Django Image Files

- ➔ Django provides built-in library and methods that help to upload a file to the server.
- ➔ The **models.ImageField()** is used to create a file input and submit the file to the server. While working with files, make sure the HTML form tag contains **enctype=”multipart/form-data”** property.

- **Complete Django-setup**

❖Image uploading

- **Create a templates/app folder and a HTML file(index.html) inside it containing form for accepting images.**

```
<center>
    <h1>Image Uploading Form</h1>
    <form action="" method="post" enctype="multipart/form-data">
      <table>
        <tr>
          <td>Image Name</td>
          <td><input type="text" name="imgname" id=""></td>
        </tr>
        <tr>
          <td><td><input type="file" name="image" id=""></td></td>
        </tr>
       <tr>
          <td>Upload image</td>
          <td><input type="file" name="image" id=""></td>
        </tr>
      </table>
    </form>
  </center>
```

- **Creating a view to show index.html**

```
def IndexPage(request):
   return render(request,"app/index.html")
```

- **Creating url for this view**

- ➔ In myapp/urls.py, first import views and then add a path for this view

```
from . import views

urlpatterns = [
    path("", views.IndexPage, name="index")
]
```

- **Create a database and connect Django with MYSQL**

➔ Make sure after doing this, run the following commands

```
python manage.py migrate
```

- **Create a model for storing images**

➔ In project models.py, make a model for storing images

```
class ImageData(models.Model):
    Imagename=models.CharField(max_length=50)
    Image = models.ImageField(upload_to="img/")
```

➔ Here ***upload_to="img/"*** is the path where we want the images to be stored in the server.

➔ You also have to install Pillow for this.

```
python -m pip install Pillow
```

➔ After doing this, run the following commands in terminal

```
python manage.py makemigrations
python manage.py migrate
```

- **Create a view to store image in database**

➔ In views.py file, first import models

```
from .models import *
```

➔ Create a view for the same

```
def UploadImage(request):
    if request.method=="POST":
        imagename=request.POST['imgname']
        pics=request.FILES['image']

        newimg=ImageData.objects.create(Imagename=imagename,
Image=pics)
        return render(request,"app/show.html")
```

- ### Add path for this view

➔ In myapp/urls.py, write the following statement

```
path("upload/",views.UploadImage, name="imageupload")
```

➔ Also make sure to add this in index.html form tag and make sure to write csrf_token below form tag

```
<form action="{% url 'imageupload' %}" method="post"
enctype="multipart/form-data">
        {% csrf_token %}
```

➔ Now start your server using following command

```
python manage.py runserver
```

❖Image fetching

- ### settings.py configuration (used to fetch image)

➔ First import os module and add below codes

```
Import os
MEDIA_ROOT = os.path.join(BASE_DIR, ‘media’)
MEDIA_URL = ‘/media/’
```

➔ **MEDIA_ROOT** is for server path to store files in the computer.

➔ **MEDIA_URL** is the reference URL for the browser to access the file over HTTP.

- ### urls.py configuration (used to fetch image)

➔ In project **urls.py** we should edit the configuration like this

```
from django.conf import settings
from django.conf.urls.static import static
if settings.DEBUG:
        urlpatterns +=static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

- **View for fetching Image**

➔ In views.py, create a view for fetching image

```
# Image Fetching View
def ImageFetch(request):
   all_img=ImageData.objects.all()
   return render(request,"app/show.html",{'key1':all_img})
```

- **URL for fetching Image**

➔ Add a path for the view in myapp/urls.py

```
path("showimg/",views.ImageFetch, name="show")
```

➔ Now we want whenever user upload a image, automatically show him all the images.

➔ It means we have to **call ImageFetch view from UploadImage view**. We know that this can only be done by **redirect.**

➔ Import redirect in views.py

```
from django.shortcuts import render,redirect
```

➔ In UploadImage view, delete the return statement and paste the following statement

```
return redirect('show')
```

➔ Here **show** is the value of the attribute **name** in urls.py which is used to call **ImageFetch** view.

- **Template to show images**

➔ show.html creation

```
<h1>Show Image</h1>
{% if key1 %}
{% for i in key1 %}
Image Name : {{i.Imagename}}
<br>
<img src="{{MEDIA_URL}}{{i.Image.url}}" alt="">

{% endfor %}
{% endif %}
```

➔ Here image url is saved in our database so we have to use **i.Image.url** to make it work.