```
Cd Documents
Mkdir CSE
Cd CSE

Python --version
Python3 --version
Pip3 --version
Pip3 install django==2.0.3

django-admin --version

django-admin startproject myproject
Ls
Myproject
Cd Myproject

Python3  manage.py runserver
127.0.0.1:8000 ---> check in your browser
Or
Localhost:8000

Ctrl+c
Documents/CSE/myproject/…….

python3 manage.py startapp myapp
Ls
Cd myapp
Ls
Cd ..

Python3 manage.py runserver

Pkill -9 python

Python3 manage.py runserver
```

----

manage.py: It is a command-line utility which allows us **to interact with the project** in various ways and also used to manage an application that we will see later on in this tutorial.

__init__.py: It is an empty file that tells to the Python that this **directory should be considered as a Python package**.

settings.py: This file is used to configure application settings such as **database connection, static files linking etc**.

urls.py: This file contains the listed URLs of the application. In this file, we can mention the URLs and corresponding actions to perform the task and display the view.

wsgi.py: It is an entry-point for WSGI-compatible web servers to serve Django project.

Views.py

myapp/views.py

```
from django.shortcuts import render

from django.http import HttpResponse

def hello(request):

    return HttpResponse("<h2>Hello World!!! </h2>")
```

myproject/urls.py

```
from django.contrib import admin

from django.urls import path

from myapp import views

urlpatterns = [

path('admin/', admin.site.urls),

 path('hello/', views.hello),

]
```

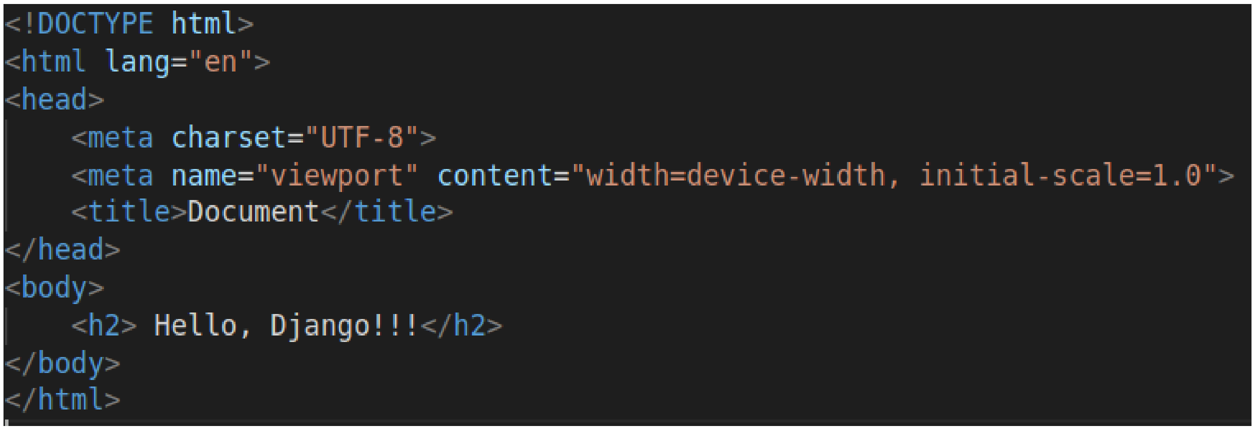
=>127.0.0.1:8000/hello

Ctrl+~   --------------------

Template(html files)

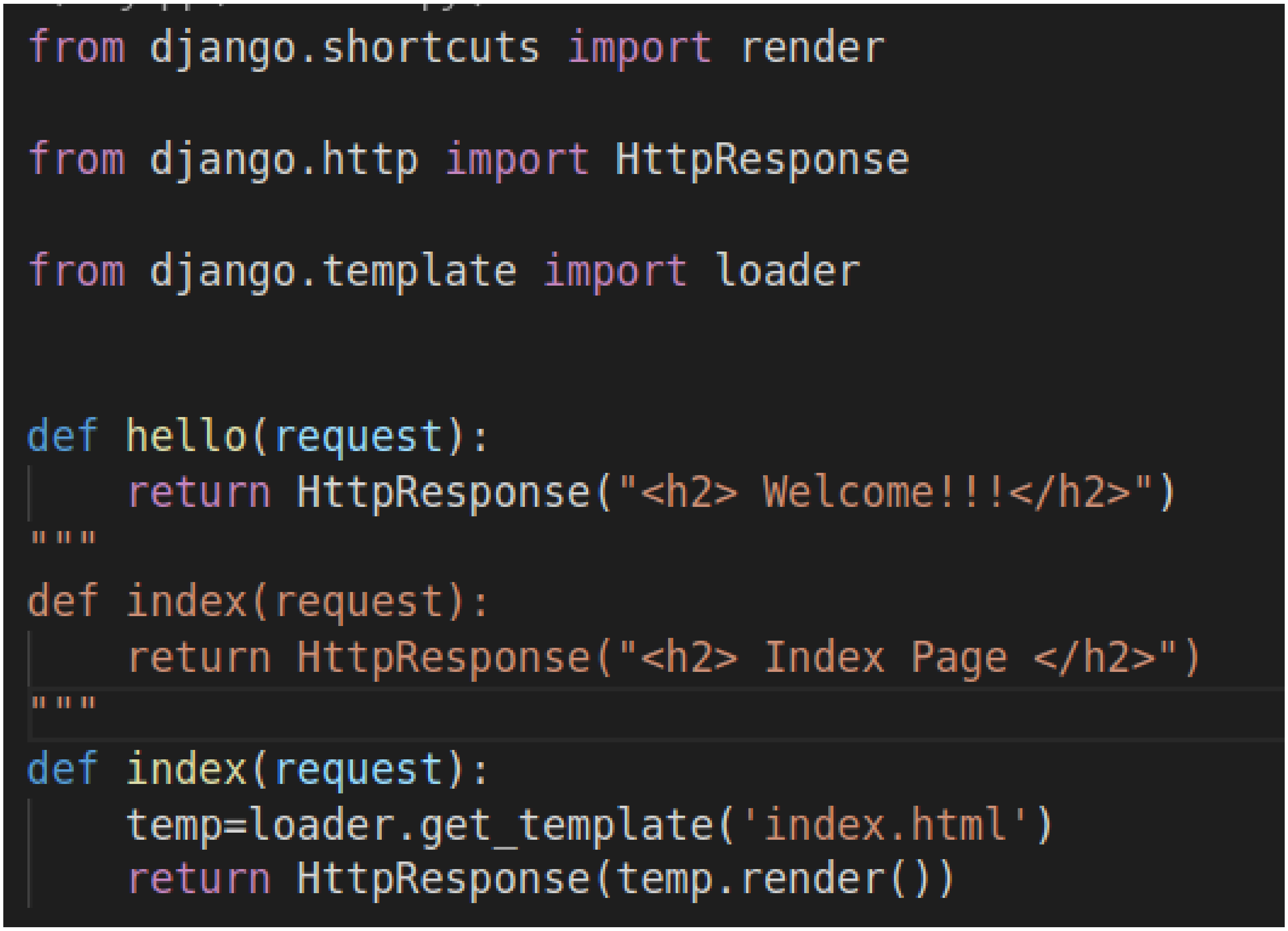
Create a new directory myapp/templates/index.html

Index.html :-



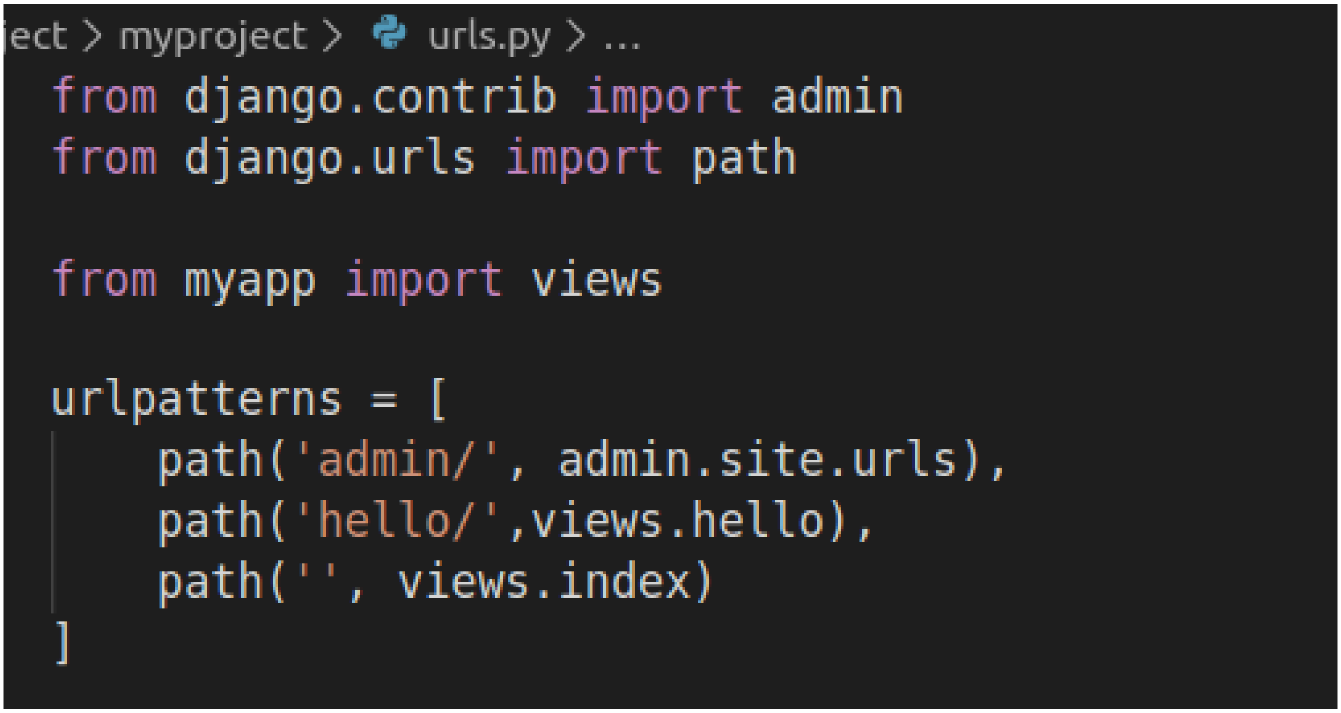
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h2> Hello, Django!!!</h2>
</body>
</html>
```

Views.py:-



