**Python with Django Internship Report**

**Personal Details**

Name : Asma Naaz Gazi

College Name : LJ Institute of Engineering & Technology

Degree : BE

Semester : 7th

Github profile : https://github.com/asmagazi

**Company Details**

Company Name : Akash Technolabs

External Guide : Akash Padhiyar

Training Duration : 26-05-2021 to 21-06-2021

Index

|  |  |  |
| --- | --- | --- |
| index | Task Details | Page No. |
| 1 | Registration Form  Table-based Website | 4 |
| 2 | Variables and Datatypes | 8 |
| 3 | Basic Programs of Python | 11 |
| 4 | Functions in python | 13 |
| 5 | Python OOP’s  Concept | 15 |
| 6 | Django Framework | 21 |
| 7 | Django Basic Programs | 22 |
| 8 | Django Theme Integration | 25 |
| 9 | Django Methods  (GET-POST) | 28 |
| 10 | Django db | 32 |

Introduction:

Django with python

For better understanding of Django, we must have some basic knowledge about the python.

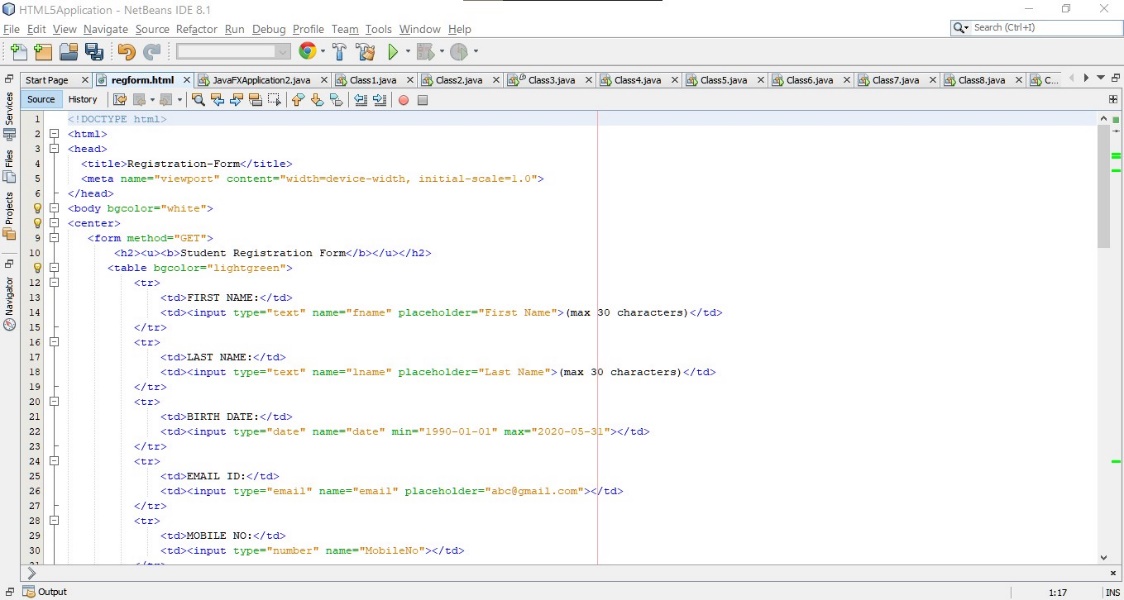
Python is a programming language whereas, Django is a python Framework for full-stack web application development and server development.

It is used for rapid development, pragmatic, maintainable, clean design and secure website.

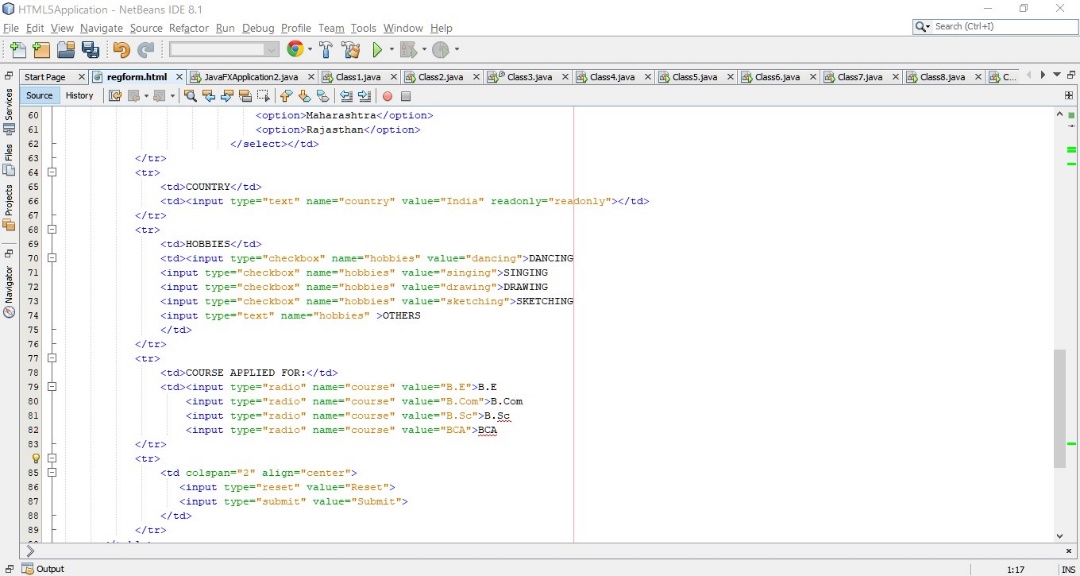
There are lots of Advantages of Django Framework over any other Frameworks.

* Easy to use.
* Simple and fast.
* Excellent Documentation for real world application.

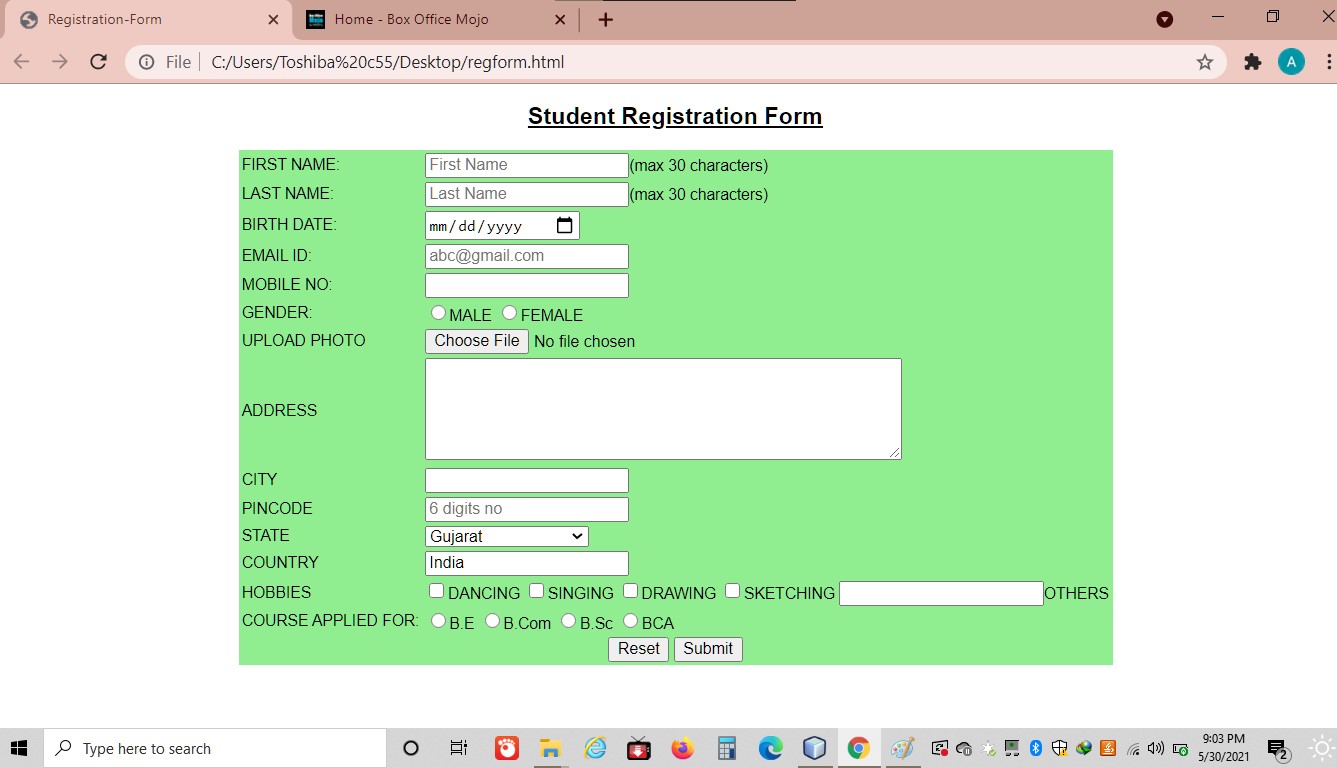
**Registration Form Input**







**Registration Form Output**



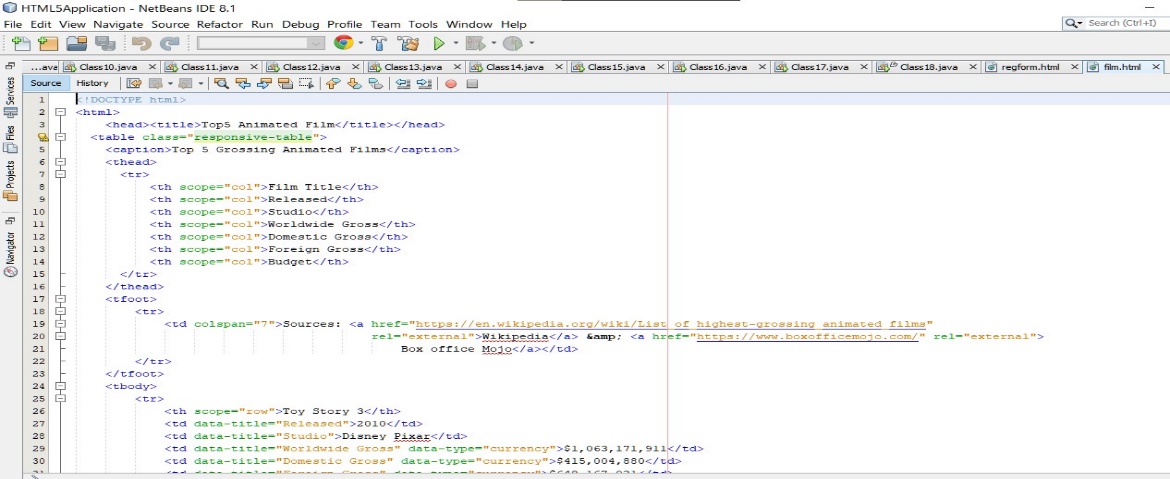
This is a Simple Student Registration Form created with the help of HTML and CSS.

