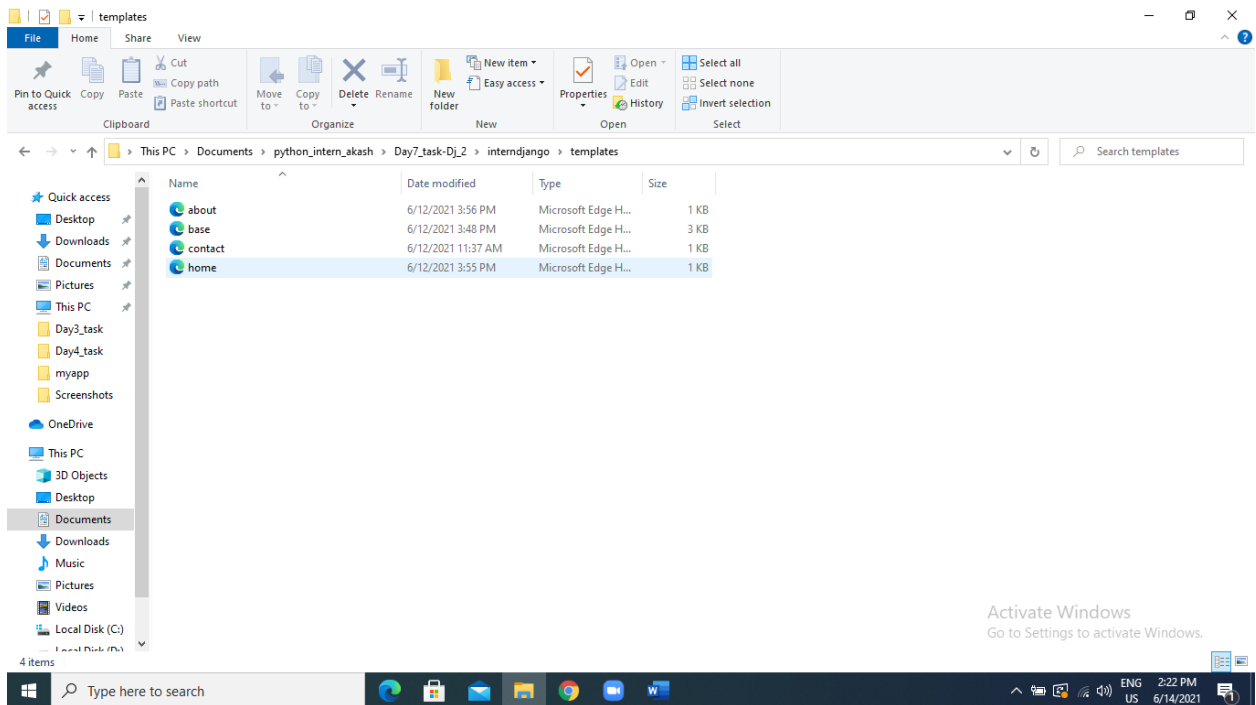
GIT HUB URL -- https://github.com/RiyaSarkar-147/Python_15days_internship

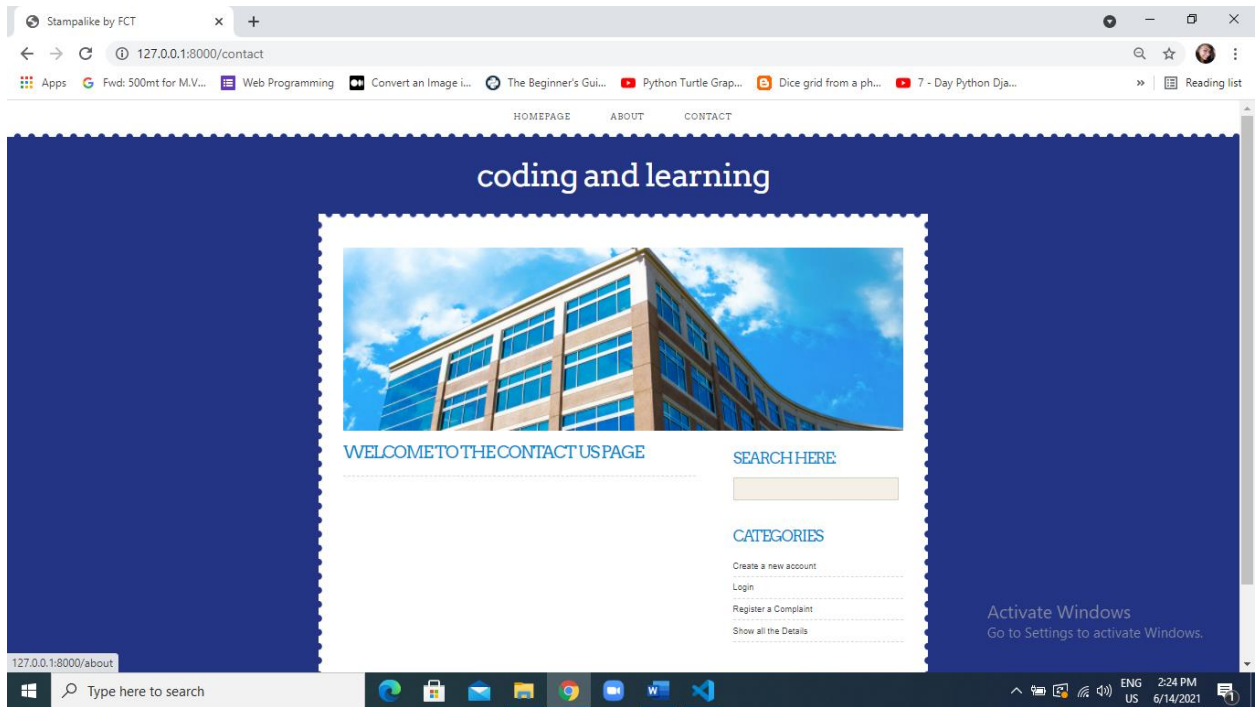