```python
from django.shortcuts import render

from django.http import HttpResponse

from django.template import loader


def hello(request):
    return HttpResponse("<h2> Welcome!!!</h2>")
"""
def index(request):
    return HttpResponse("<h2> Index Page </h2>")
"""
def index(request):
    temp=loader.get_template('index.html')
    return HttpResponse(temp.render())
```

Urls.py:-

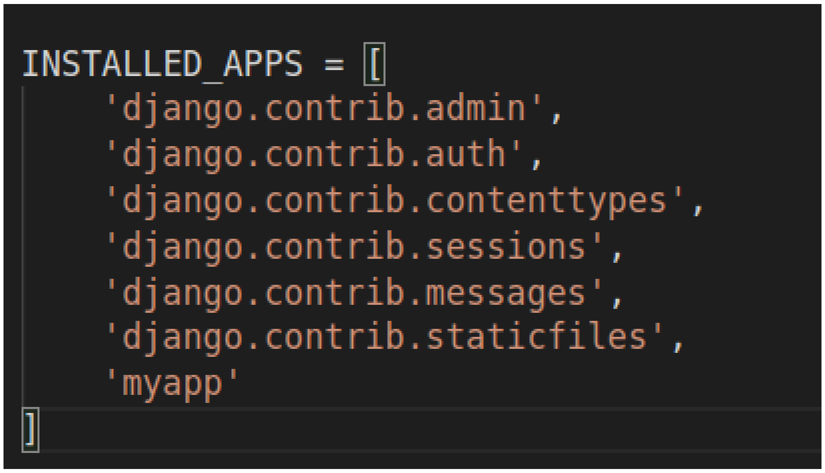


ject > myproject > urls.py > ...

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

from myapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('hello/',views.hello),
    path('', views.index)
]
```

Settings.py:-

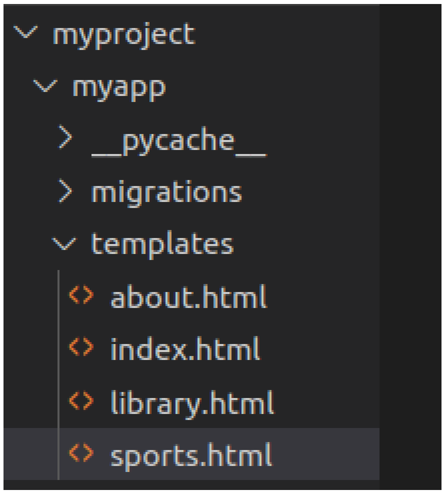
Include your app name settings.py



```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp'
]
```

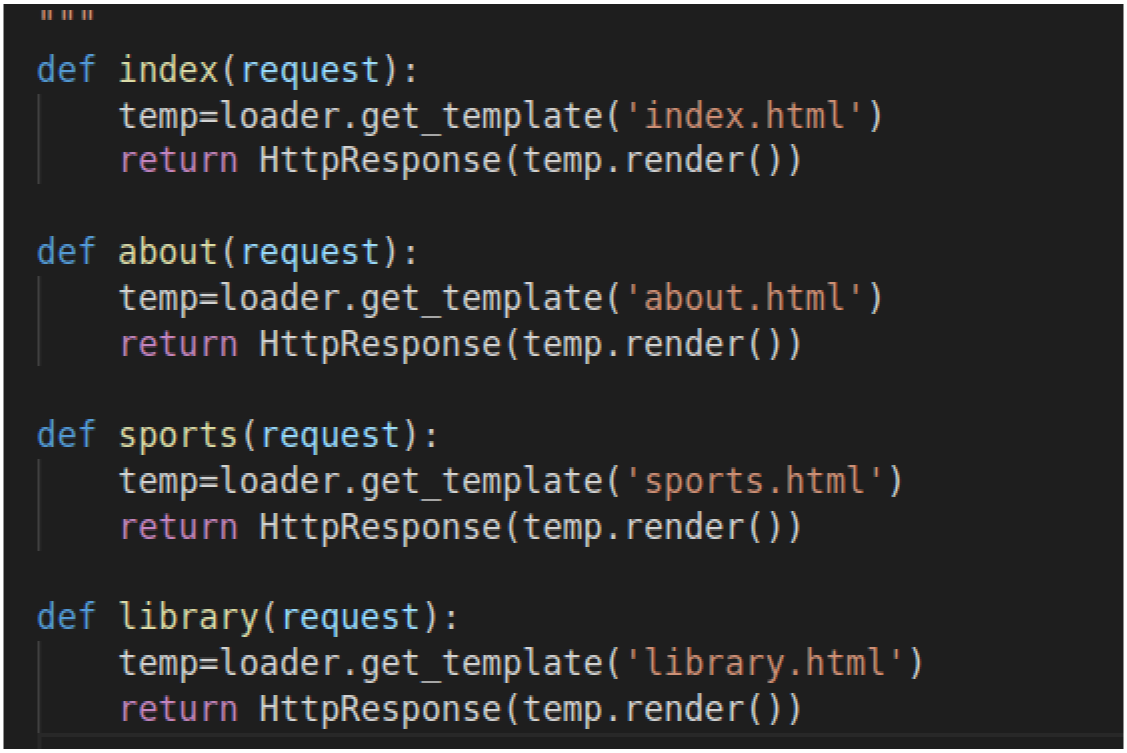

Routings:-

Create new files inside the templates and design all the pages…

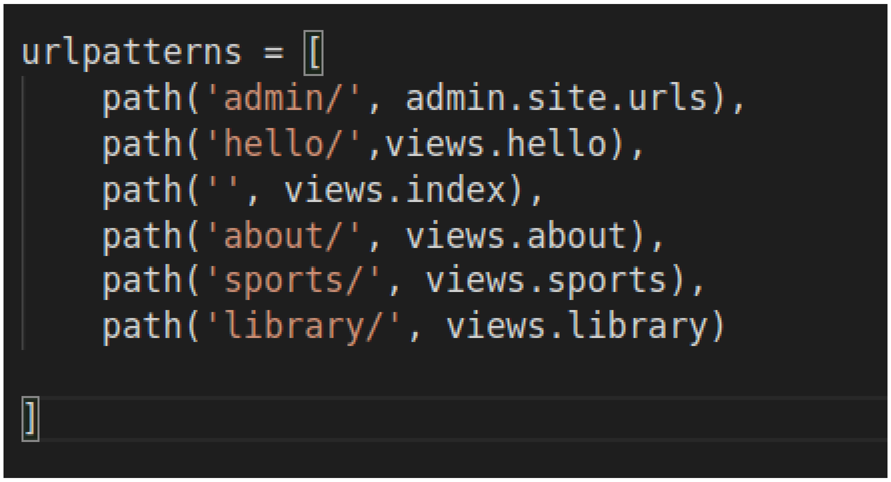


```
myproject
  myapp
    __pycache__
    migrations
    templates
      about.html
      index.html
      library.html
      sports.html
```

Views.py

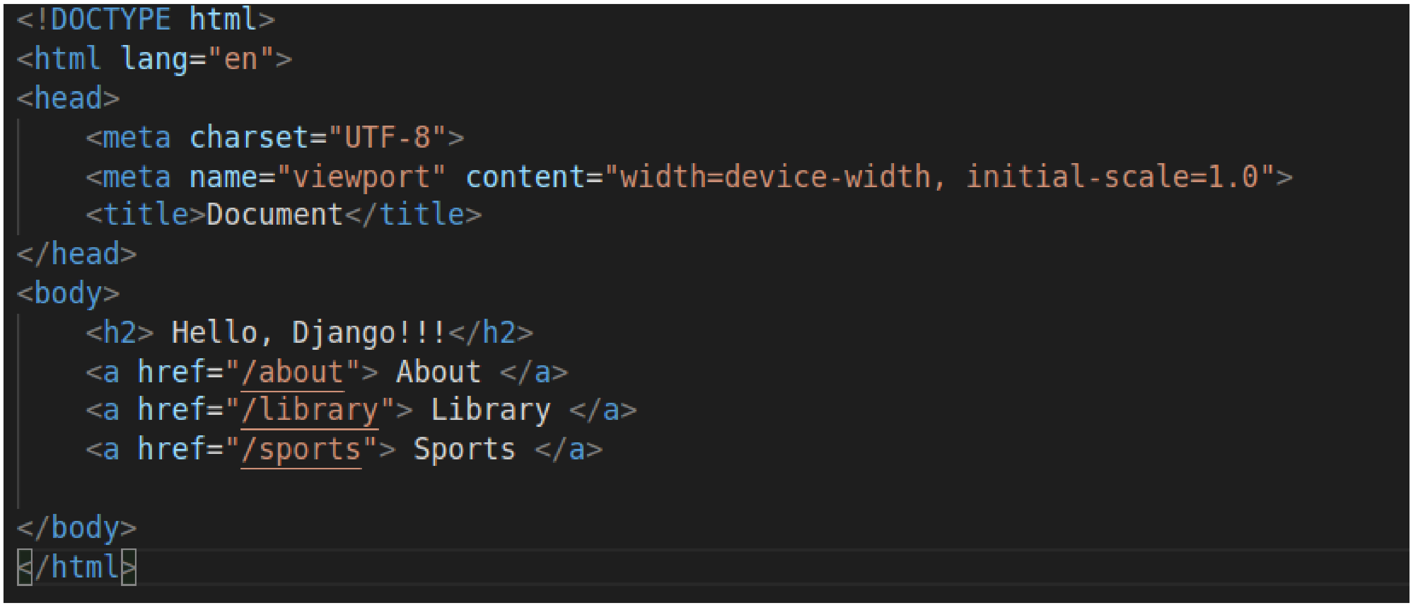


```python
"""
def index(request):
    temp=loader.get_template('index.html')
    return HttpResponse(temp.render())

def about(request):
    temp=loader.get_template('about.html')
    return HttpResponse(temp.render())

def sports(request):
    temp=loader.get_template('sports.html')
    return HttpResponse(temp.render())

def library(request):
    temp=loader.get_template('library.html')
    return HttpResponse(temp.render())
```



```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('hello/',views.hello),
    path('', views.index),
    path('about/', views.about),
    path('sports/', views.sports),
    path('library/', views.library)

]
```