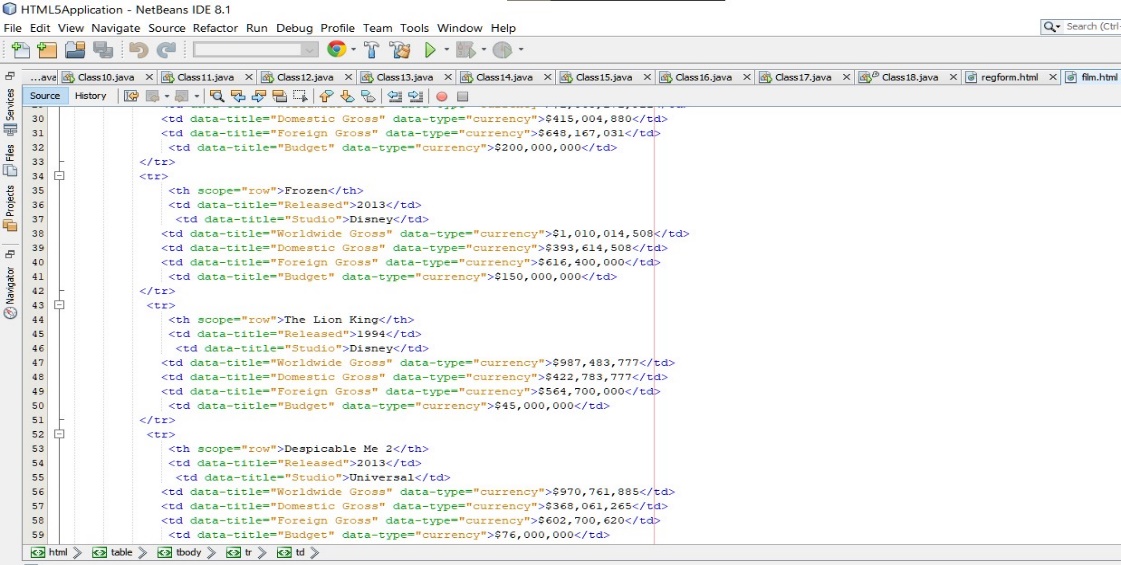
There are Various text fields, Number fields,

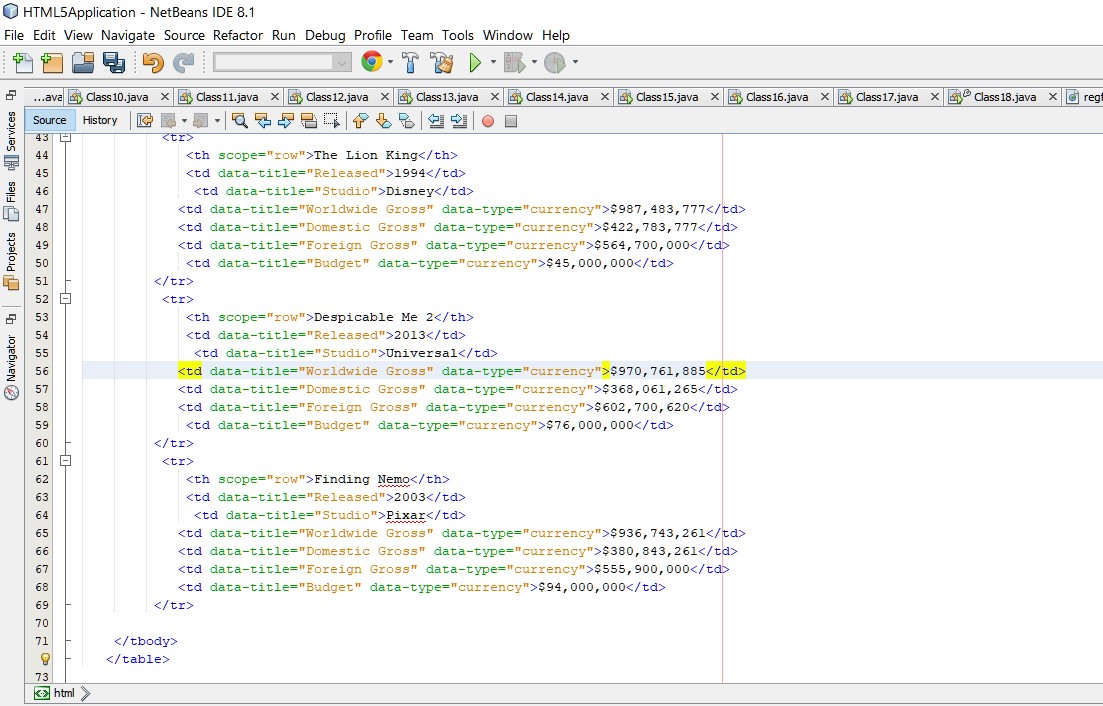
check boxes and Radio button and here we get input from user.

We get output when we run program on any of the browser.

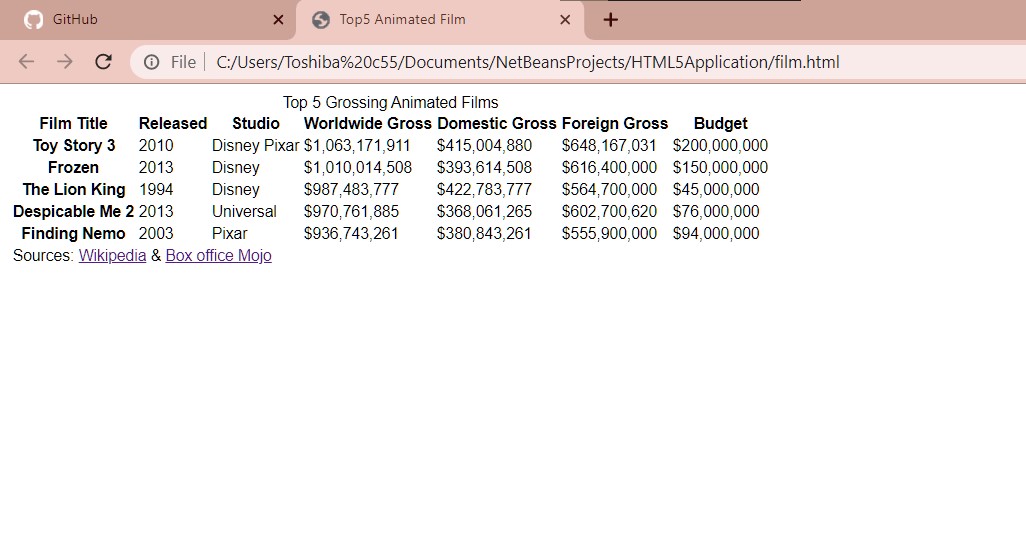
**Table-based website Input**





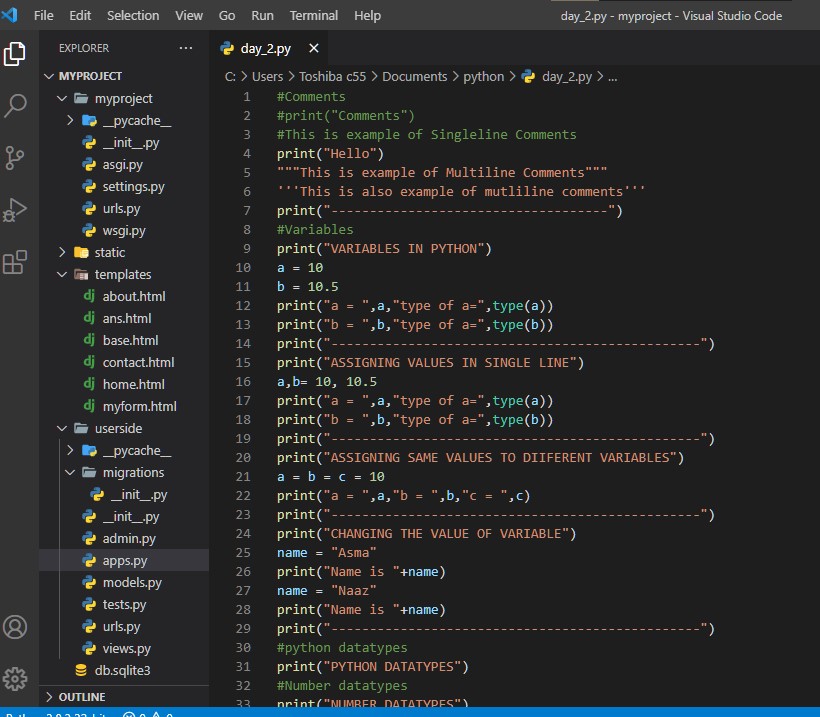


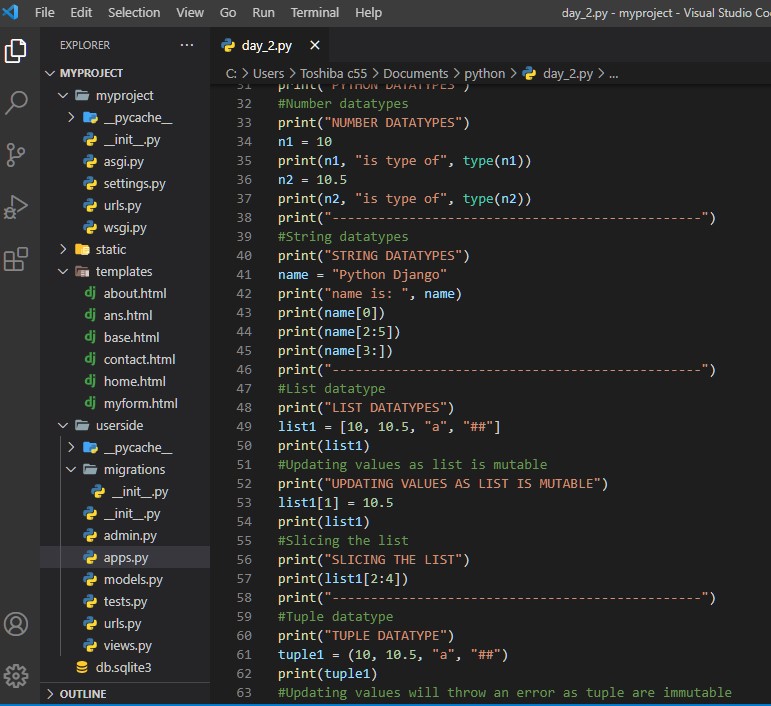
**Table-based Output**

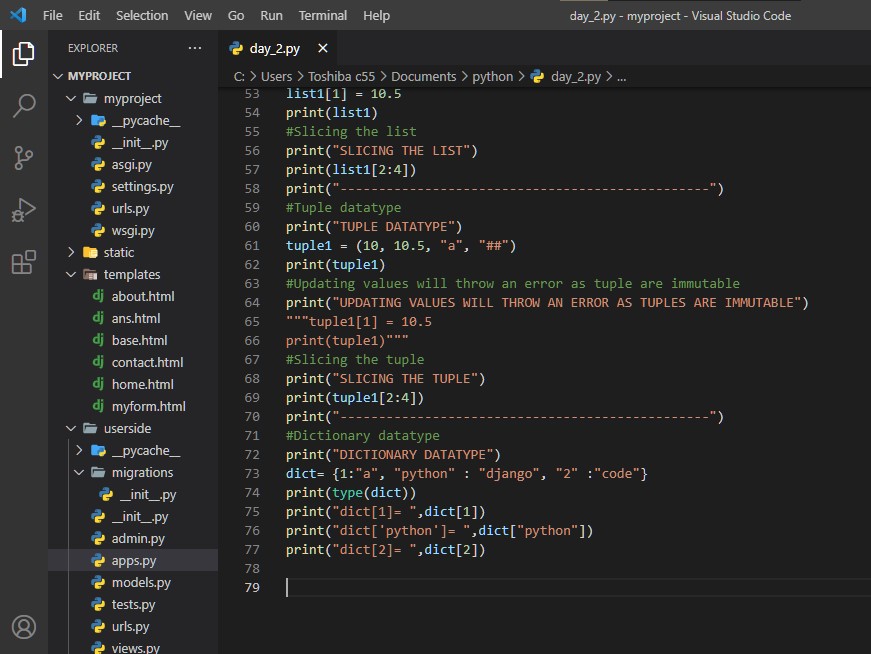
These website shows the data about the Top 5 Grossing Animated Films.

It is a small table-based website which was created using HTML-5 and Css

* **Python Basics**







* Here, the comments is shown that single line comment is written with hash ‘#’ Symbol whereas,multiline comments are defined with in single or double ‘triple quotes’.
* The print() function is also shown where we print the given statement.
* Here it is also shown that how to declare the single variable,multi variable, how to change the value of the variables,how to manipulate the variables in a different ways.
* **PYTHON DATATYPES:**

There are five different data types in Python:

1. Number Data type
2. String Data type
3. List Data type
4. Tuple Data type
5. Dictionary Data type

Note: All this datatypes are shown above

1. In number Data type we can assign different values to variable and with help of type() and isinstance() function , we can find the type of variable.
2. In string Data type we can concat two strings and also can splitting of string with split function.
3. List is mutable and very much similar to array object. We can access list objects with the help of index numbers. Also we can drop and add objects in list.
4. Tuple is immutable Data type and can accessed by index numbers. If we try to update any existing value then it throws error.
5. Dictionary use keys and values to store objects.