Day7 : CREATION OF AN APP NAMED MYAPP

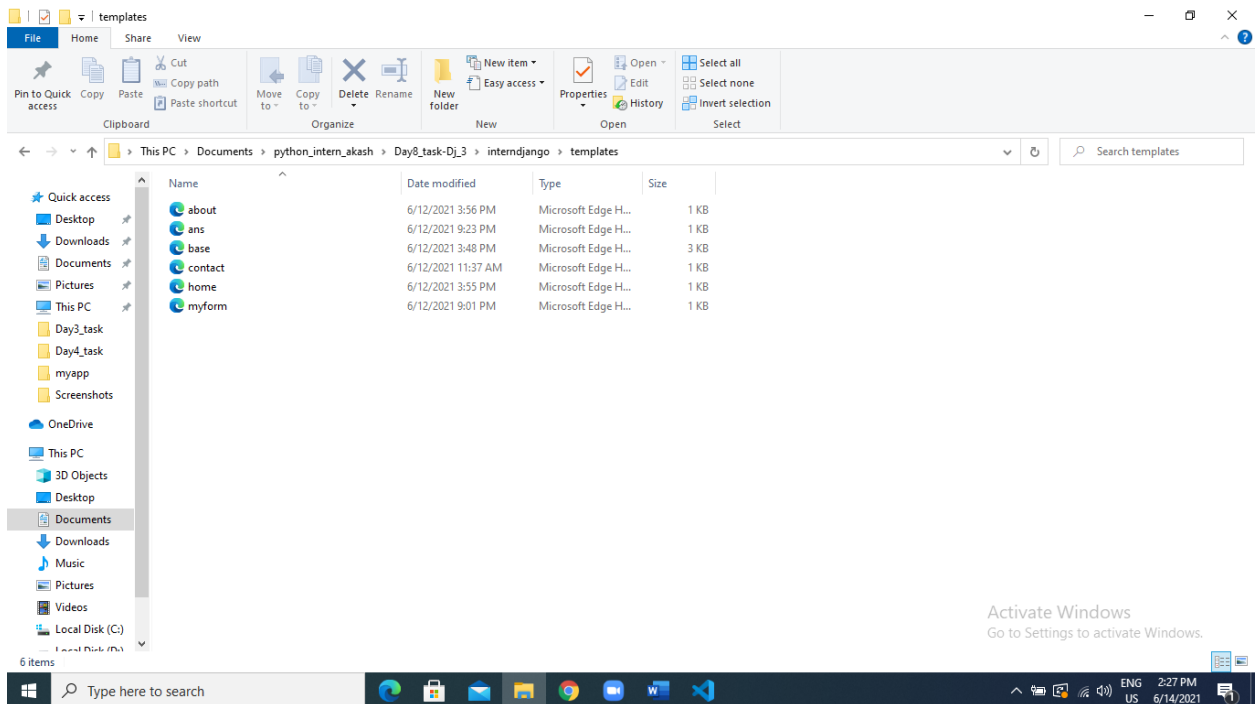


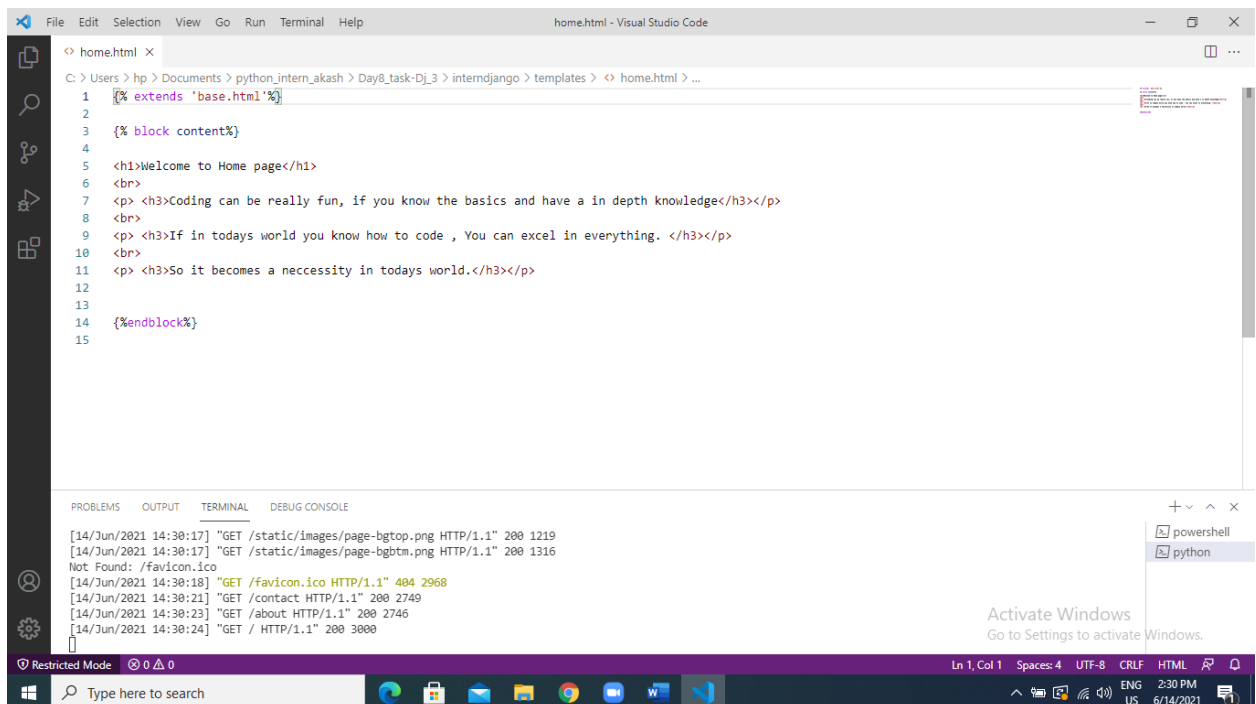
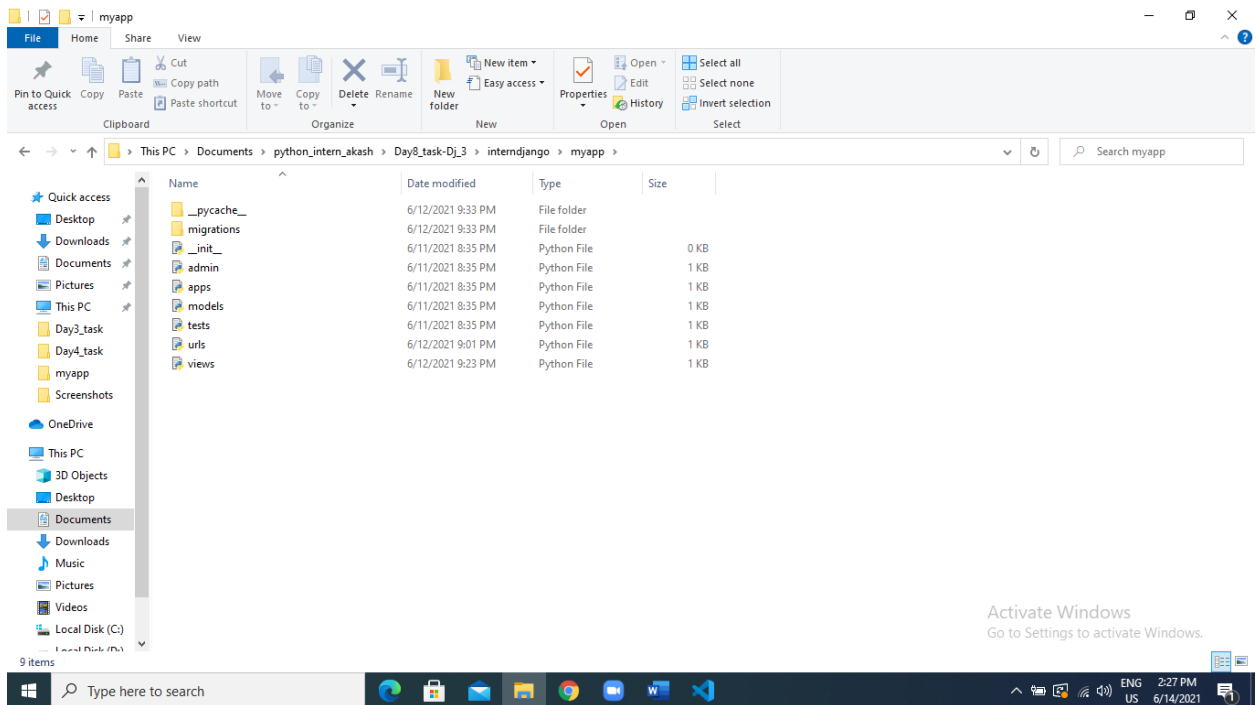