Index.html:-

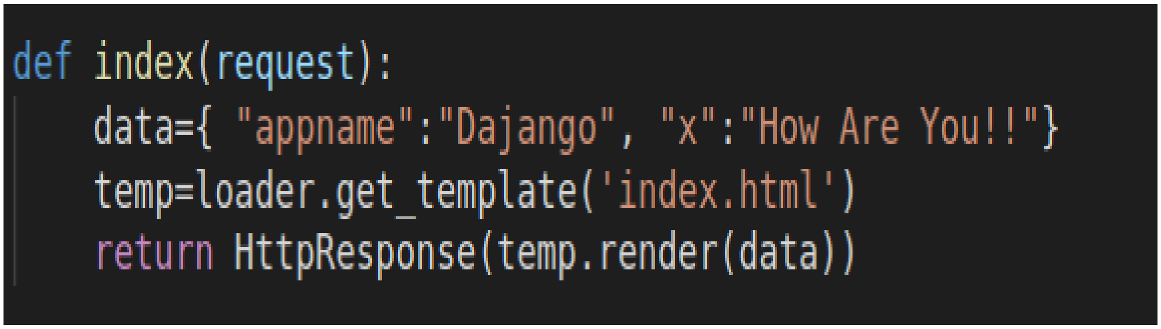


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h2> Hello, Django!!!</h2>
    <a href="/about"> About </a>
    <a href="/library"> Library </a>
    <a href="/sports"> Sports </a>

</body>
</html>
```

URL REWRITINGS

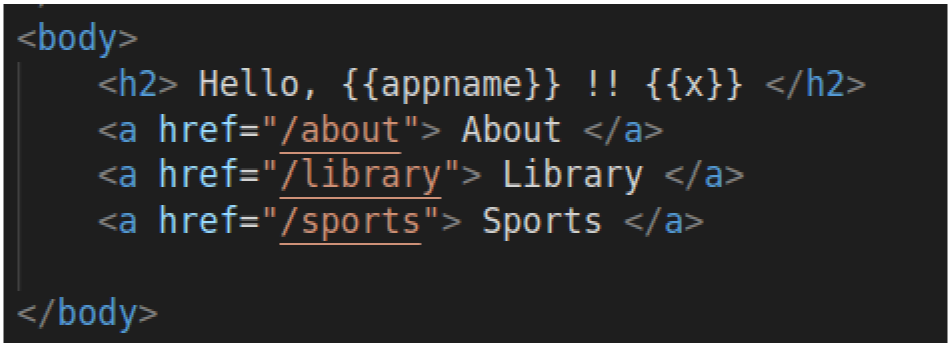
Variables:

Views.py



```python
def index(request):
    data={ "appname":"Dajango", "x":"How Are You!!"}
    temp=loader.get_template('index.html')
    return HttpResponse(temp.render(data))
```

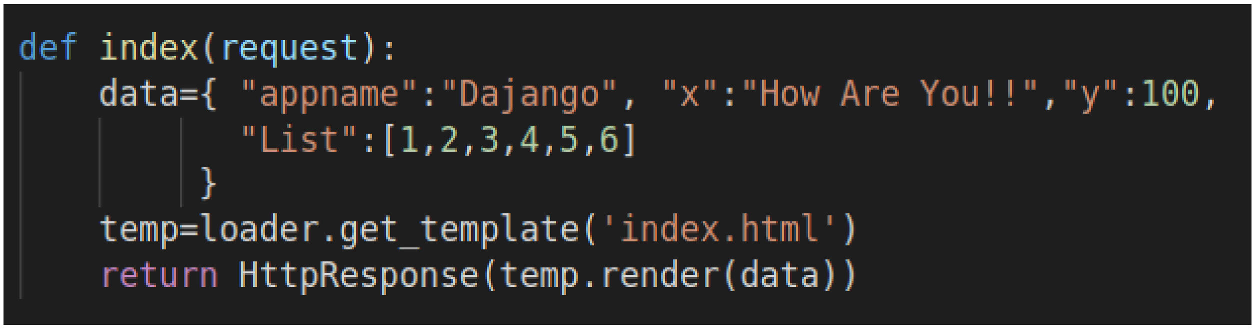
Index.html



```html
<body>
    <h2> Hello, {{appname}} !! {{x}} </h2>
    <a href="/about"> About </a>
    <a href="/library"> Library </a>
    <a href="/sports"> Sports </a>

</body>
```

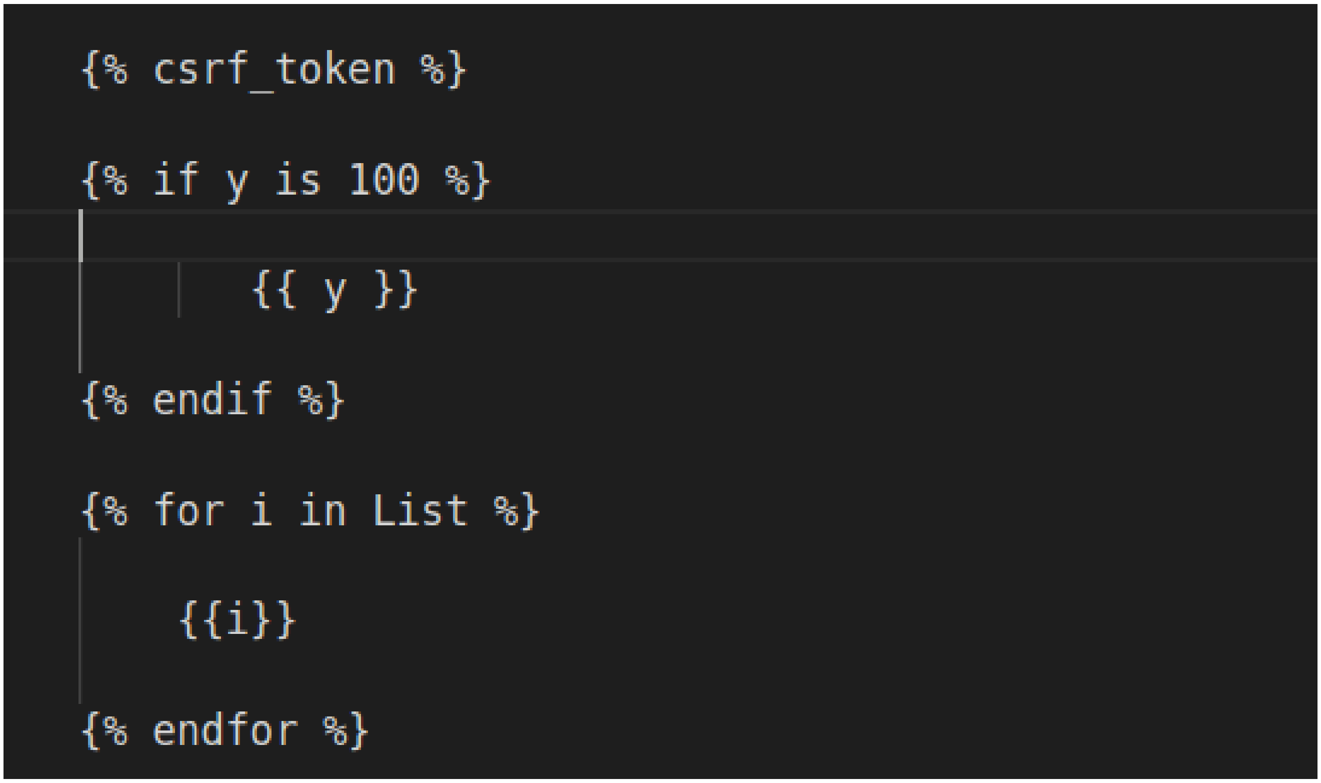
Using Tags:-

Views.py



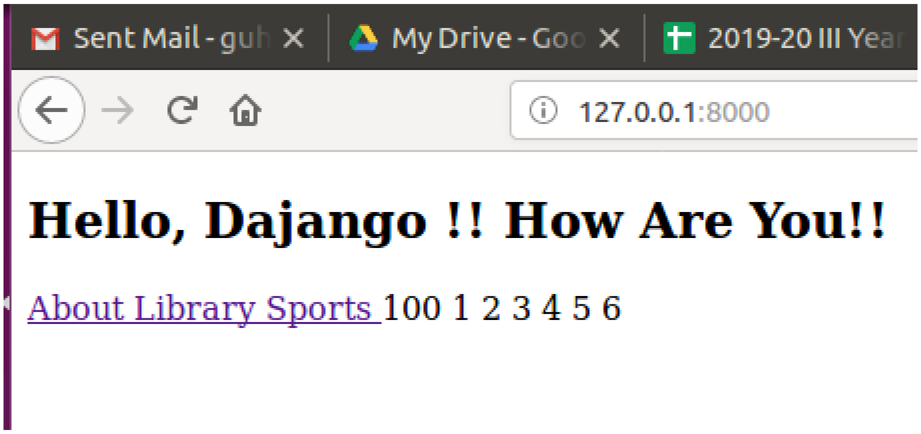
```python
def index(request):
    data={ "appname":"Dajango", "x":"How Are You!!","y":100,
            "List":[1,2,3,4,5,6]
          }
    temp=loader.get_template('index.html')
    return HttpResponse(temp.render(data))
```

Index.html