* **OPERATORS IN PYTHON:**

1. **Arithmetic Operator** :

(+, -, \*, /, %, //, \*\*)

1. **Comparison Operator:**

(>, <, ==, !=, >=, <=)

1. **Logical Operator:**

(and, or, not)

1. **Assignment Operator:**

(=, +=, -=, \*=, /=, %=, //=, \*\*=)

1. **Membership Operator:**

(in, not in)

1. **Identity Operator**:

(is, is not)

* **INPUT FUNCTION**: In this function, we can take the value from user and by default the value is string value but we can convert into Integer by using typecast.
* **CONDITIONAL STATEMENT** : There are basically three types of conditional statement we use in python:

1.if

2.Else

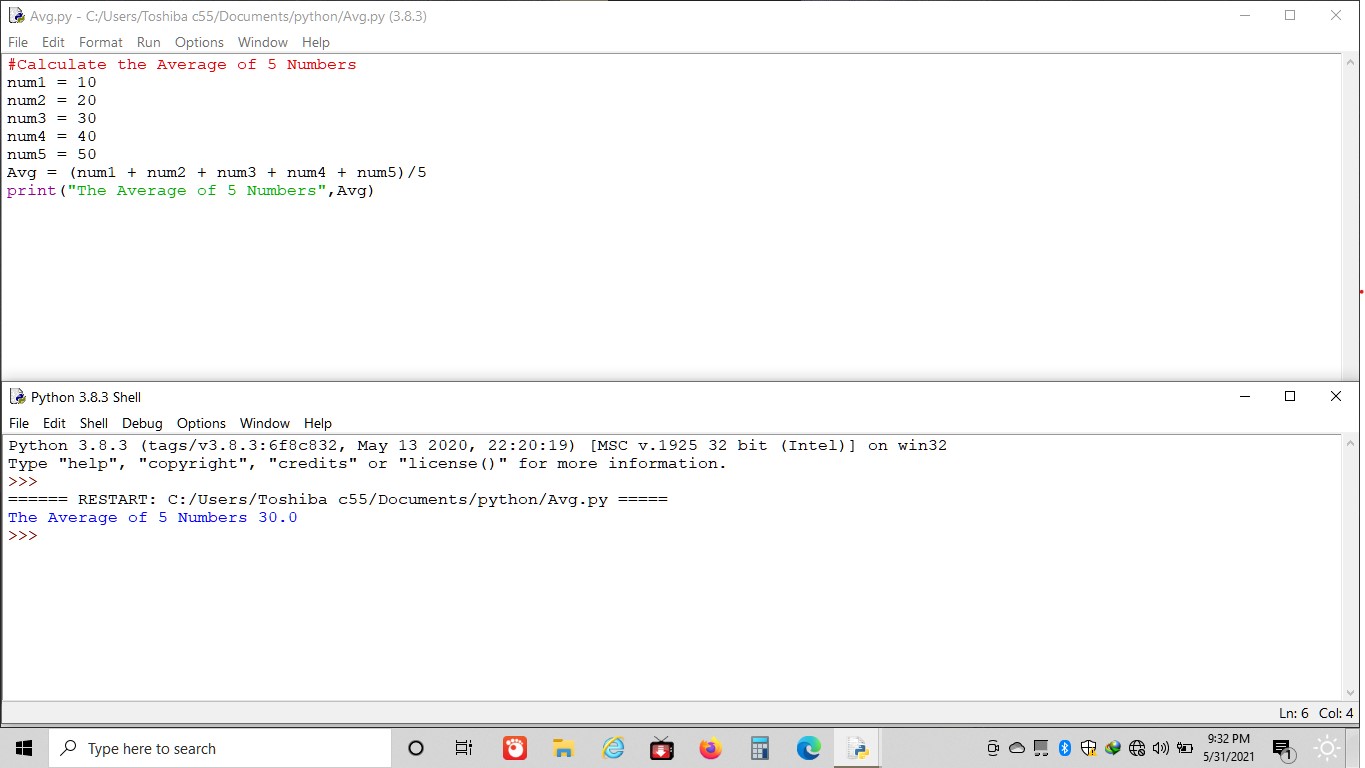
3.Nested if…else

* **LOOP :**

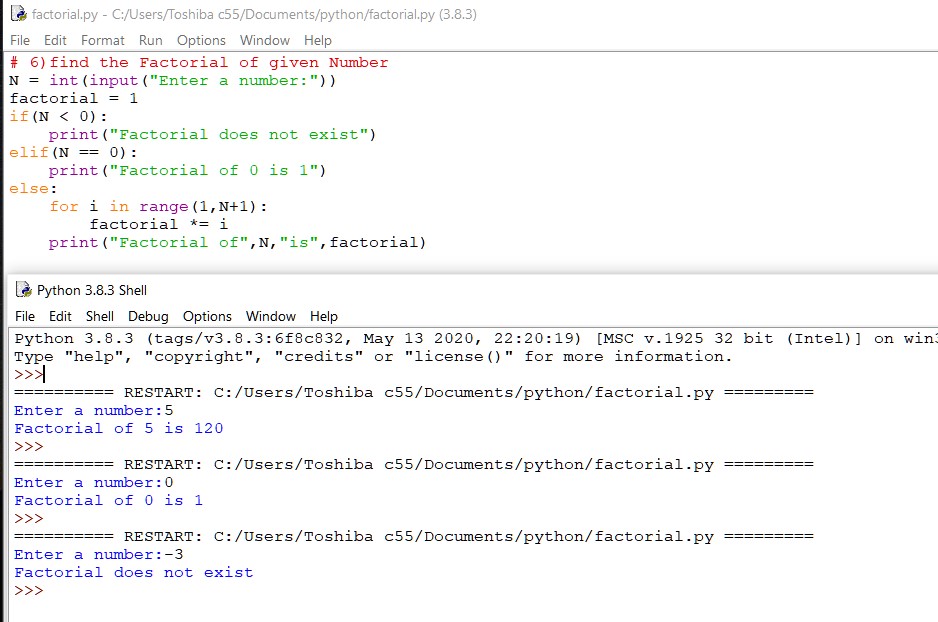
1. For loop
2. While loop

**Average**

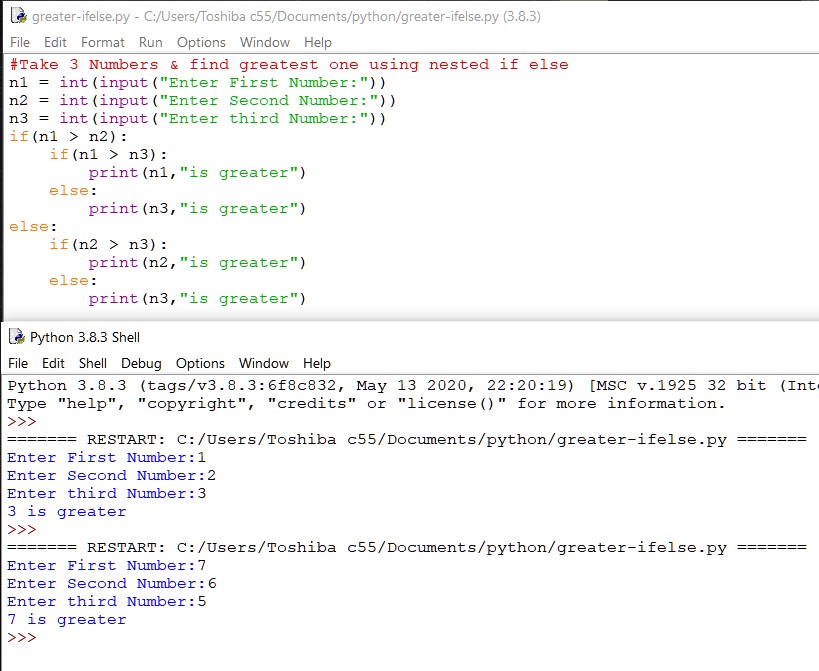


**Even-odd** 

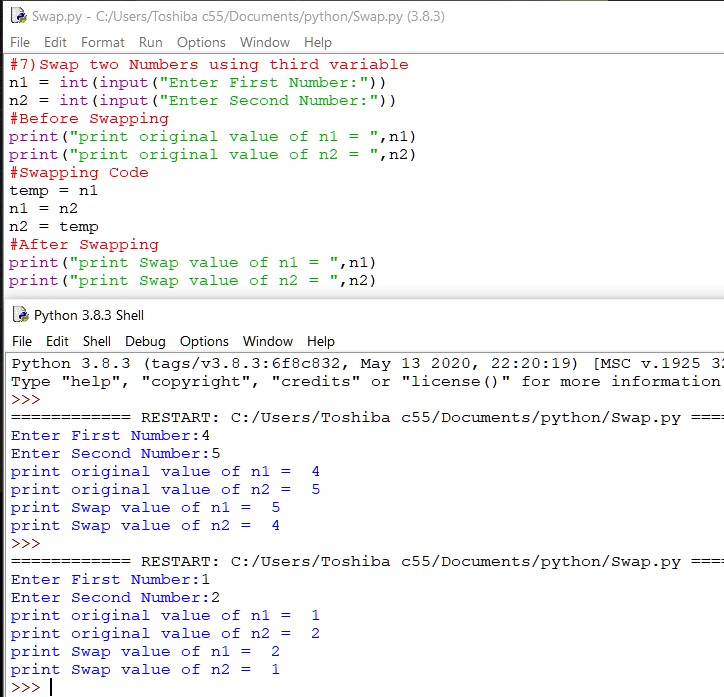
**Factorial**



**greater**



**Swapping**



In above all the examples like finding Average, greater number, smaller number, min-max, Swapping ,even-odd we make use of Conditional statement, loops, and aslo use input() function to take the value from the user.

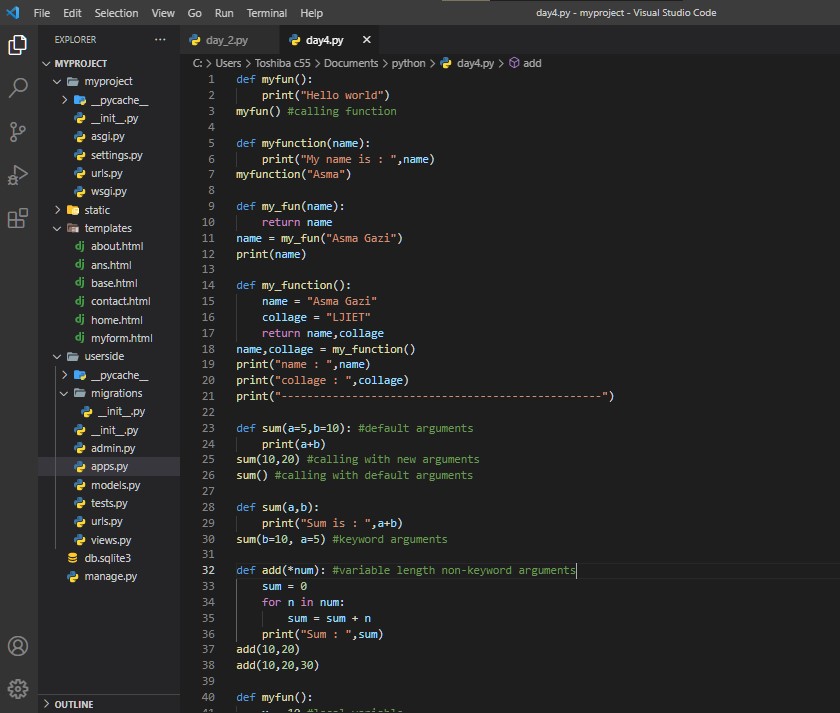
* **FUNCTIONS IN PYTHON :**

Function is a group of related statements that perform specific task.

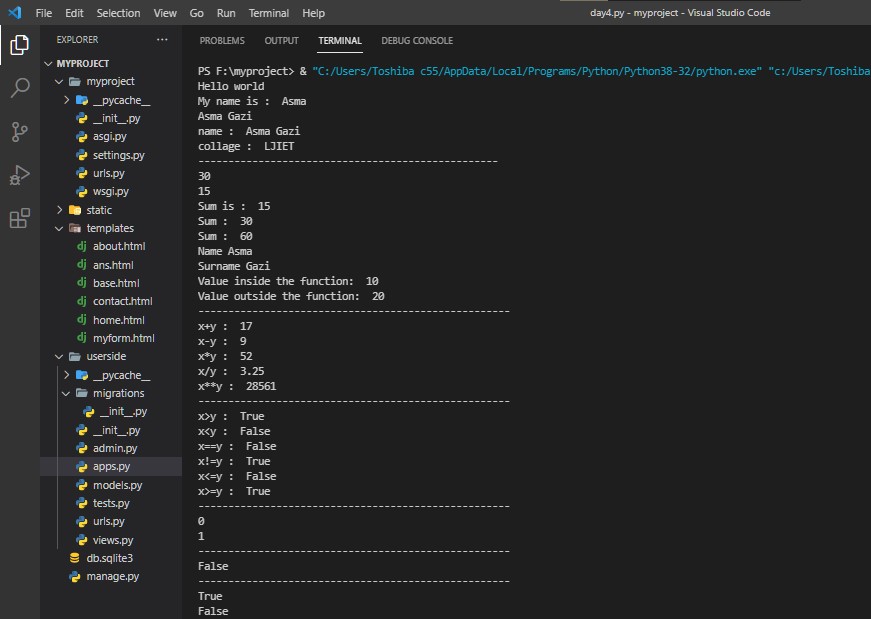
* **def** keyword marks the start of function header.
* A **function name** to uniquely identify it.
* A **colon(:)** to mark the end of function header.
* The **function body**.
* An optional **return statement** to return a value from the fuction

-->Function example

**Input**



**Output**

****

Here, In the above examples different functions are shown

* function without parameter
* function with parameter
* function without parameter return a value
* function with parameter return a value
* **PYTHON FUNCTION ARGUMENTS**

There are three types of python function Arguments using which we can call a function,

* Default Arguments
* Keyword Arguments
* Variable-length Arguments
* **PYTHON OOP’S CONCEPT**

**Class**: A class is a blueprint for the object that are defined by the **Class** keyword.It contains datafields to store data and methods.