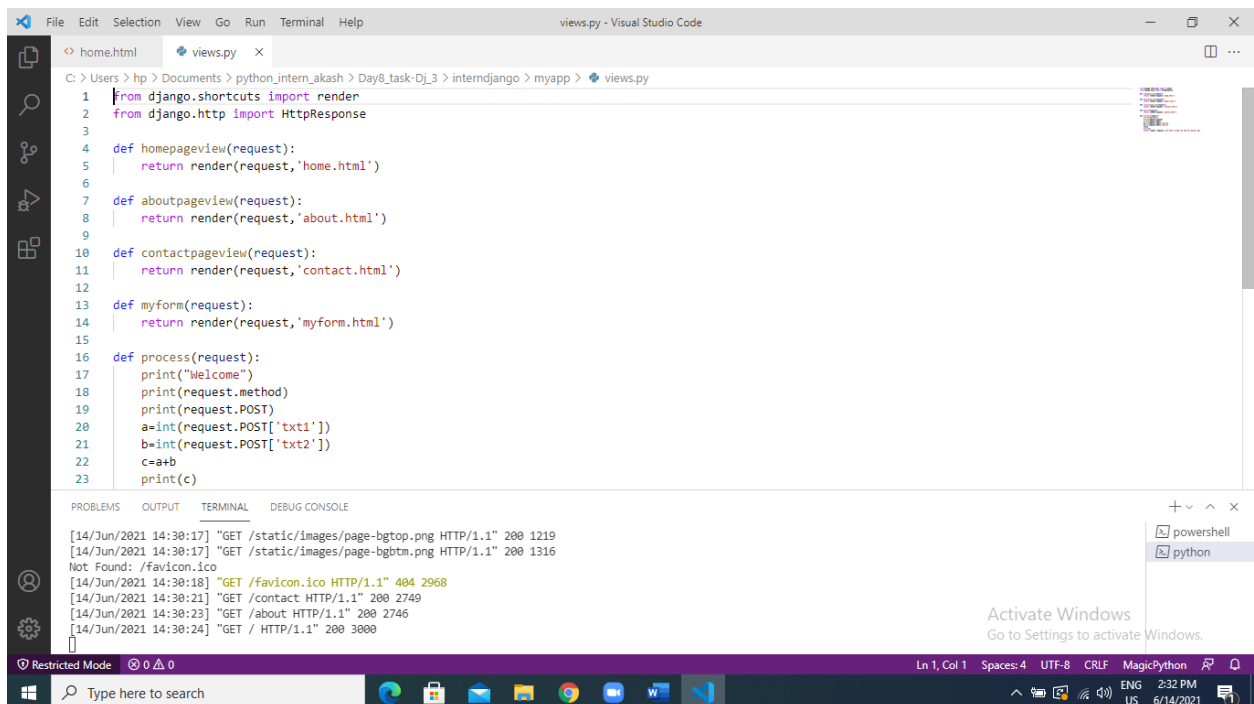
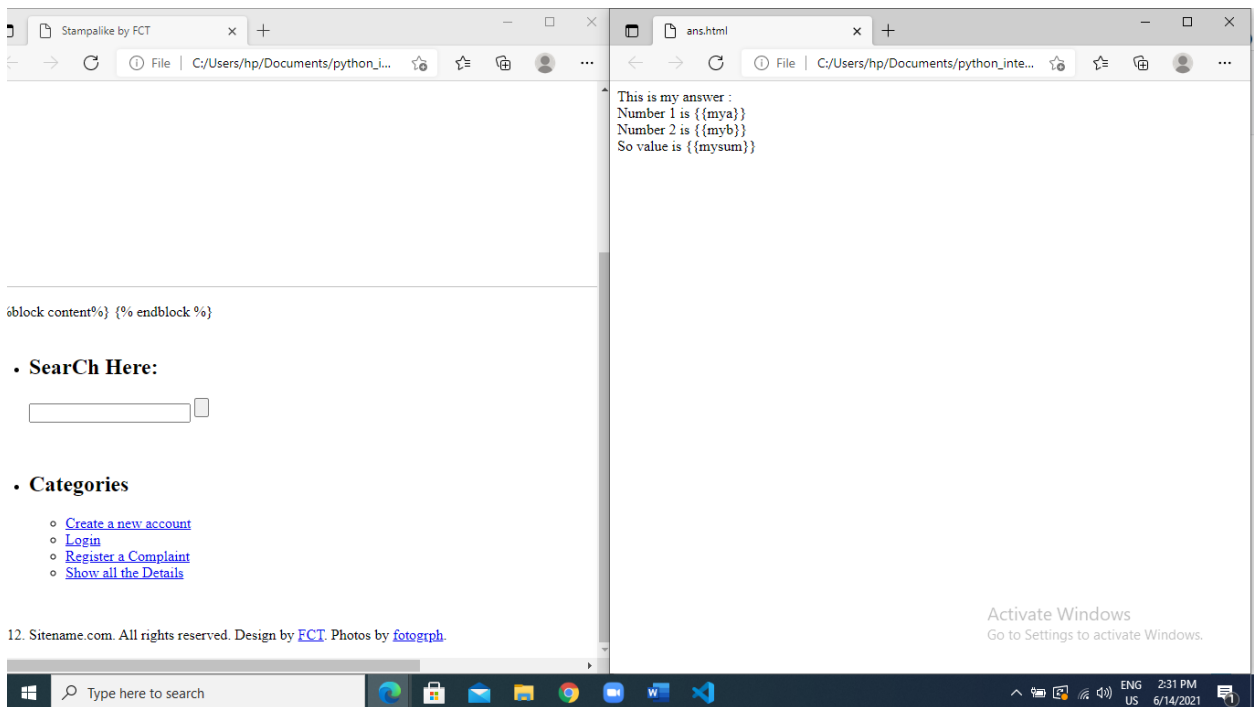


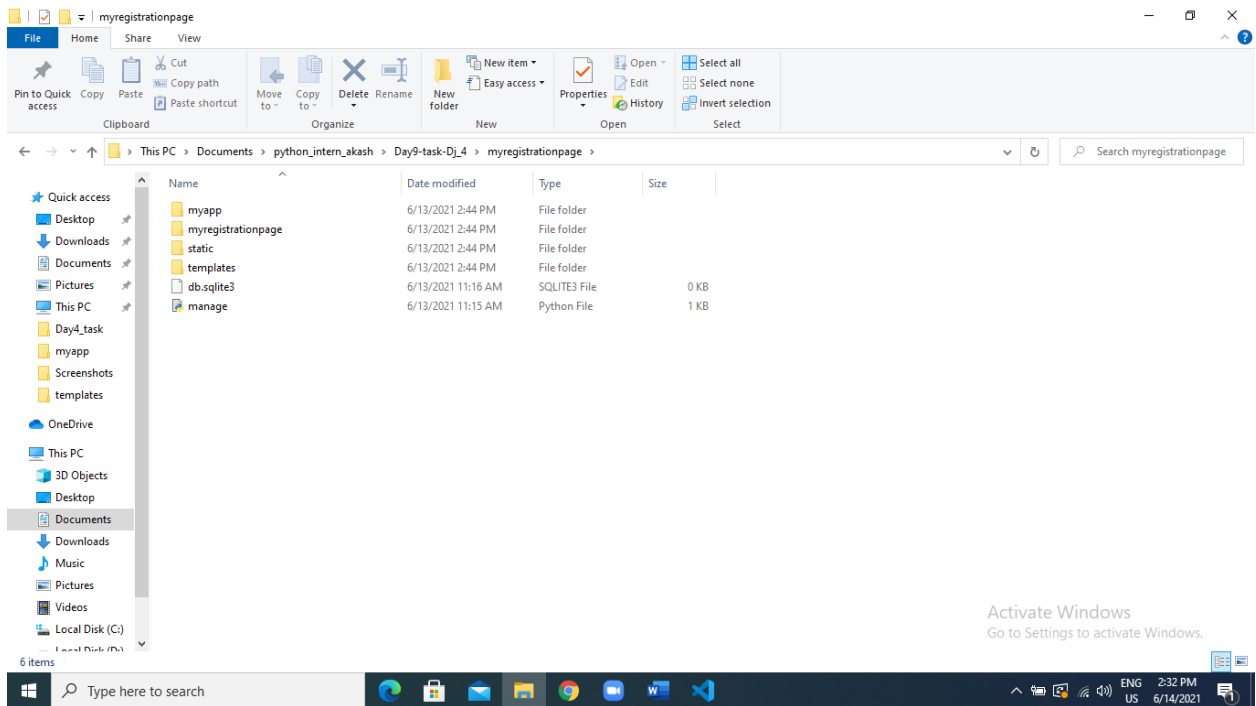


Day 8 : CREATION OF FORM









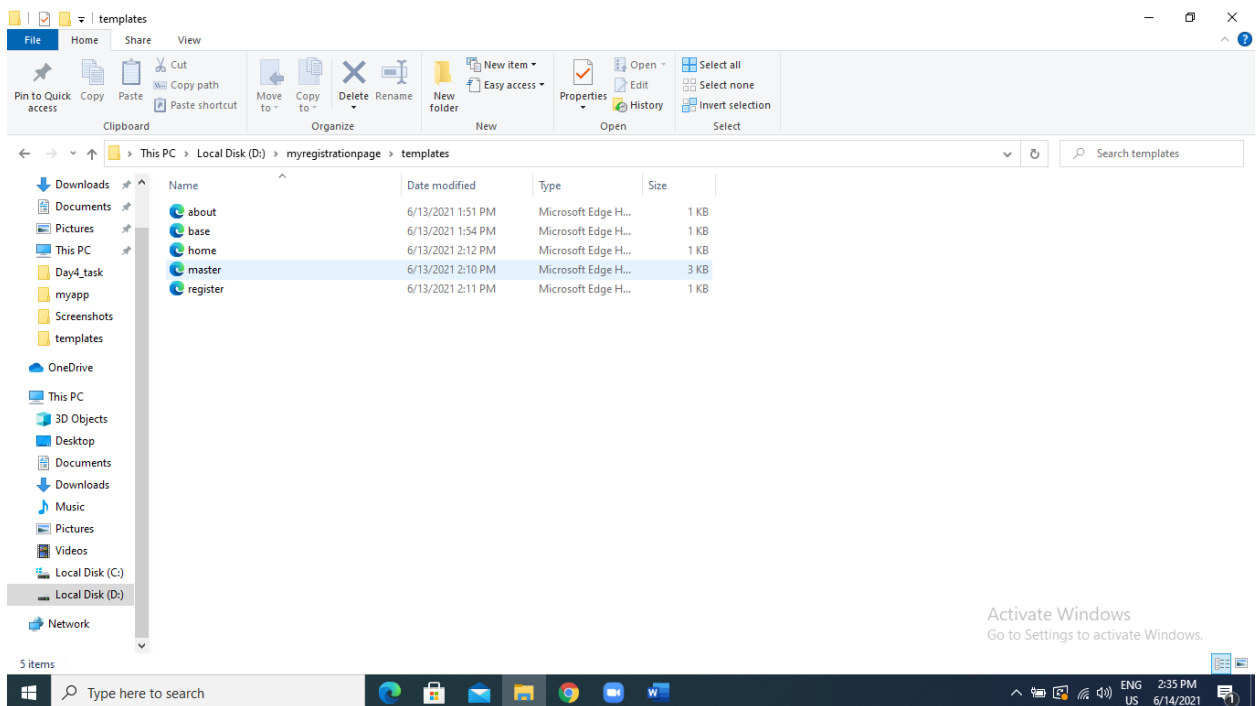