```
{% csrf_token %}

{% if y is 100 %}

        {{ y }}

{% endif %}

{% for i in List %}

    {{i}}

{% endfor %}
```

Output

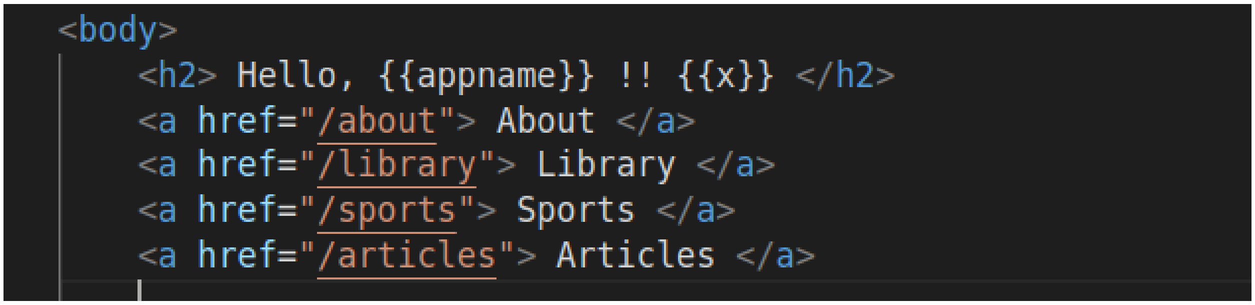


Hello, Dajango !! How Are You!!

About Library Sports 100 1 2 3 4 5 6

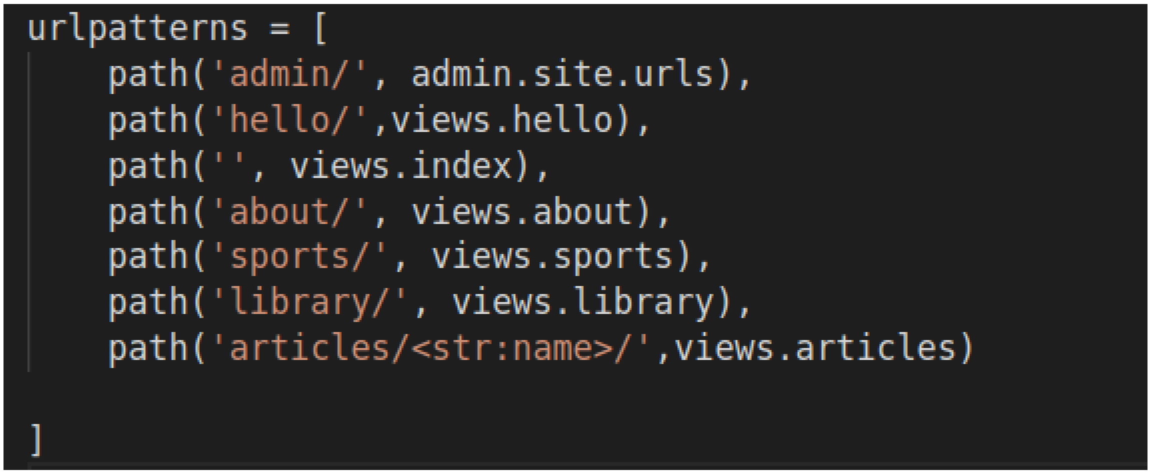
Url Mapping

Index.html



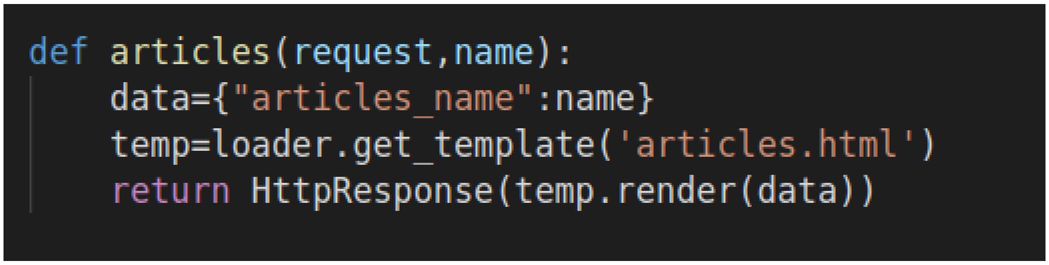
```
<body>
    <h2> Hello, {{appname}} !! {{x}} </h2>
    <a href="/about"> About </a>
    <a href="/library"> Library </a>
    <a href="/sports"> Sports </a>
    <a href="/articles"> Articles </a>
```

Urls.py



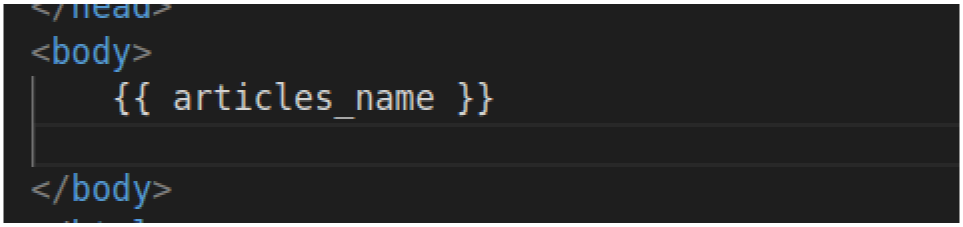
```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('hello/',views.hello),
    path('', views.index),
    path('about/', views.about),
    path('sports/', views.sports),
    path('library/', views.library),
    path('articles/<str:name>/',views.articles)

]
```

Views.py:- **127.0.0.1:8000/articles/django**



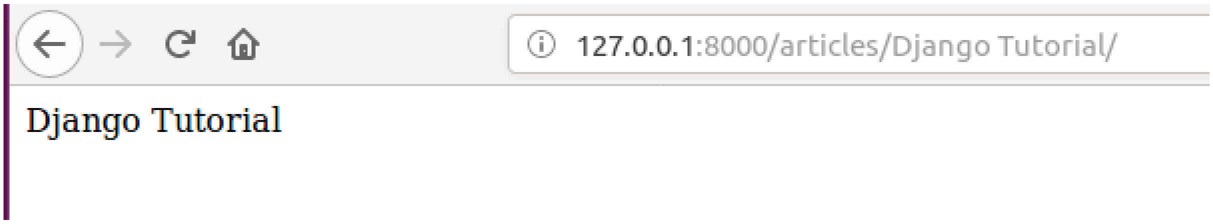
```
def articles(request,name):
    data={"articles_name":name}
    temp=loader.get_template('articles.html')
    return HttpResponse(temp.render(data))
```

Articles.html



```
</head>
<body>
    {{ articles_name }}

</body>
```

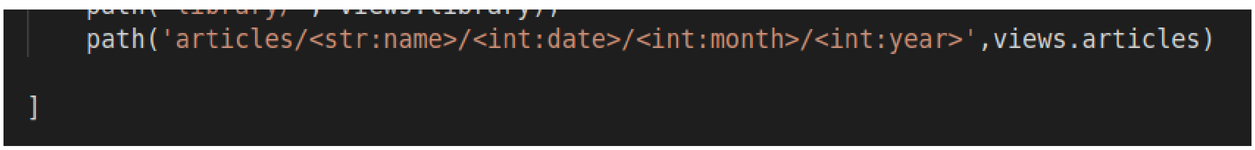
Result



Django Tutorial

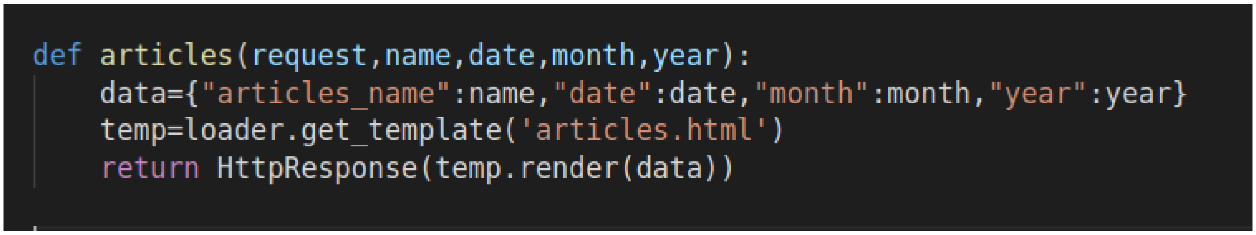
More…..

Urls.py



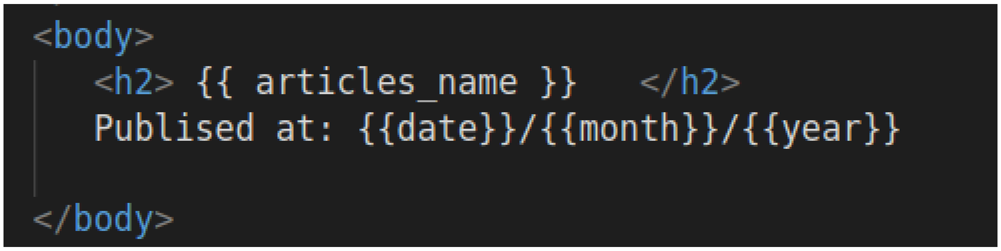
```
    path('library/',views.library),
    path('articles/<str:name>/<int:date>/<int:month>/<int:year>',views.articles)

]
```

Views.py



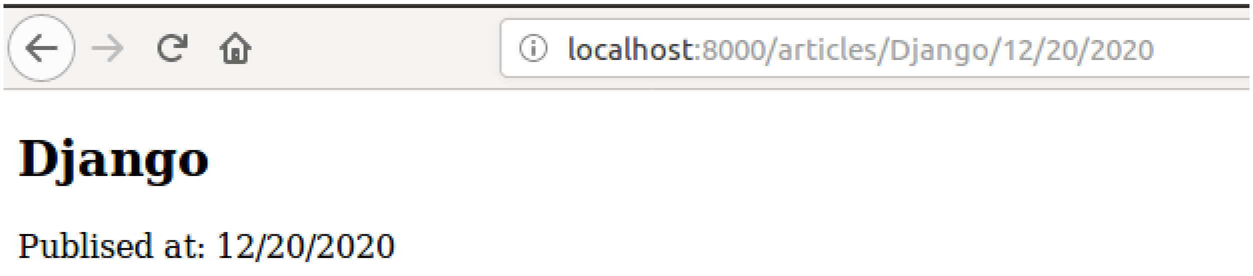
```
def articles(request,name,date,month,year):
    data={"articles_name":name,"date":date,"month":month,"year":year}
    temp=loader.get_template('articles.html')
    return HttpResponse(temp.render(data))
```

Articles.html



```
<body>
   <h2> {{ articles_name }}   </h2>
   Publised at: {{date}}/{{month}}/{{year}}