**Object** : It is an instance of a class.

**self**: This parameter refers to object which invokes the method, and all the methods including some special method like intializer have first parameter self.

**Constructor:** A constructor is a class function that instantitates an object to predefined values.

It is a \_\_init\_\_init() constructor

**There are two types of Constructor**:

* Default Constructor
* Parameterized Constructor
* **INHERITENCE**:

Inheritance allows programmer to create a general class first then later extend it to more specialized class.

--There are five types of inheritance:

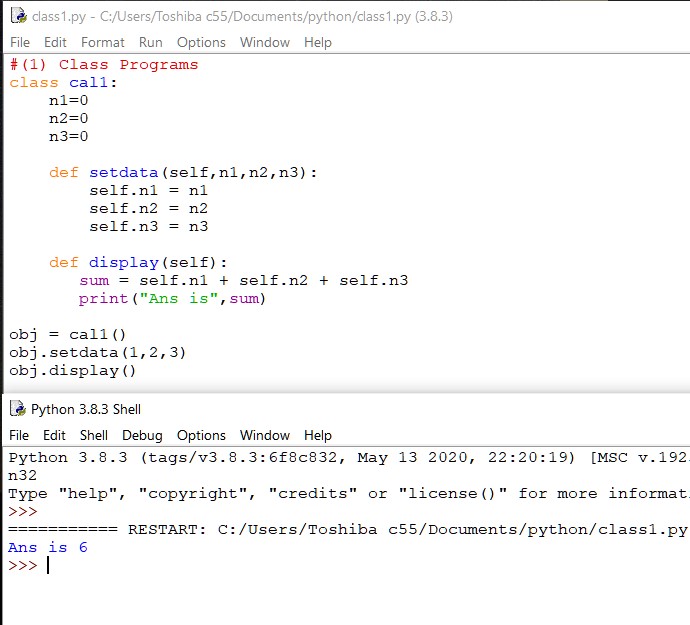
1. Single-level Inheritance:

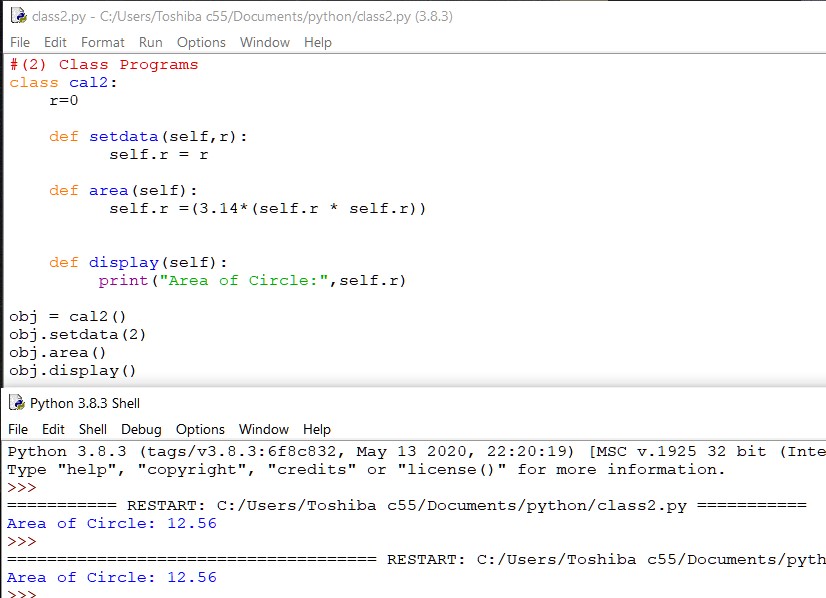
2. Multi-level Inheritance:

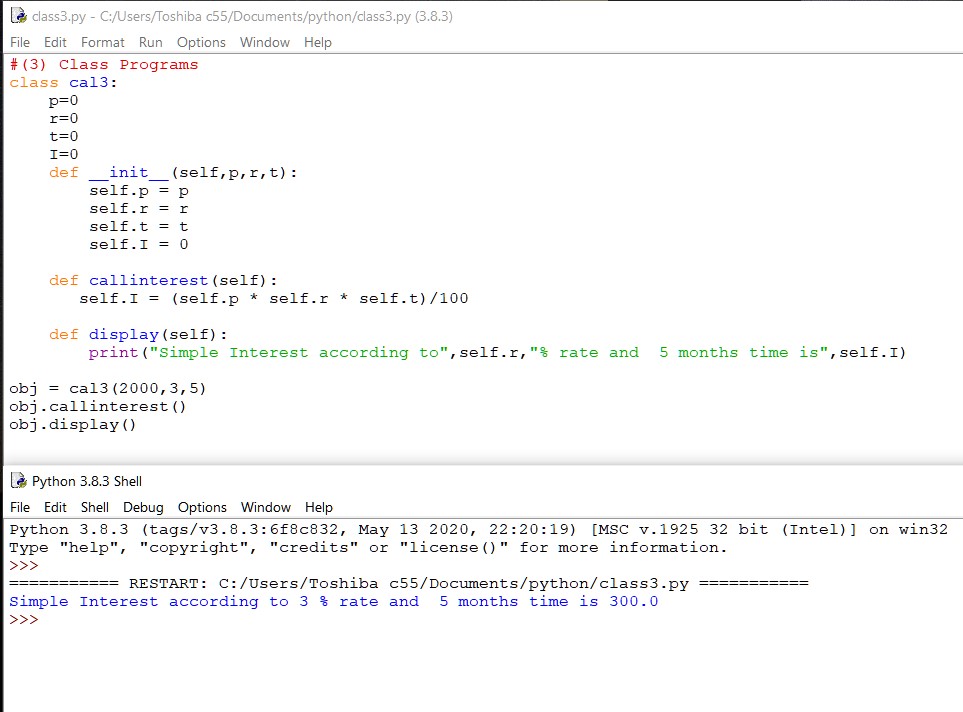
3. Multiple Inheritance:

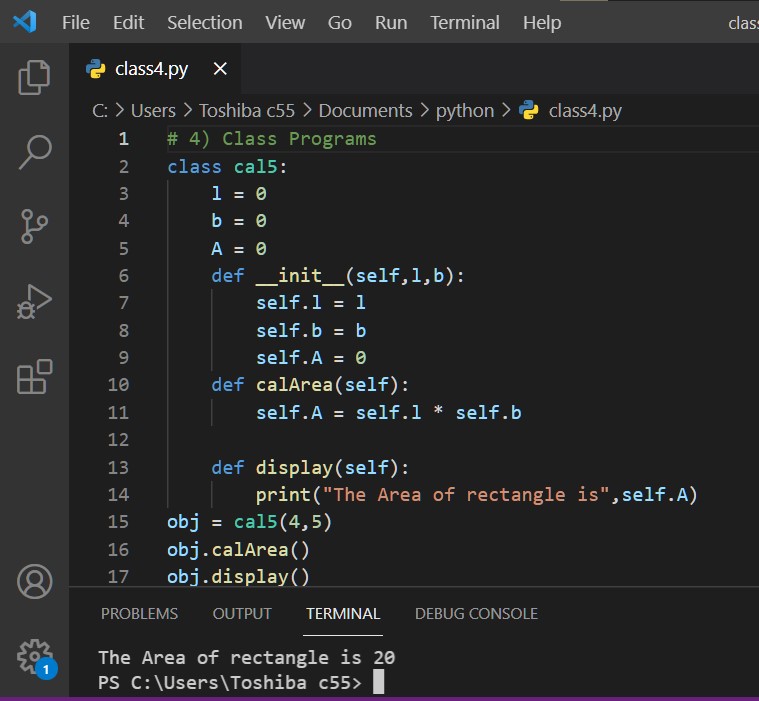
4. Hierarchical Inheritance:

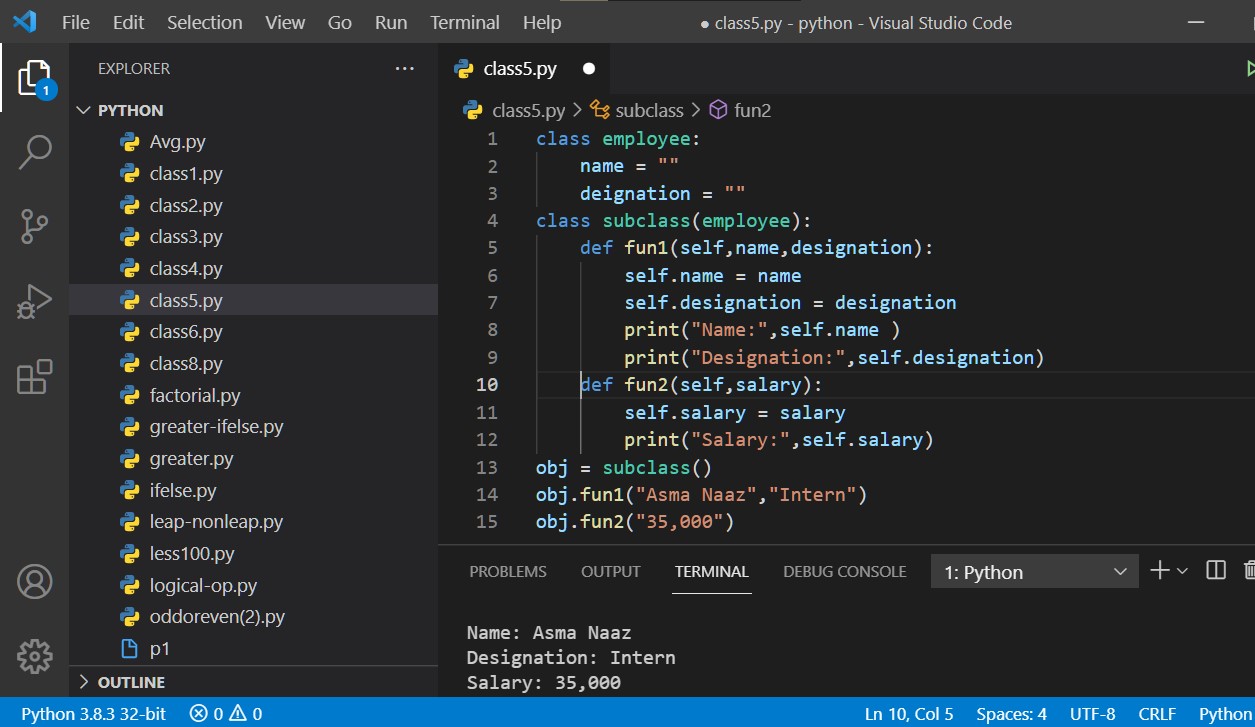
5. Hybrid Inheritance:

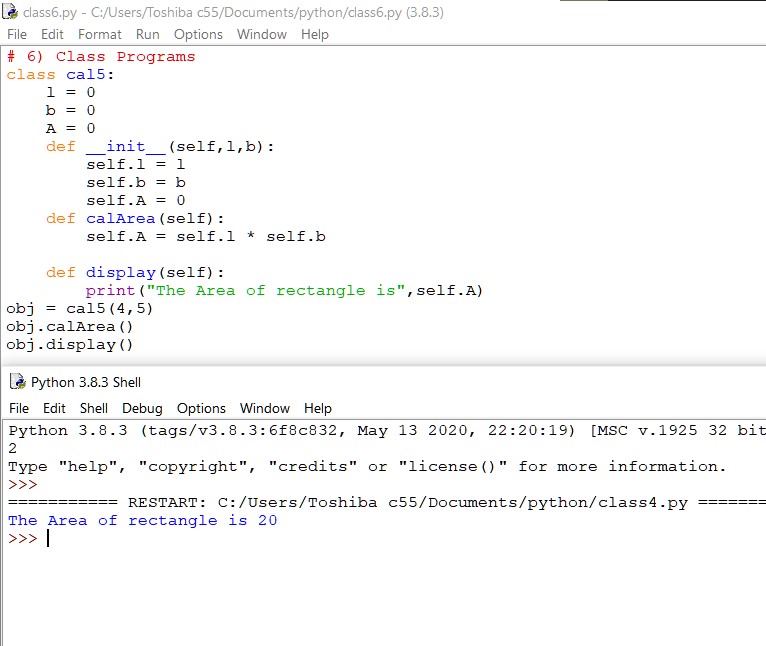


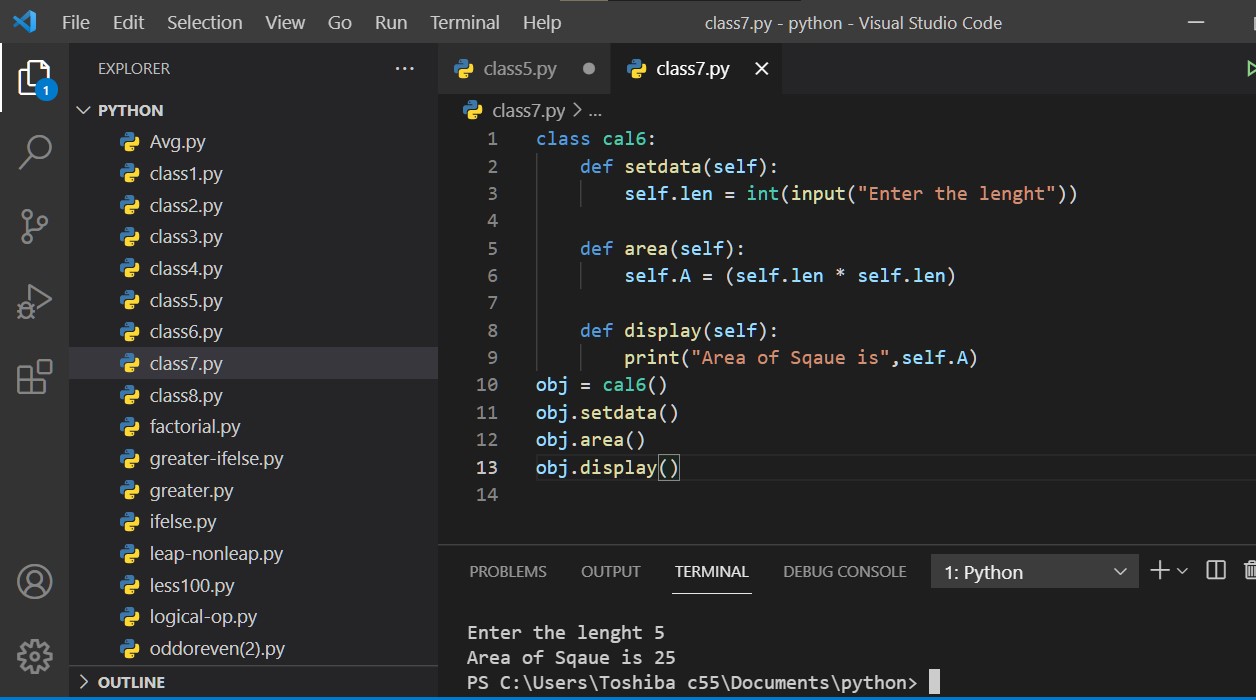


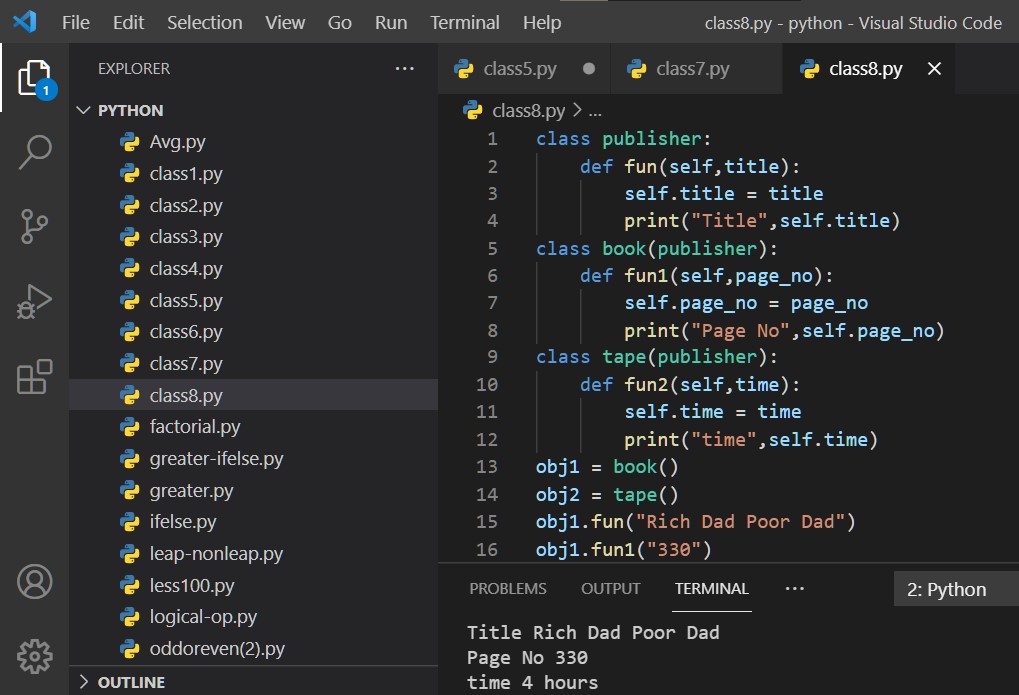


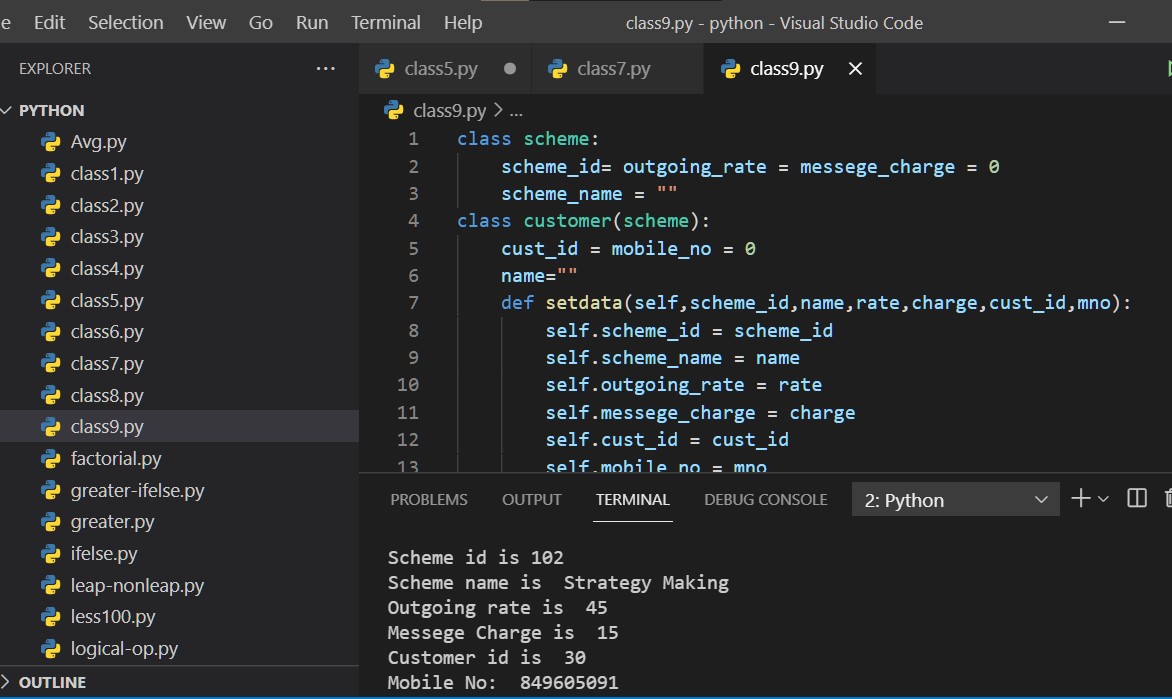


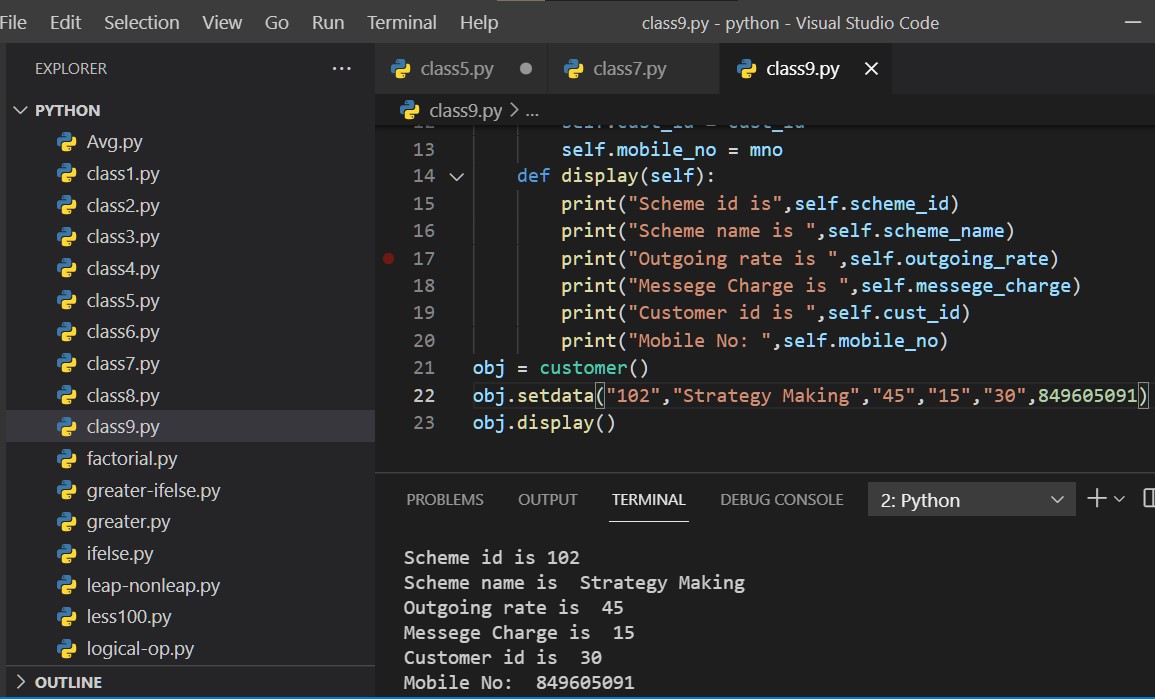


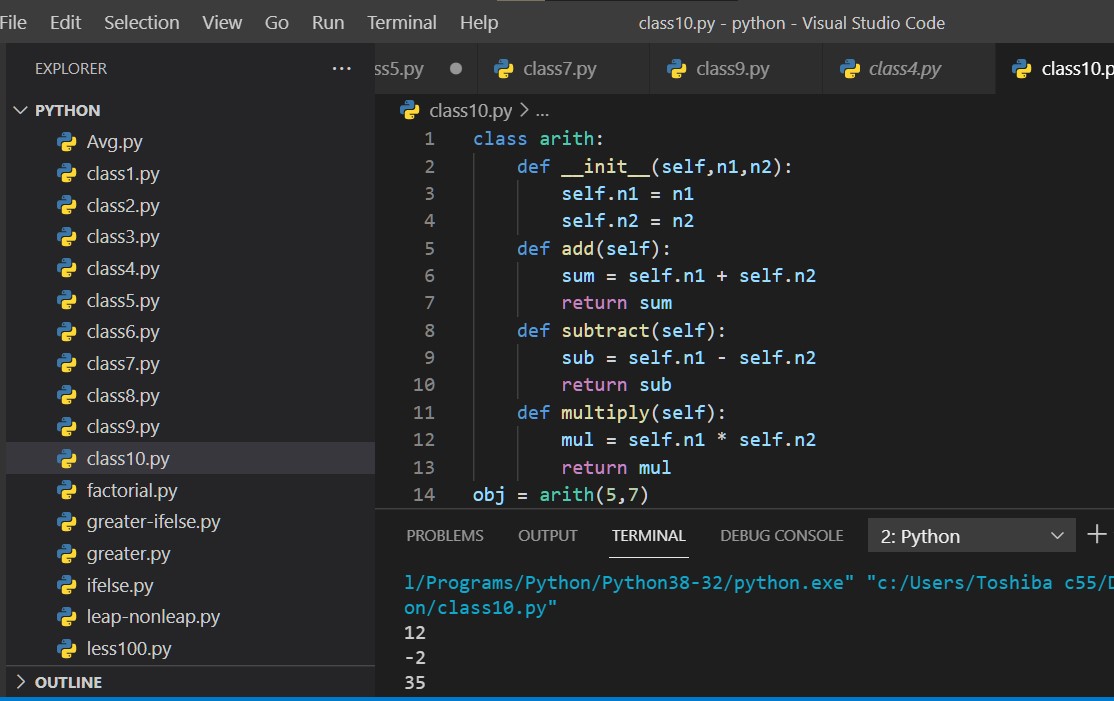












All the above examples has used oop’s concept.