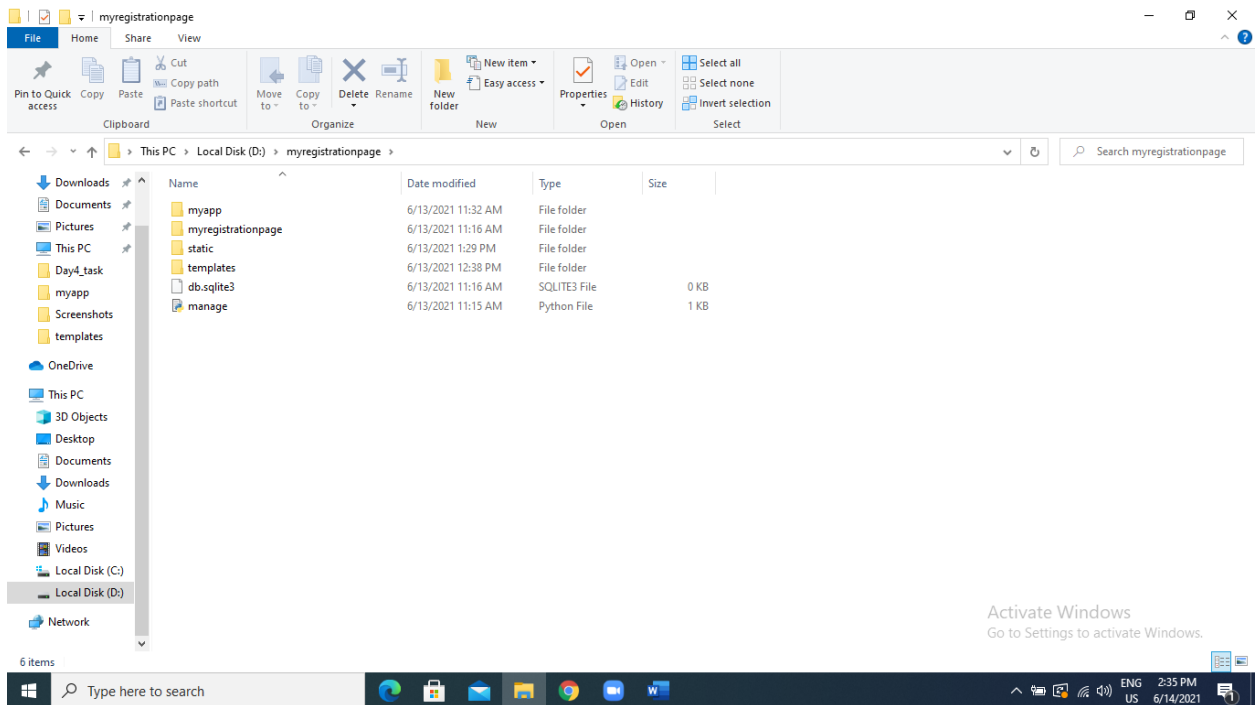
The screenshot shows the Visual Studio Code editor with a file named `urls.py` open. The file contains a Django URL configuration for a registration page. The code includes comments explaining the purpose of the `urlpatterns` list and provides examples for function-based and class-based views. It also shows how to include another URL configuration. The `urlpatterns` list is defined with two entries: one for the admin site and one for the main application.