</body>
```

Result:-


localhost:8000/articles/Django/12/20/2020

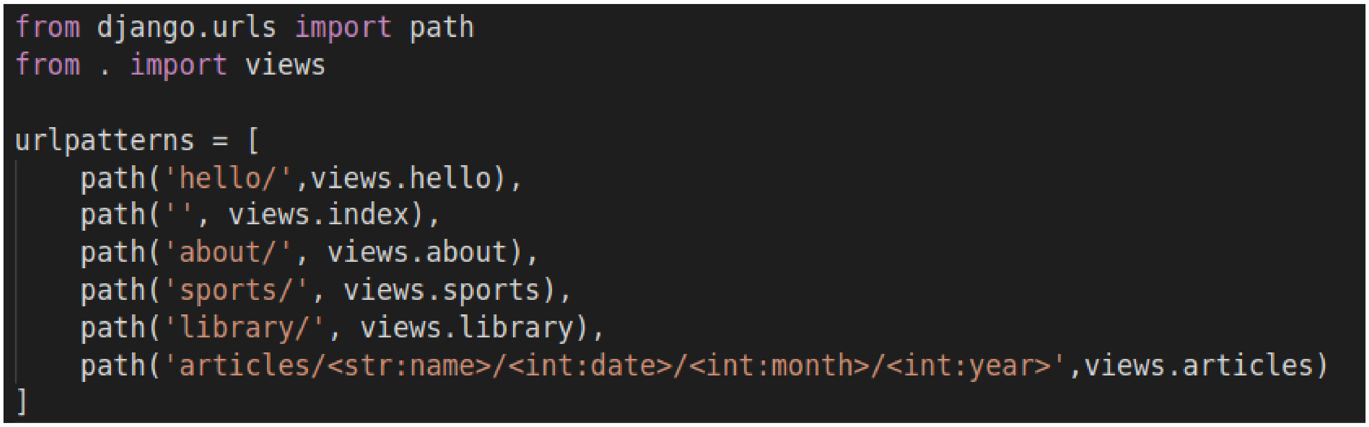
Django

Publised at: 12/20/2020

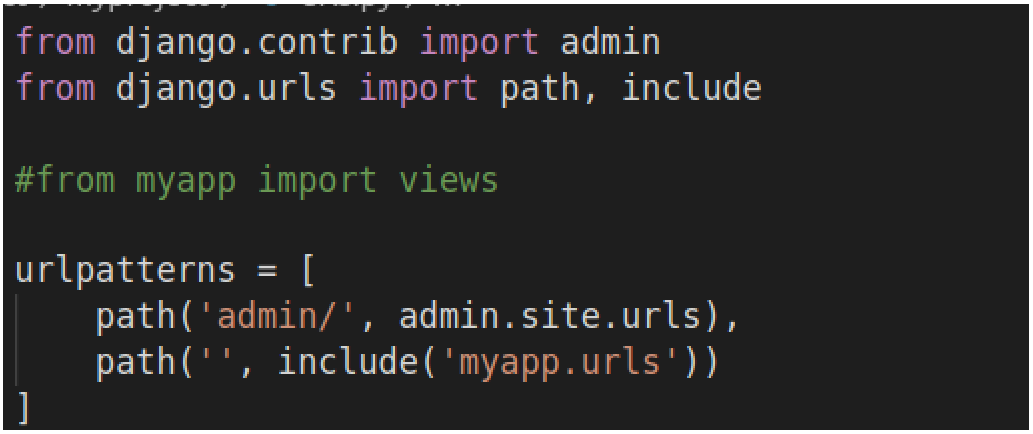
Include and path:

Create urls.py inside the myapp

Change the code like below mentioned


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

urlpatterns = [
    path('hello/',views.hello),
    path('', views.index),
    path('about/', views.about),
    path('sports/', views.sports),
    path('library/', views.library),
    path('articles/<str:name>/<int:date>/<int:month>/<int:year>',views.articles)
]
```

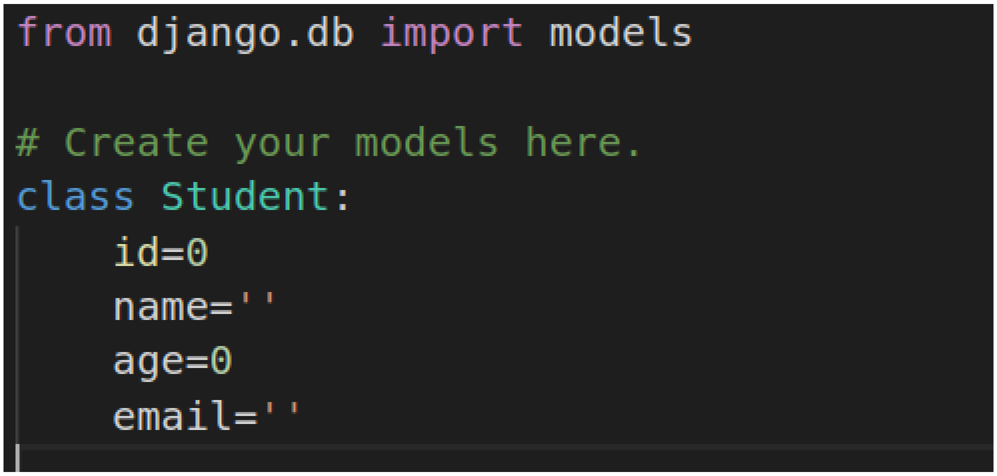
myproject/urls.py :-



```
from django.contrib import admin
from django.urls import path, include

#from myapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('myapp.urls'))
]
```

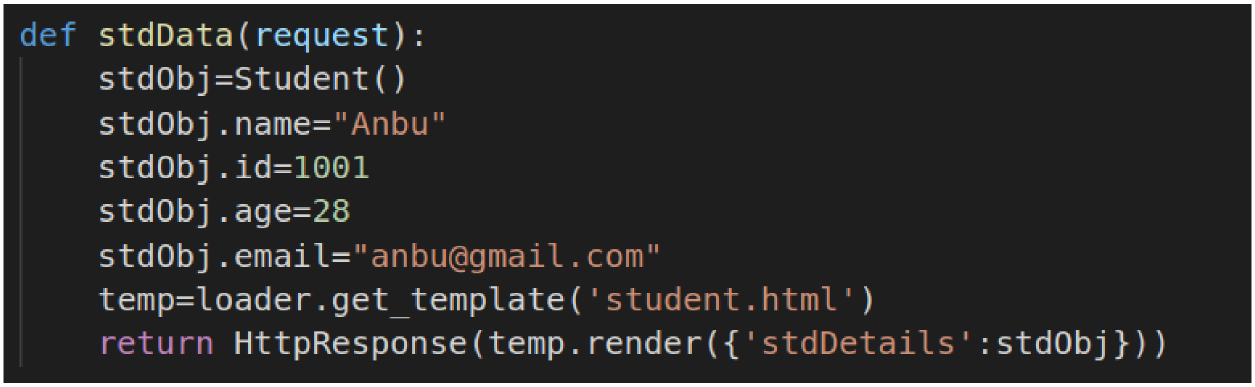
----

myapp/models.py



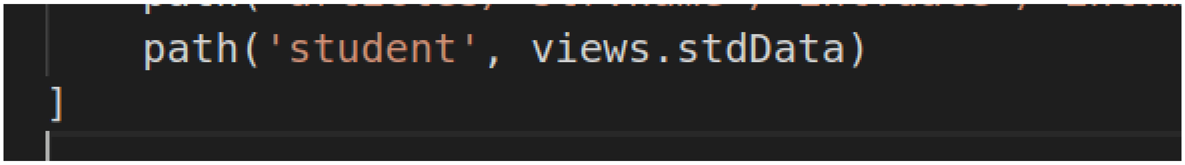
```
from django.db import models

# Create your models here.
class Student:
    id=0
    name=''
    age=0
    email=''
```

Views.py



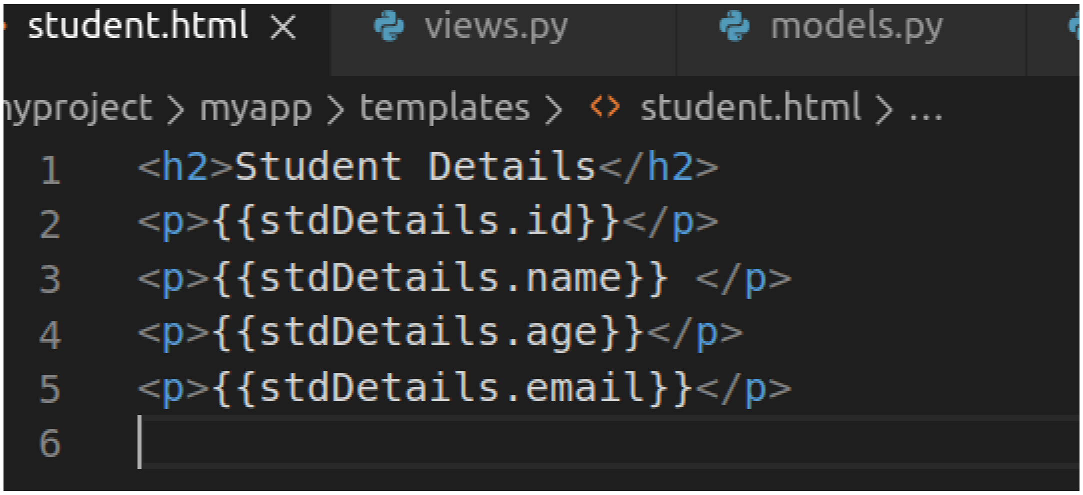
```
def stdData(request):
    stdObj=Student()
    stdObj.name="Anbu"
    stdObj.id=1001
    stdObj.age=28
    stdObj.email="anbu@gmail.com"
    temp=loader.get_template('student.html')
    return HttpResponse(temp.render({'stdDetails':stdObj}))
```

Urls.py



```
    path('student', views.stdData)
]
```

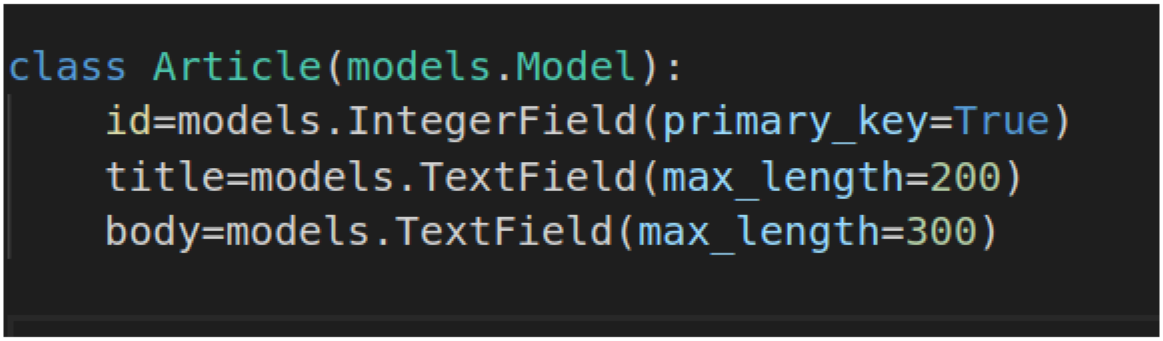
Student.html



```
<h2>Student Details</h2>
<p>{{stdDetails.id}}</p>
<p>{{stdDetails.name}} </p>
<p>{{stdDetails.age}}</p>
<p>{{stdDetails.email}}</p>
```