These all programs includes Class, object,self,\_\_init\_\_() constructor and different methods.

It also include inheritence concept and inherit to single or mutiple classes.

In oop’s concept, method overloading and method overriding happens.

* **DJANGO FRAMEWORK**
* Django is a free and open source web application framework, written in Python.
* Django was developed by Adrian Holovaty and Jacob Kalpan-moss at World Online News for efficient development.
* A web framework is a set of components that helps you to develop websites faster and easier.
* Django framework follows the DRY(Don’t Repeat Yourself) principle.
* It provides support for multiple databases such as MySQL, SQLite, and Oracle.

**Django follows MVT:**

* Model :

-Defines the data structure.

-Takes care for querying the database.

* View :

-Defines what data should be presented.

-Returns HTTP response

* Template :

-Renders the data in suitable format

-HTML/XML/etc..

* **DJANGO PROJECT**

In this project, first of all I had installed Django in my command prompt using command “pip install django” and upgrade its version to the latest one.

And then I had created my project called as a ‘myproject’ by using command “django-admin startproject myproject”,then after changing directory to that project I had created one app called as ‘userside’ by using a command “django-admin startapp userside”

After all these, I had run the program using command “python

manage.py runserver” then got one url <http://127.0.0.1:8000/>

then by clicking on this url got a ‘Welcome Screen’

In application, in views.py first I had import HttpResponse then, I had define Homepageview, Aboutpageview, and

Contactpageview for the navigation purpose.

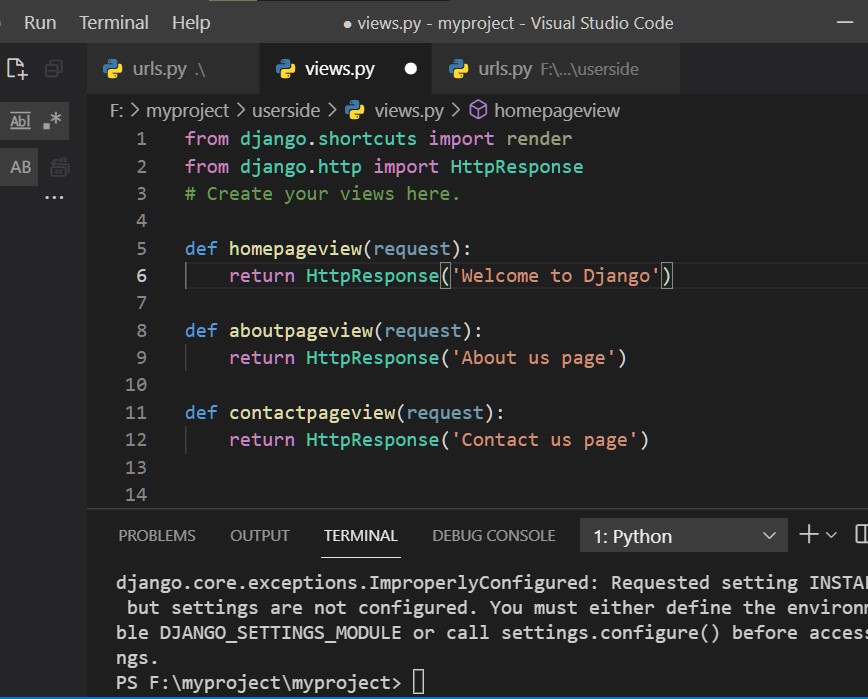
And also need to define the url and patterns in userside-url.py and need to access all the views here from views.py

There is need to register userside-url.py into the myproject-urls.py

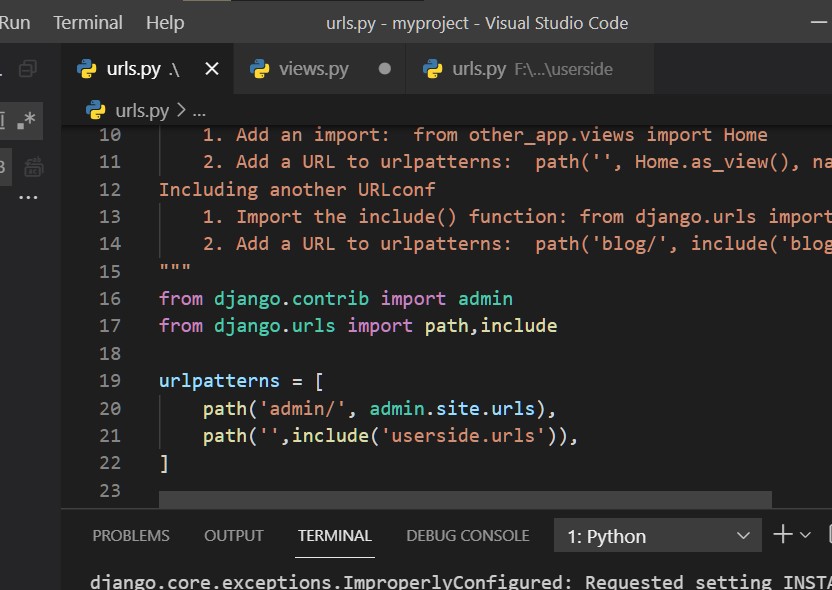
Then, after running the program we got the Homepageview

And we navigate through the port to similarly get the Aboutpageview and as well as Contactpageview.

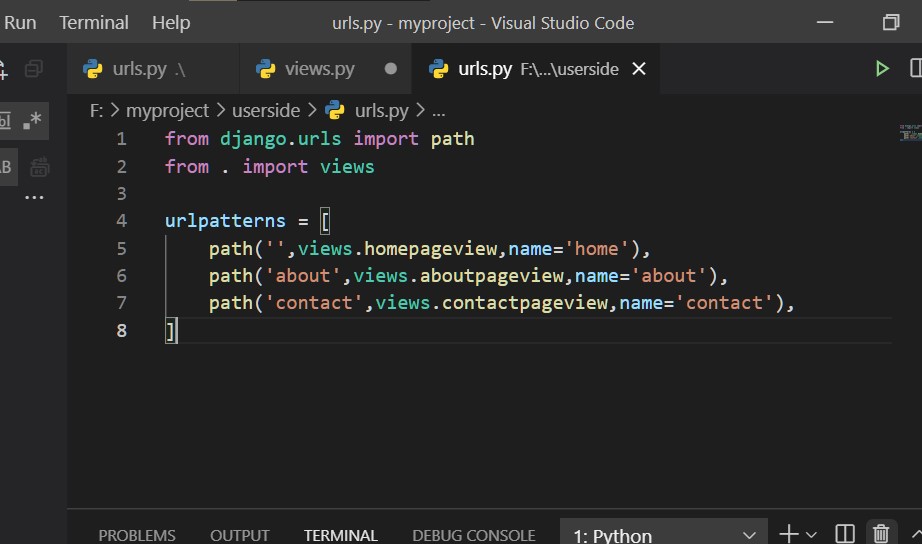
**Views.py**



**Urls.py(myproject)**



**Urls.py(Userside)**



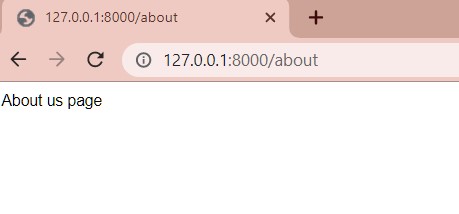
In views.py I had created a Homepageview, Aboutpageview,

Contactpageview.

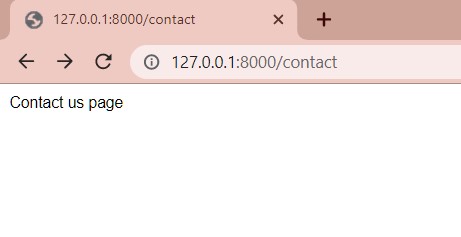
**Homepageview**



**Aboutpageview**

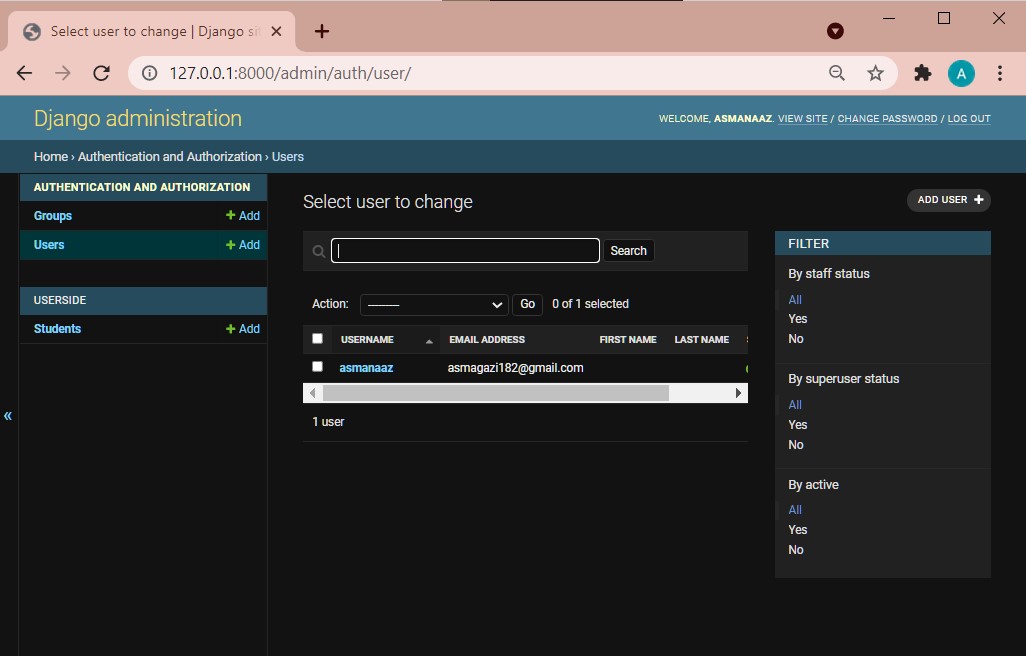
****

**Contactpageview**

****

Django already had its admin panel, where I had created one superuser by using command “python manage.py createsuperuser”.Then it ask for Username, Email address, and password.

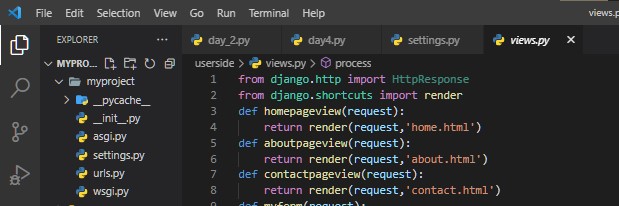
**Django Administration**

****

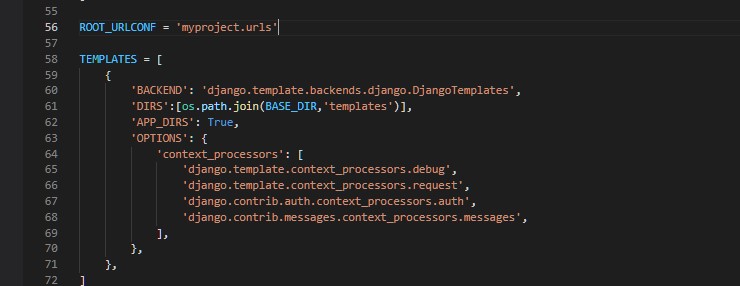
* **Django Theme Integration**

First I had created templates folder in which I had includes three files named as home.html, about.html, and contact.html.