```
1 """myregistrationpage URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/3.2/topics/http/urls/
5 Examples:
6 Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('', include('myapp.urls')),
22 ]
23
```

The terminal window at the bottom shows the output of a web browser, displaying HTTP requests and responses. The logs include the following entries:

```
[14/Jun/2021 14:38:17] "GET /static/images/page-bgtop.png HTTP/1.1" 200 1219
[14/Jun/2021 14:38:17] "GET /static/images/page-bgbtm.png HTTP/1.1" 200 1316
Not Found: /favicon.ico
[14/Jun/2021 14:38:18] "GET /favicon.ico HTTP/1.1" 404 2968
[14/Jun/2021 14:38:21] "GET /contact HTTP/1.1" 200 2749
[14/Jun/2021 14:38:23] "GET /about HTTP/1.1" 200 2746
[14/Jun/2021 14:38:24] "GET / HTTP/1.1" 200 3000
```

Day 9 :::: Practice by self task ::::: Sign in page

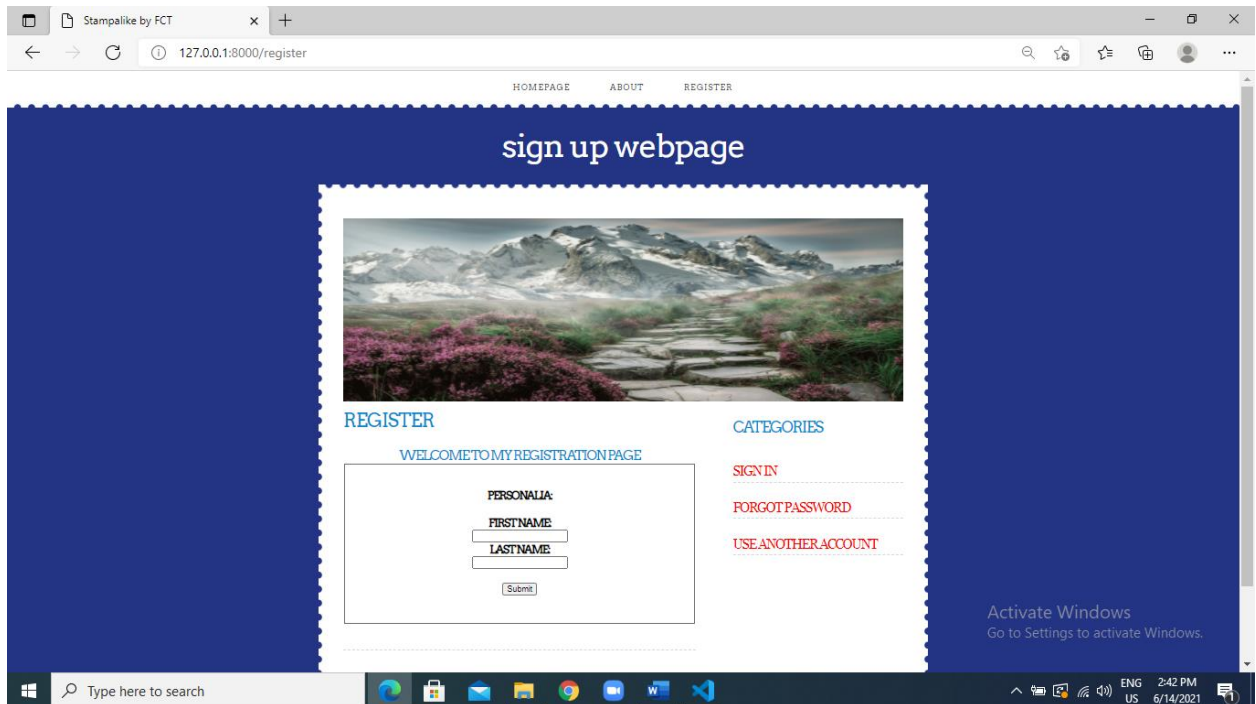
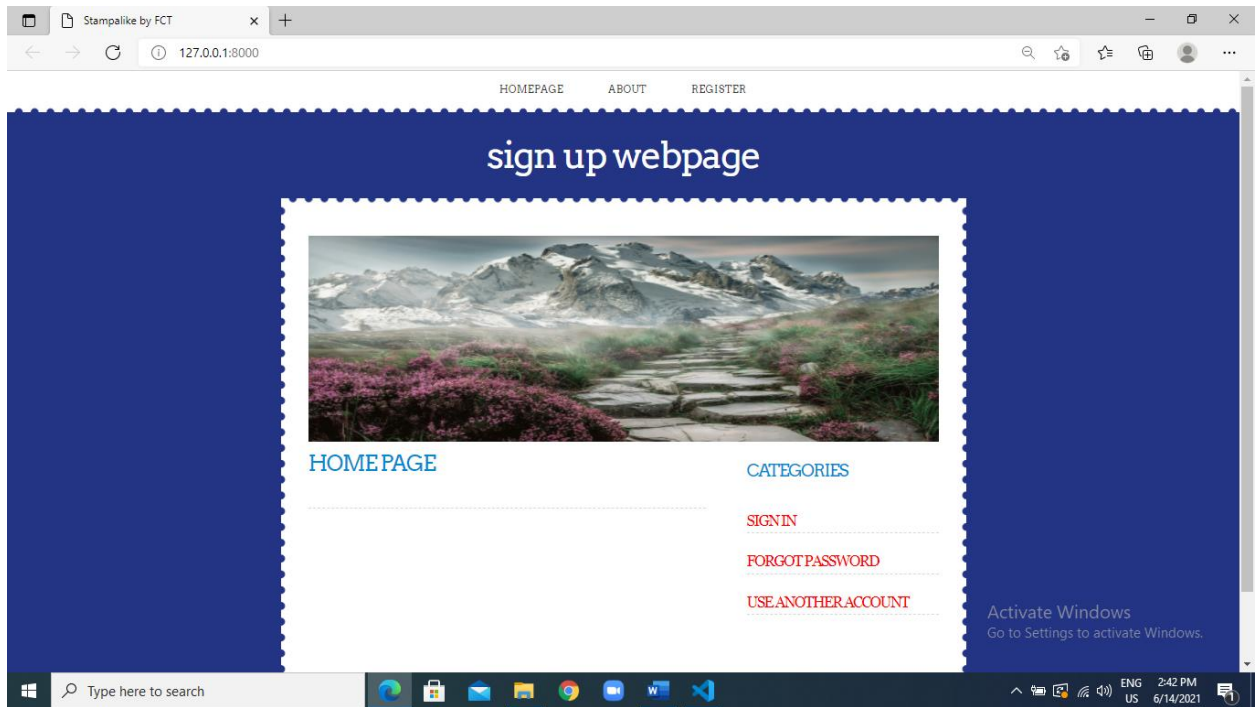


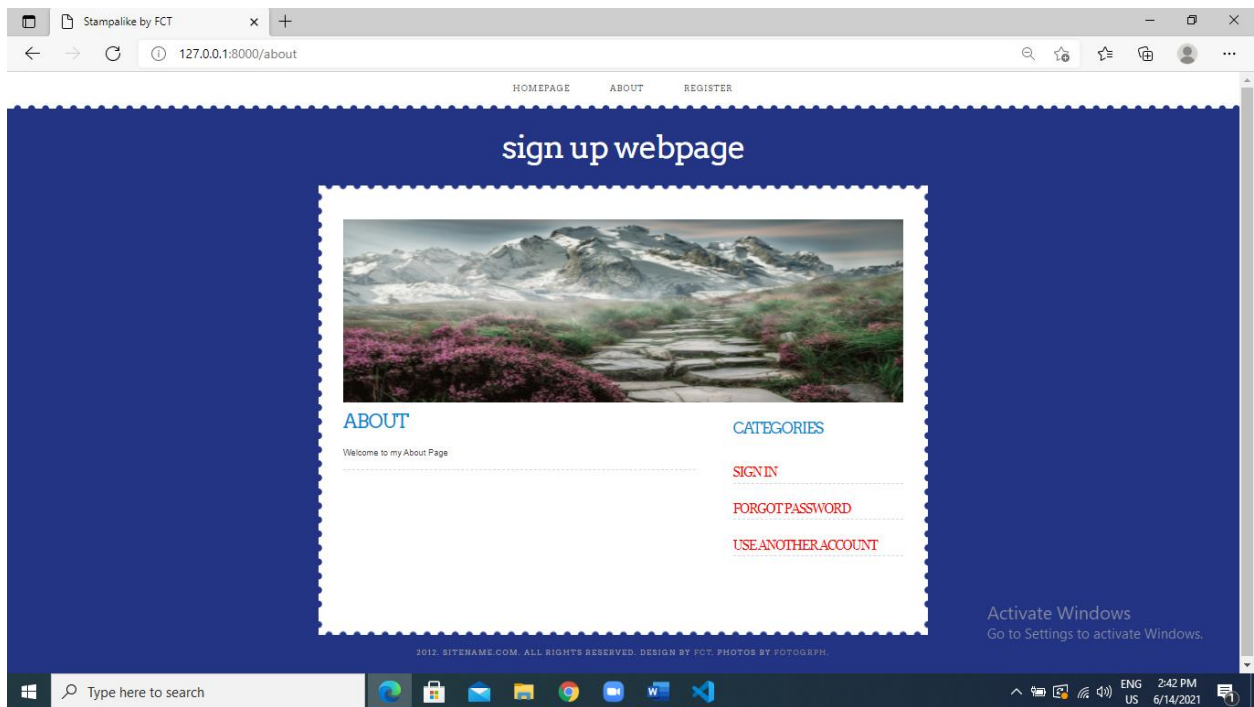
```
File Edit Selection View Go Run Terminal Help views.py - Visual Studio Code
views.py
D: > myregistrationpage > myapp > views.py
1 from django.http.response import HttpResponseRedirect
2 from django.shortcuts import render
3 from django.http import HttpResponseRedirect
4
5 def home(request):
6     return render(request, 'home.html')
7
8 def register(request):
9     return render(request, 'register.html')
10
11 def about(request):
12     return render(request, 'about.html')
13
14
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE python + ^ x

Django version 3.2.4, using settings 'myregistrationpage.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[14/Jun/2021 14:36:24] "GET / HTTP/1.1" 200 2071
[14/Jun/2021 14:36:24] "GET /static/nature.jpg HTTP/1.1" 200 866962
Not Found: /favicon.ico
[14/Jun/2021 14:36:24] "GET /favicon.ico HTTP/1.1" 404 2620

Python 3.9.5 64-bit 0 0 0 Ln 14, Col 1 Spaces: 4 UTF-8 CRLF Python ENG 2:36 PM 6/14/2021





THANK – YOU!!!

WORK SUBMITTED BY : RIYA SARKAR

EN NO- 181250107049

SSIT-BHAT