Migrations:-

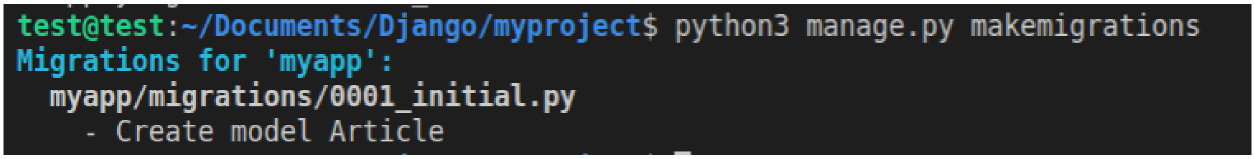
myapp/models.py



```
class Article(models.Model):
    id=models.IntegerField(primary_key=True)
    title=models.TextField(max_length=200)
    body=models.TextField(max_length=300)
```

Cmd: python3 manage.py migrate

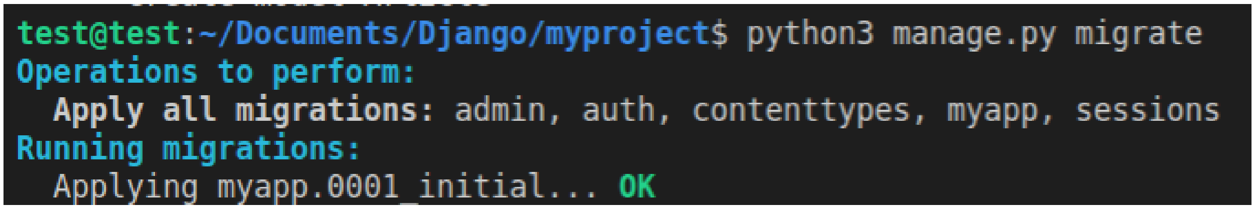
And



```
test@test:~/Documents/Django/myproject$ python3 manage.py makemigrations
Migrations for 'myapp':
  myapp/migrations/0001_initial.py
    - Create model Article
```

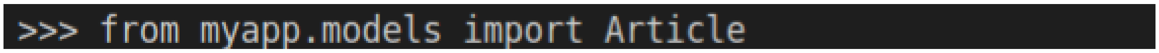
Then go and check migrations/0001_initials.py

Run again to migrate all…

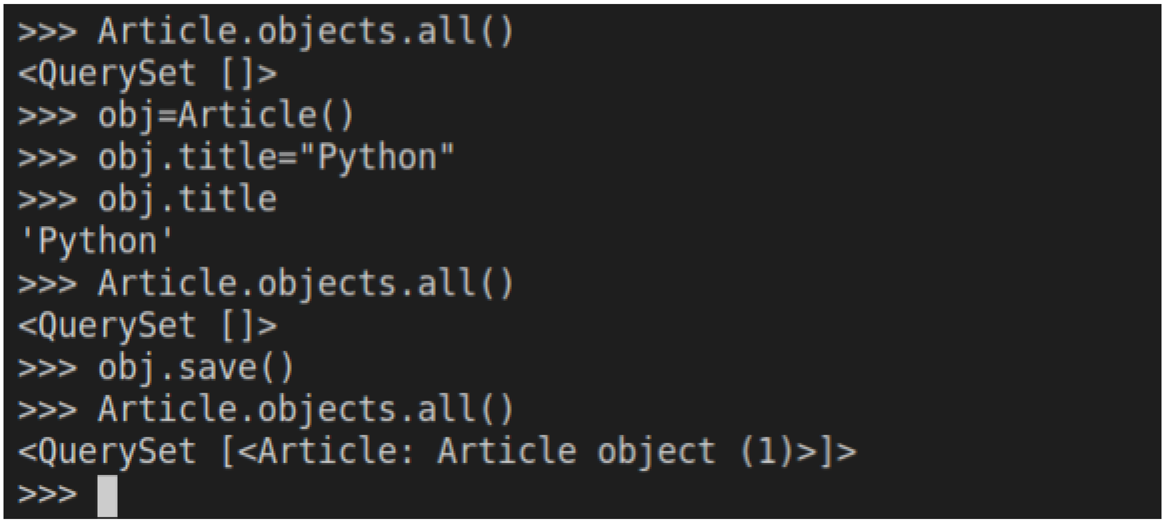


```
test@test:~/Documents/Django/myproject$ python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, myapp, sessions
Running migrations:
  Applying myapp.0001_initial... OK
```

Open python shell → python3 manage.py shell

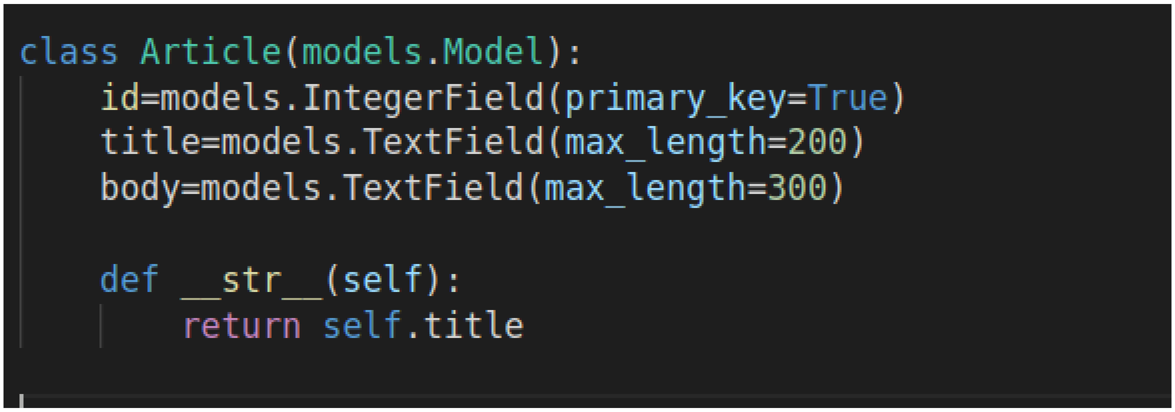


```
>>> from myapp.models import Article
```



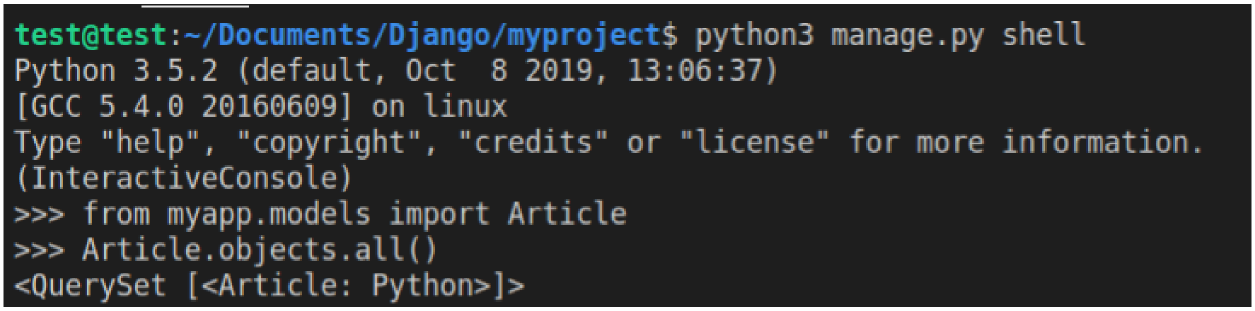
```
>>> Article.objects.all()
<QuerySet []>
>>> obj=Article()
>>> obj.title="Python"
>>> obj.title
'Python'
>>> Article.objects.all()
<QuerySet []>
>>> obj.save()
>>> Article.objects.all()
<QuerySet [<Article: Article object (1)>]>
>>>
```

myapp/models.py



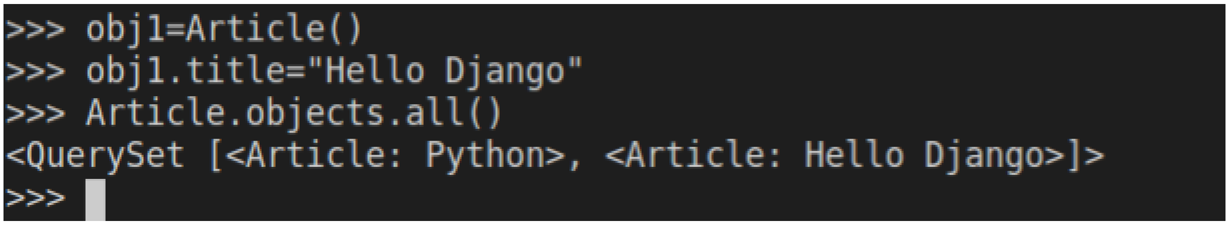
```
class Article(models.Model):
    id=models.IntegerField(primary_key=True)
    title=models.TextField(max_length=200)
    body=models.TextField(max_length=300)

    def __str__(self):
        return self.title
```

Open Shell



```
test@test:~/Documents/Django/myproject$ python3 manage.py shell
Python 3.5.2 (default, Oct  8 2019, 13:06:37)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from myapp.models import Article
>>> Article.objects.all()
<QuerySet [<Article: Python>]>
```