Then I had render and refer to home.html, about.html, and contact.html in views.py



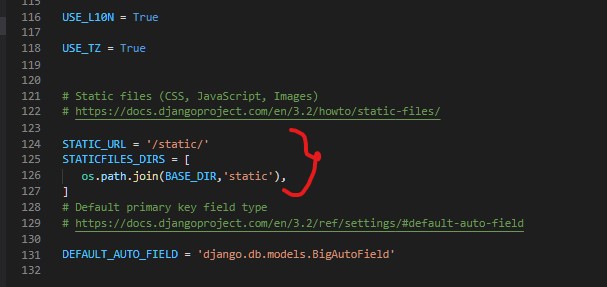
Then I had import os and and in settings.py in templates, I had set the directory to templates



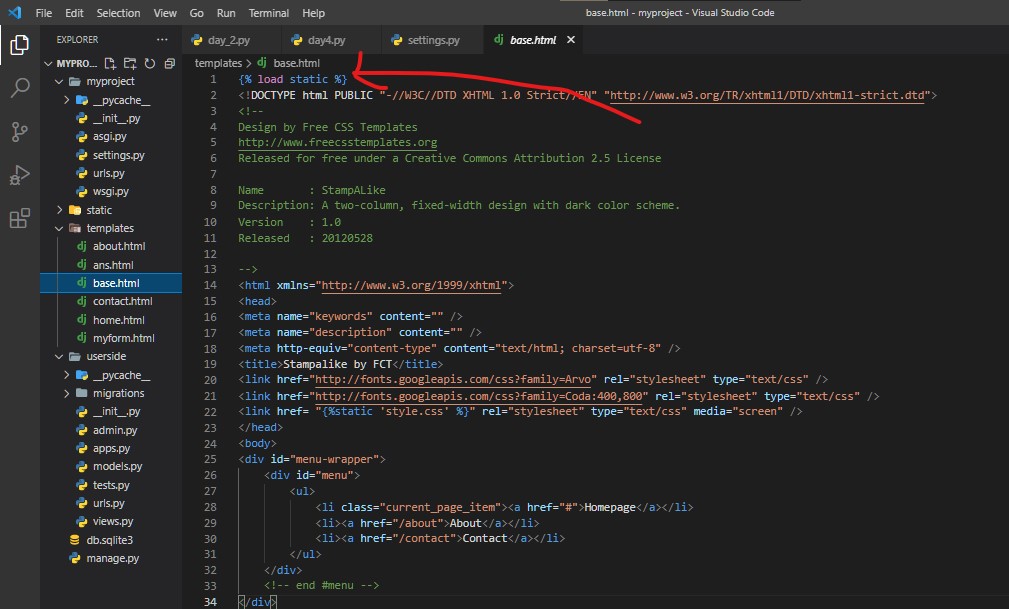
Then I had insert the stampalike theme code in one base.html file in templates folder.

And also load the static in base.html.

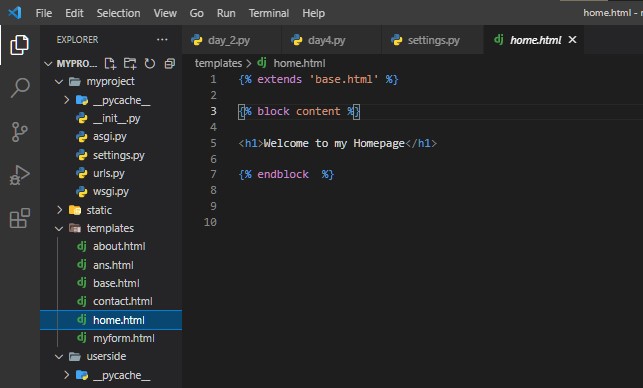
And include the staticfiles directory in settings.py

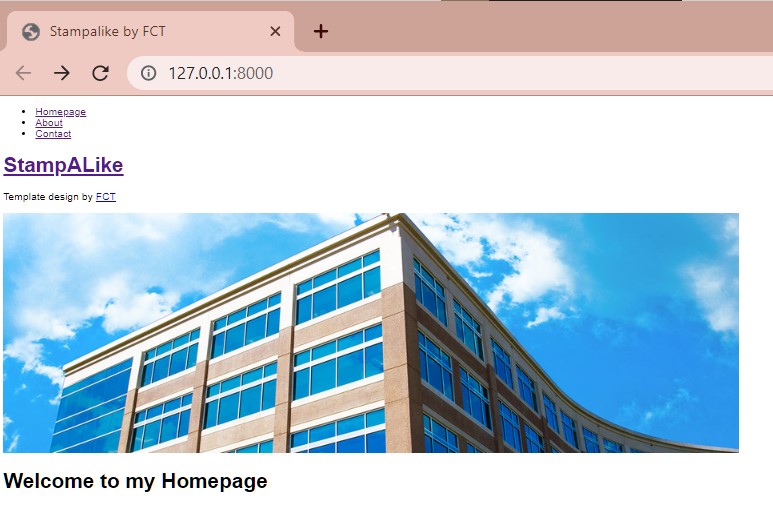


**Base.html**

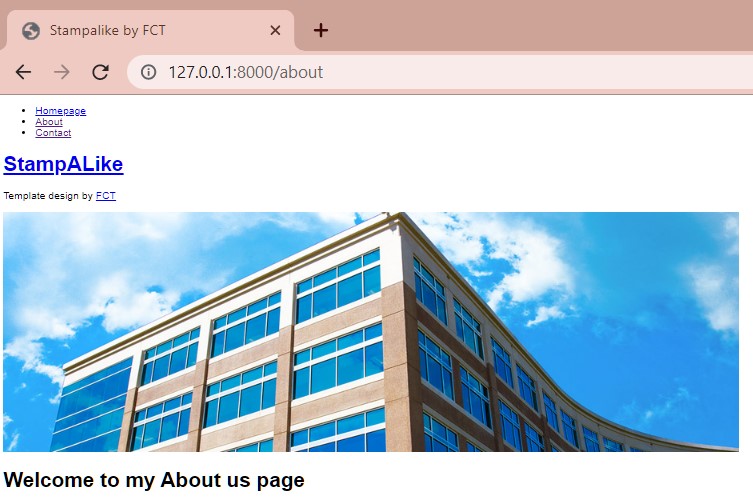
****

After that, I had extends base.html to home.html, about.html, & contact.html to apply the theme

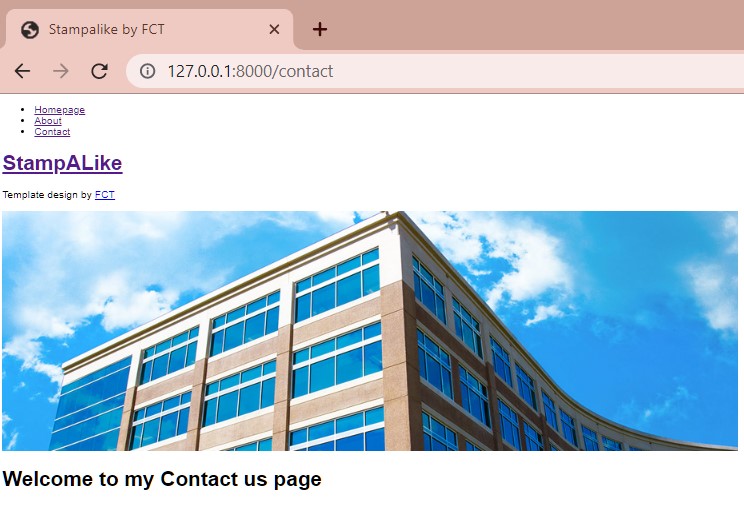


**Home\_theme\_output**

**About\_theme\_output**

****

**Contact\_theme\_output**

****

* **Django methods**

Get and Post Method

In ‘GET’ method data is visible in the url whereas, in ‘POST’ method data is not visible in the url.

So ‘GET’ is not a secure method to pass the sensitive

Data while ‘POST’ is a secure method.

Here in this project I had used ‘POST’ method.

**>>** So, here I had created one Sign Up Form in HTML which contains various datafields.

And include one action called as a formprocess and registered in urls.py also.

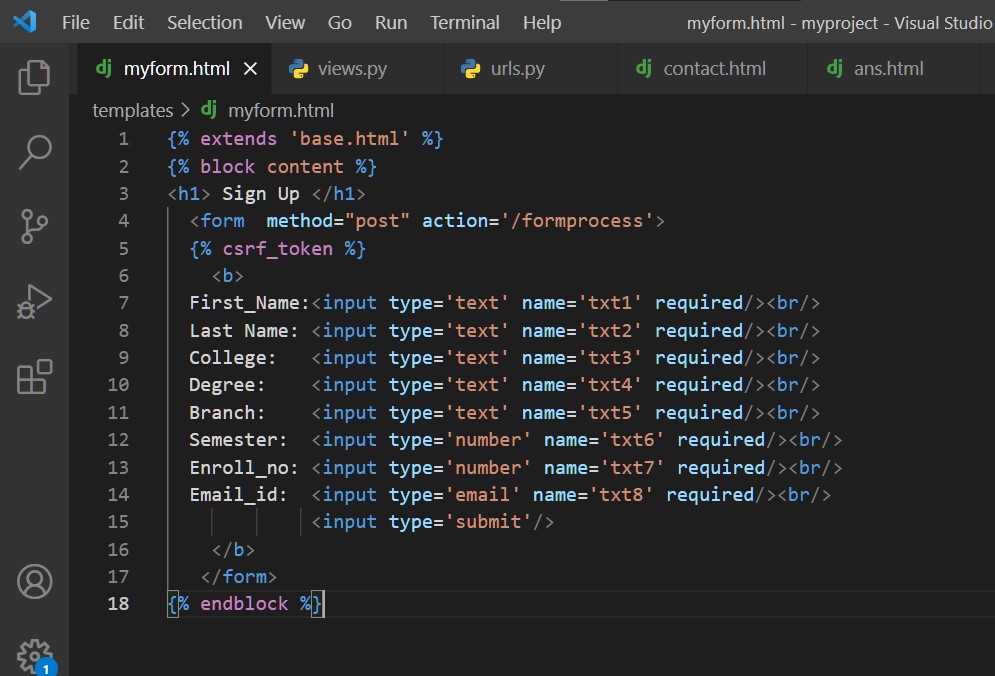
So i had take input to the Sign Up Form from the user using ‘POST’ method.

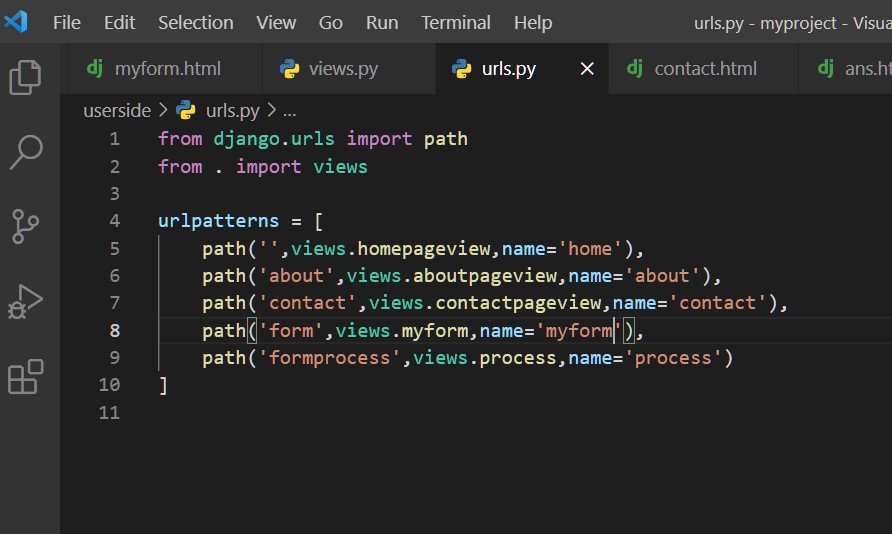
For the input purpose, I had created one HTML file i.e

ans.html.

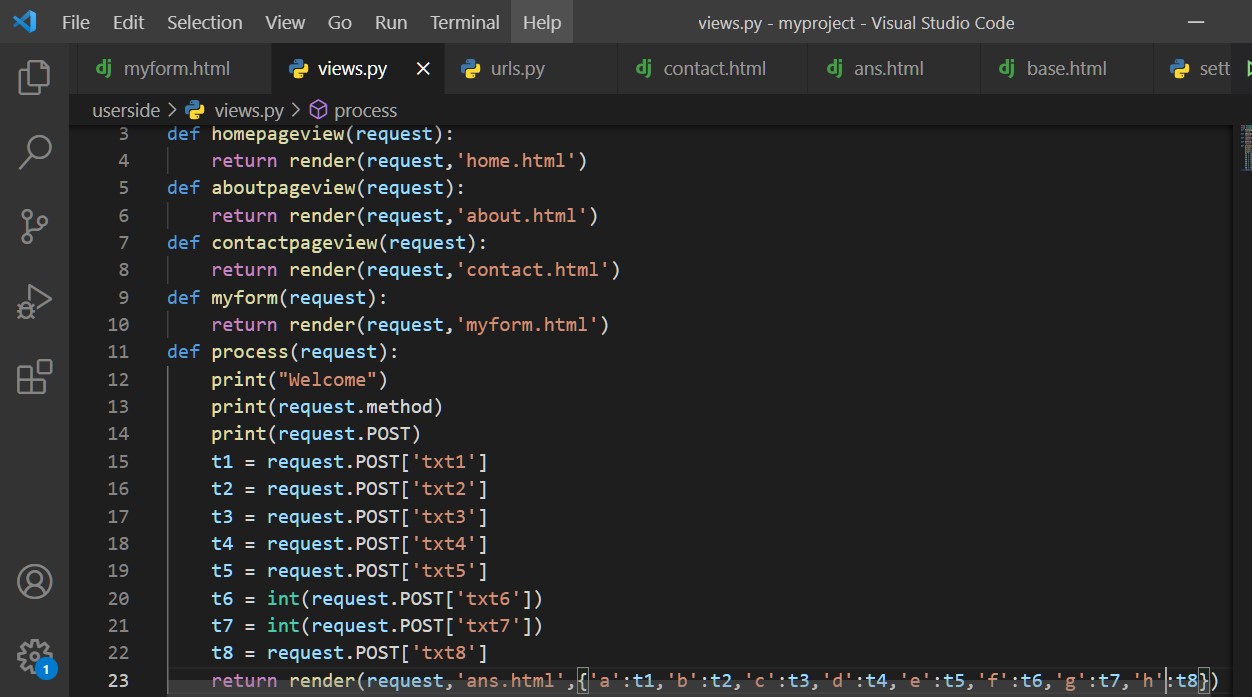
And here to apply the ‘Stampalike theme’ I had extends base.html in ans.html, and in myform.html.

**Myfrom**

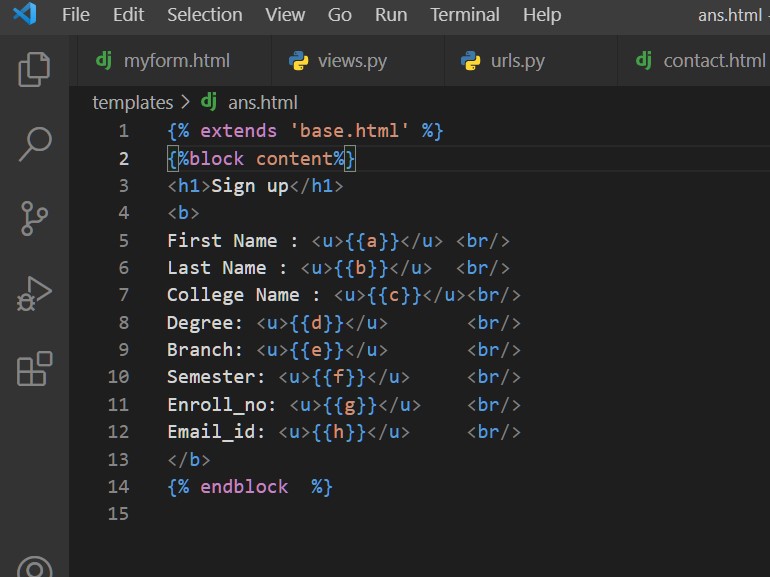


**Setting myform url path**

**Rendering the values in views.py**

****

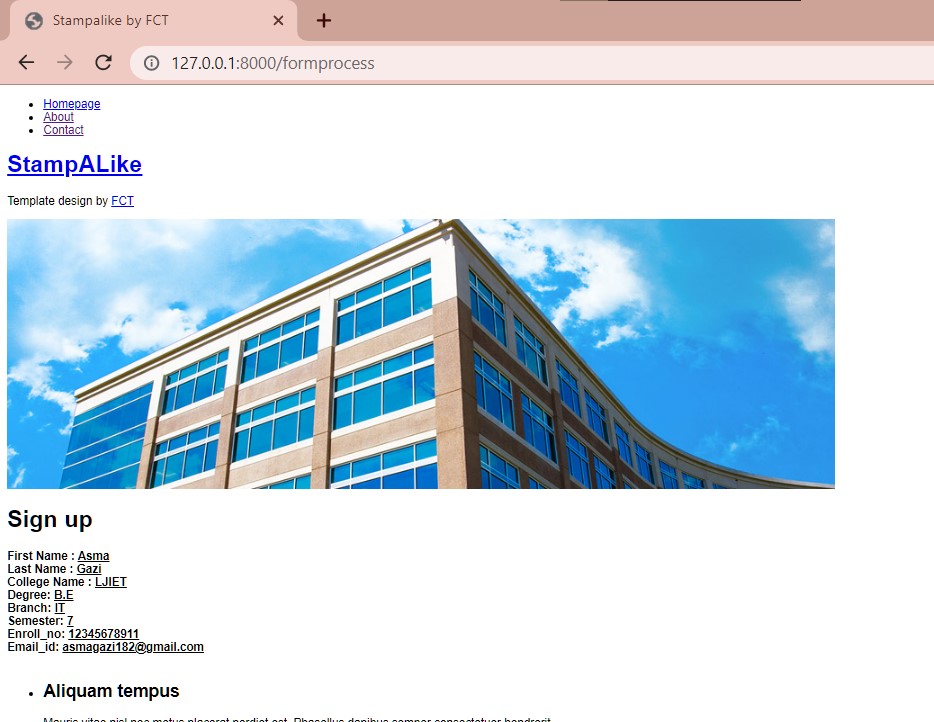
**Ans.html**

****

**SignUp Form /Form**

****

**SignUp Form/Formprocess**

****

* **Django Database**

So, here first of all I had created two models Student and Employee in models.py which contain various datafields and register this in admin.py

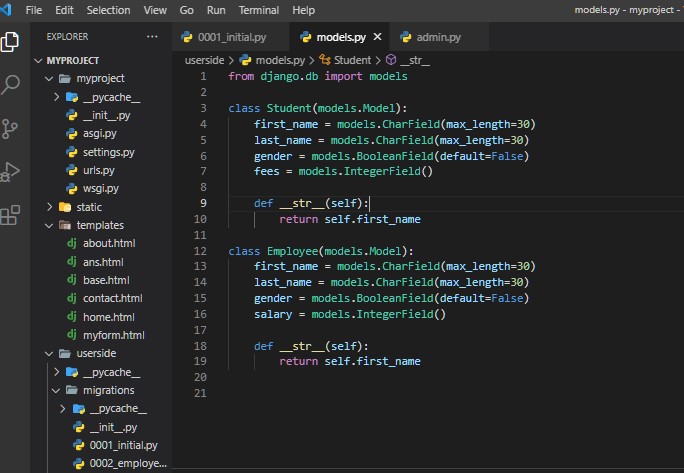
After that I had run this two below command so that I can set these two above models in my Django Administration

“python manage.py makemigrations”

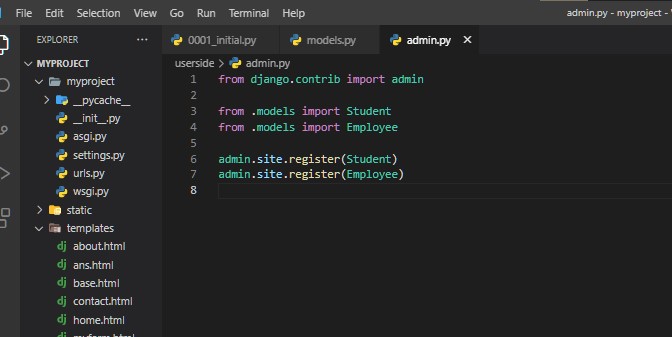
“python manage.py migrate”

And in these models we can add the input to the datafields

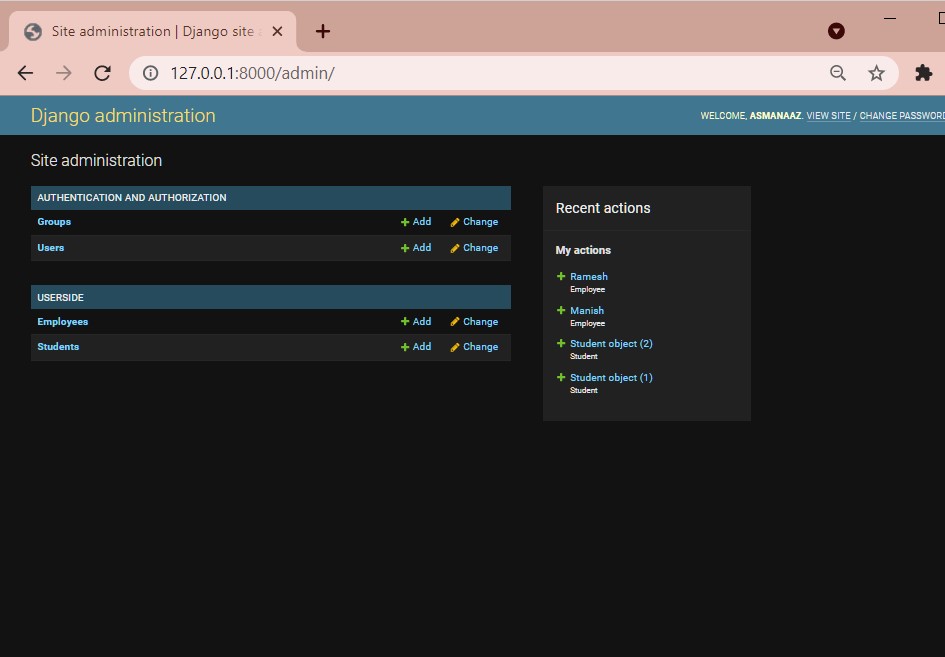
**Student and Employee Class**

****

**admin.py**

****

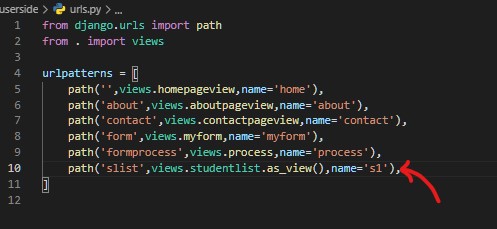
**Student and Employee**

****

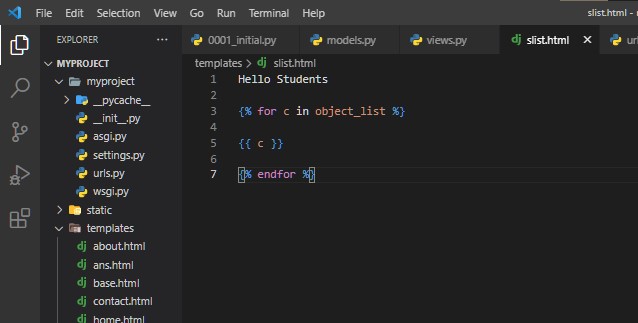
Below image show one class name as a studentlist in views.py that I had created which load the data in the userside.

Here the data is fetched using ListView,and I had stored that data in slist in templates folder.

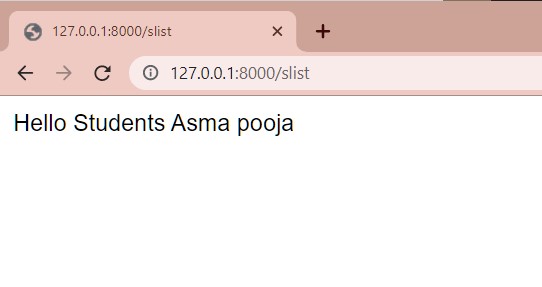
Then I had set the path in urls.py



**slist\_input**

****

**slist\_output**

****

**---------------------------- End ----------------------------**