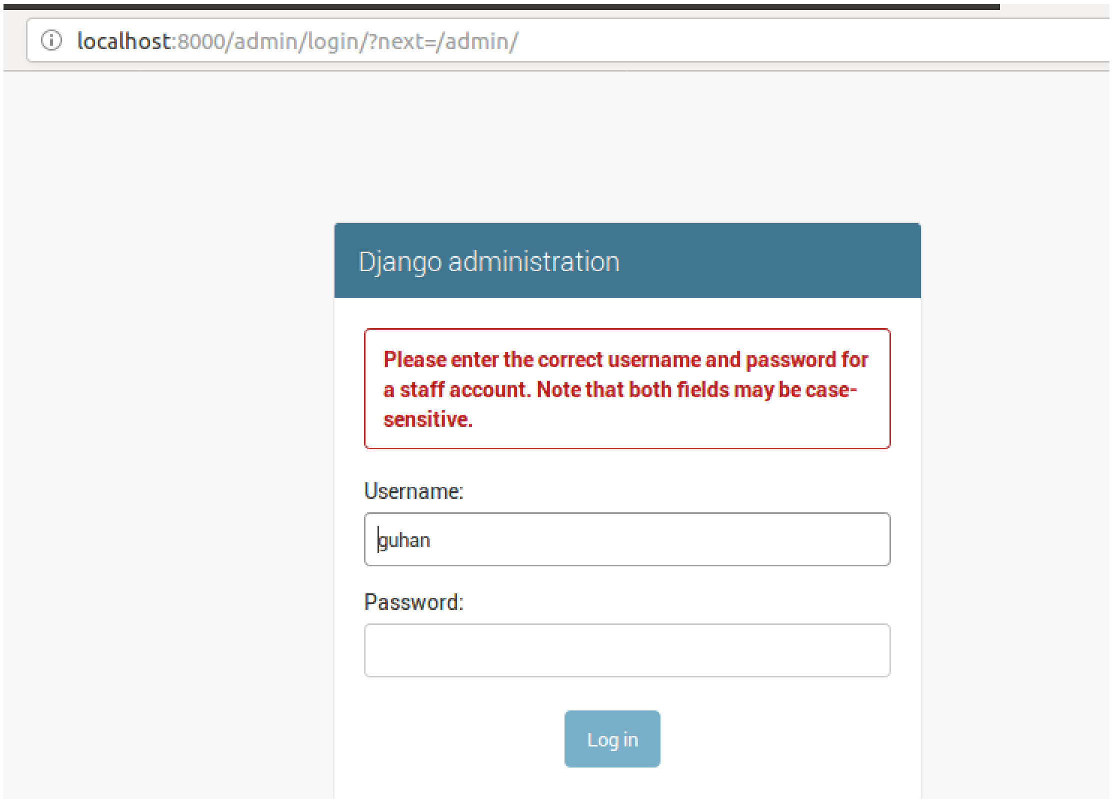
And next object



```
>>> obj1=Article()
>>> obj1.title="Hello Django"
>>> Article.objects.all()
<QuerySet [<Article: Python>, <Article: Hello Django>]>
>>> 
```

Django Admin

Cmd: Python3 manage.py runserver



localhost:8000/admin/login/?next=/admin/

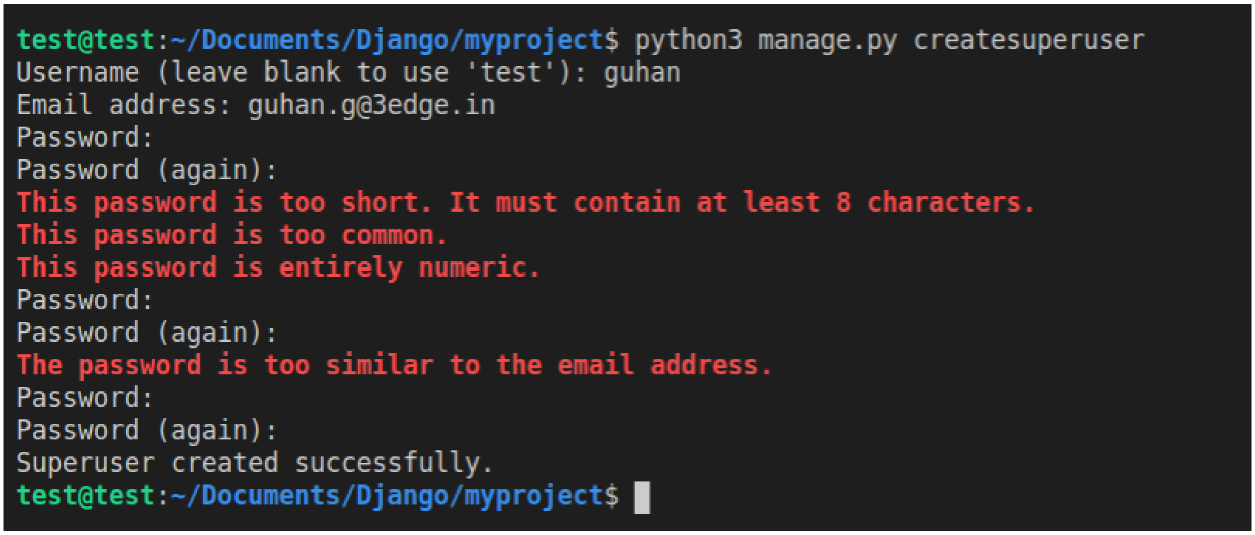
Django administration

Please enter the correct username and password for a staff account. Note that both fields may be case-sensitive.

Username: guhan

Password:

Create login credentials

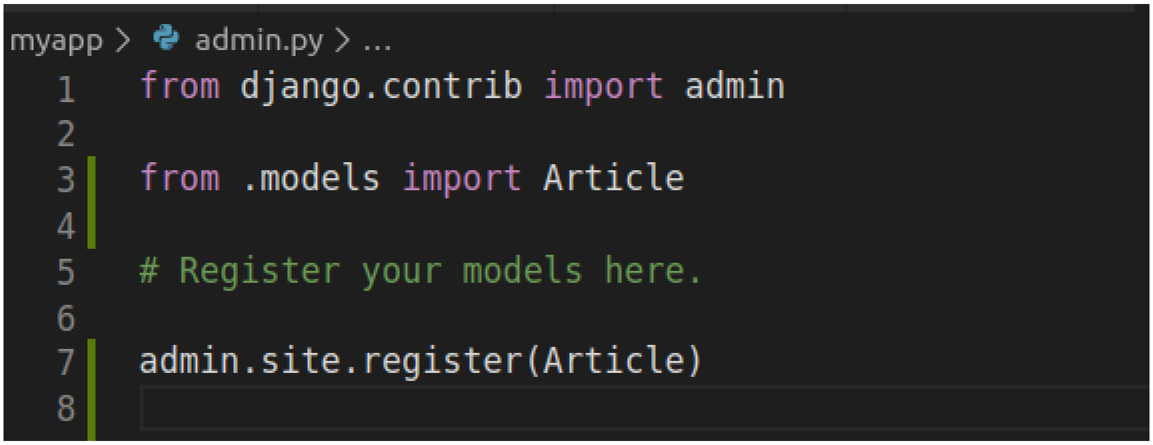


```
test@test:~/Documents/Django/myproject$ python3 manage.py createsuperuser
Username (leave blank to use 'test'): guhan
Email address: guhan.g@3edge.in
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Password:
Password (again):
The password is too similar to the email address.
Password:
Password (again):
Superuser created successfully.
test@test:~/Documents/Django/myproject$
```

Username:guhan

Password:Anbu@123

myapp/admin.py:-

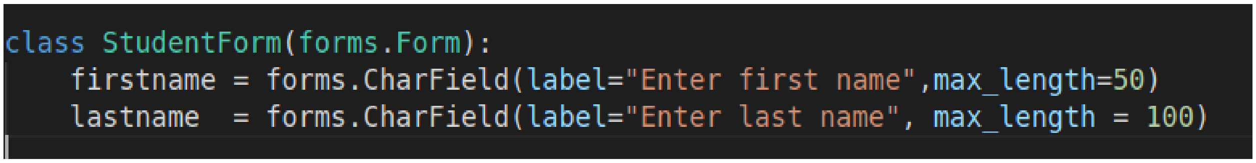


```
from django.contrib import admin

from .models import Article

# Register your models here.

admin.site.register(Article)

```

Go and refresh your web page to edit your articles…

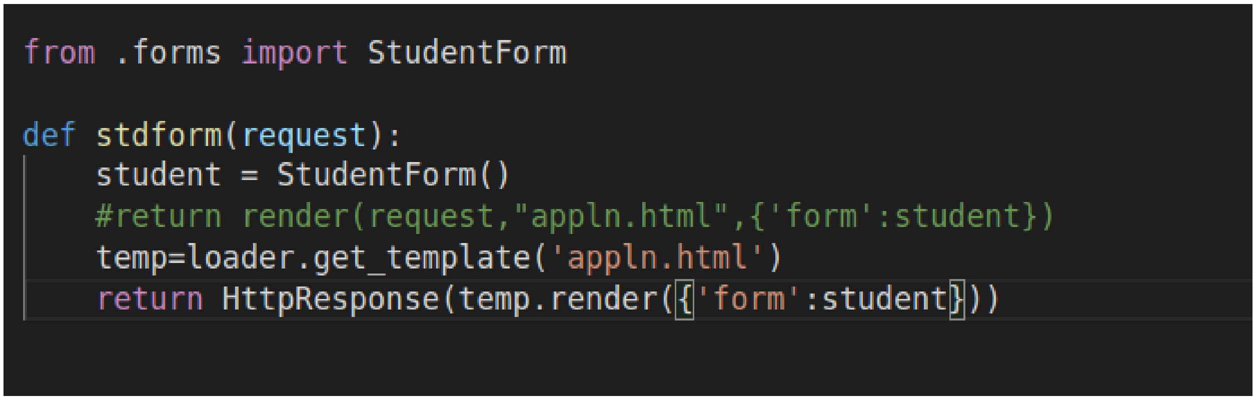
Django Forms:-

Forms.py:-



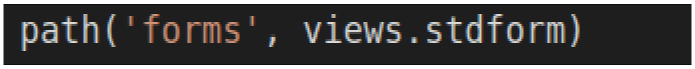
```
class StudentForm(forms.Form):
    firstname = forms.CharField(label="Enter first name",max_length=50)
    lastname  = forms.CharField(label="Enter last name", max_length = 100)
```

Views.py:-



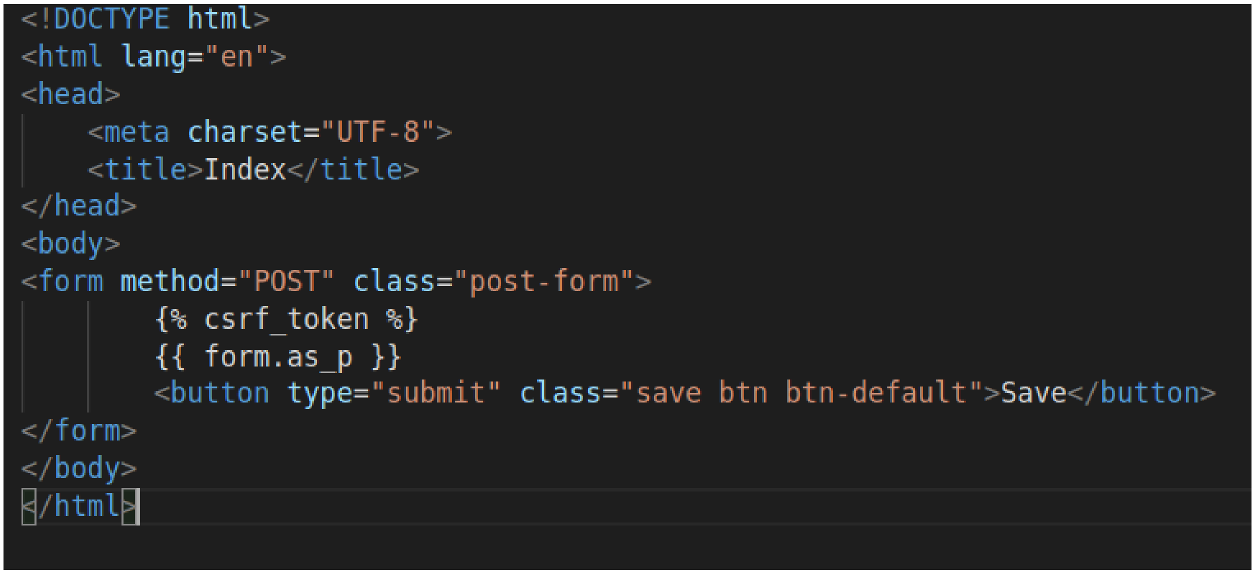
```
from .forms import StudentForm

def stdform(request):
    student = StudentForm()
    #return render(request,"appln.html",{'form':student})
    temp=loader.get_template('appln.html')
    return HttpResponse(temp.render({'form':student}))
```

Urls.py:-



```
path('forms', views.stdform)
```

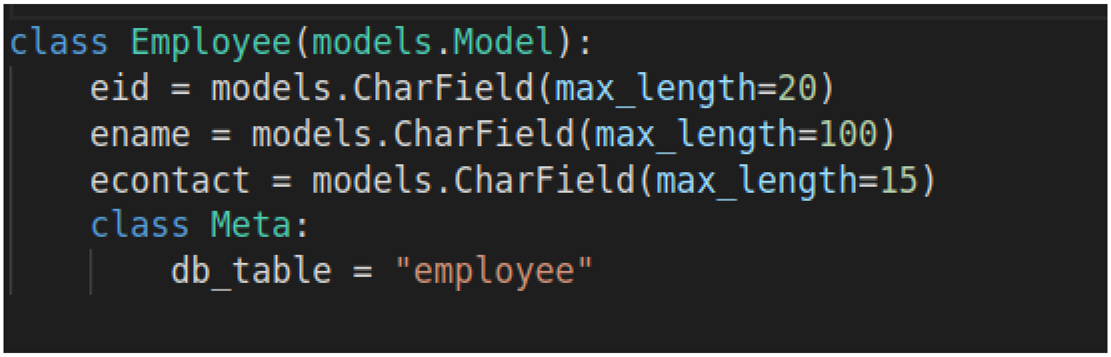
Appln.html



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Index</title>
</head>
<body>
<form method="POST" class="post-form">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="save btn btn-default">Save</button>
</form>
</body>
</html>
```

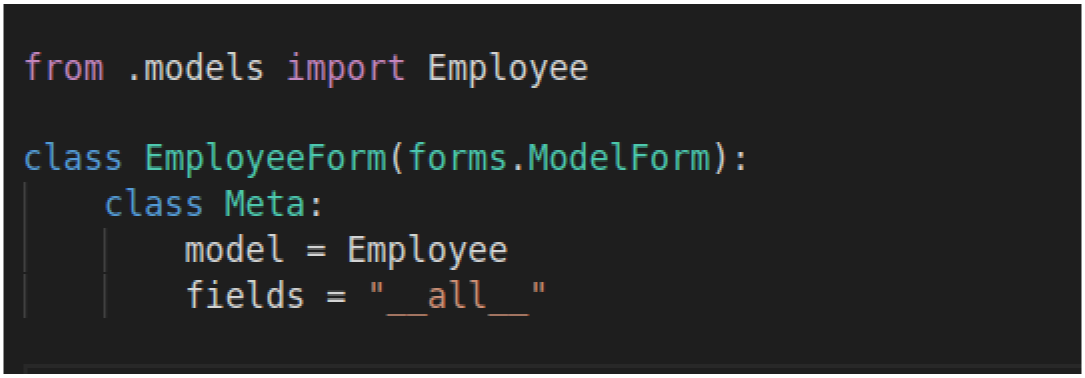
Djagno form validations

Models.py:-



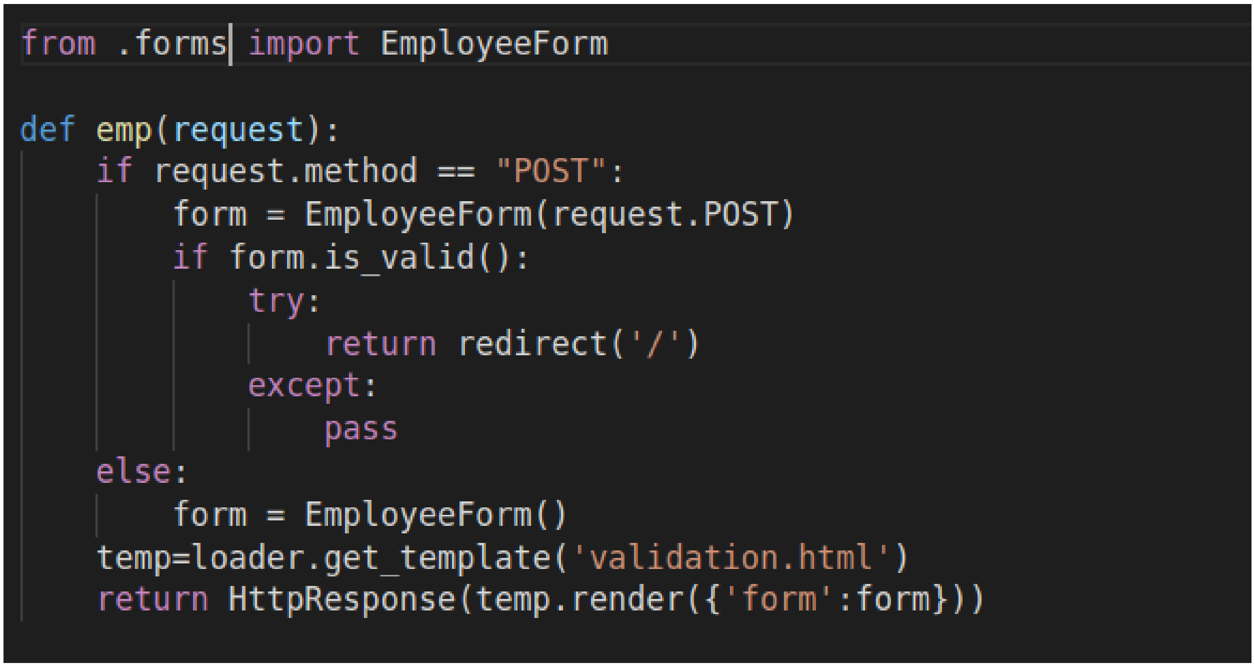
```
class Employee(models.Model):
    eid = models.CharField(max_length=20)
    ename = models.CharField(max_length=100)
    econtact = models.CharField(max_length=15)
    class Meta:
        db_table = "employee"
```

Forms.py:-



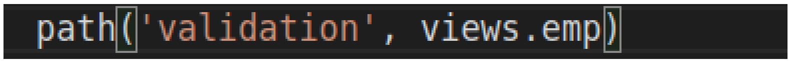
```
from .models import Employee

class EmployeeForm(forms.ModelForm):
    class Meta:
        model = Employee
        fields = "__all__"
```

Views.py:-



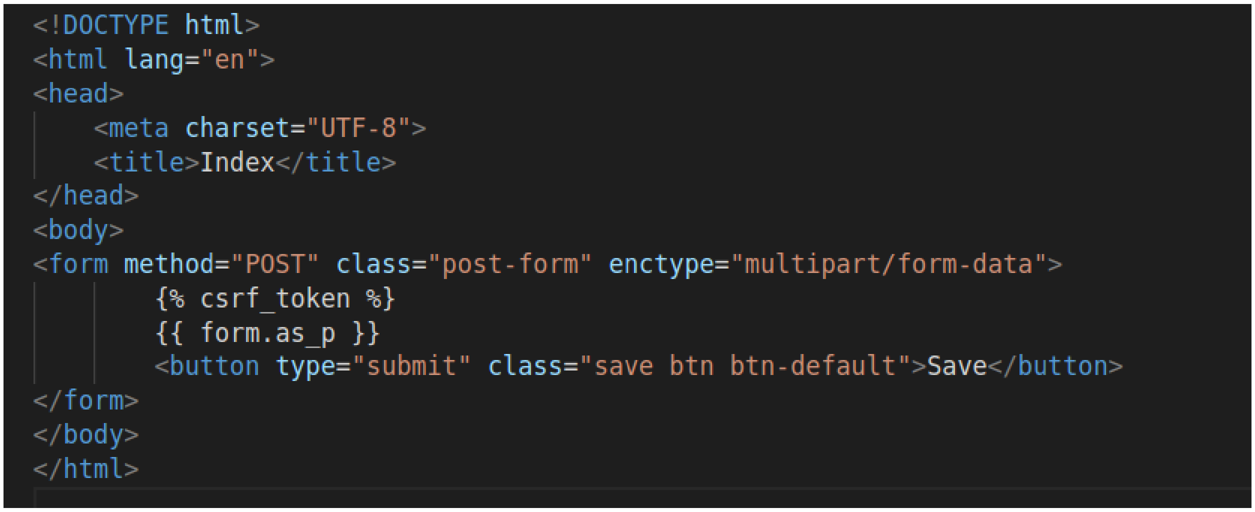
```
from .forms import EmployeeForm

def emp(request):
    if request.method == "POST":
        form = EmployeeForm(request.POST)
        if form.is_valid():
            try:
                return redirect('/')
            except:
                pass
    else:
        form = EmployeeForm()
    temp=loader.get_template('validation.html')
    return HttpResponse(temp.render({'form':form}))
```

Urls.py:-



```
path('validation', views.emp)
```

Validation.html



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Index</title>
</head>
<body>
<form method="POST" class="post-form" enctype="multipart/form-data">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit" class="save btn btn-default">Save</button>
</form>
</body>
</html>
```