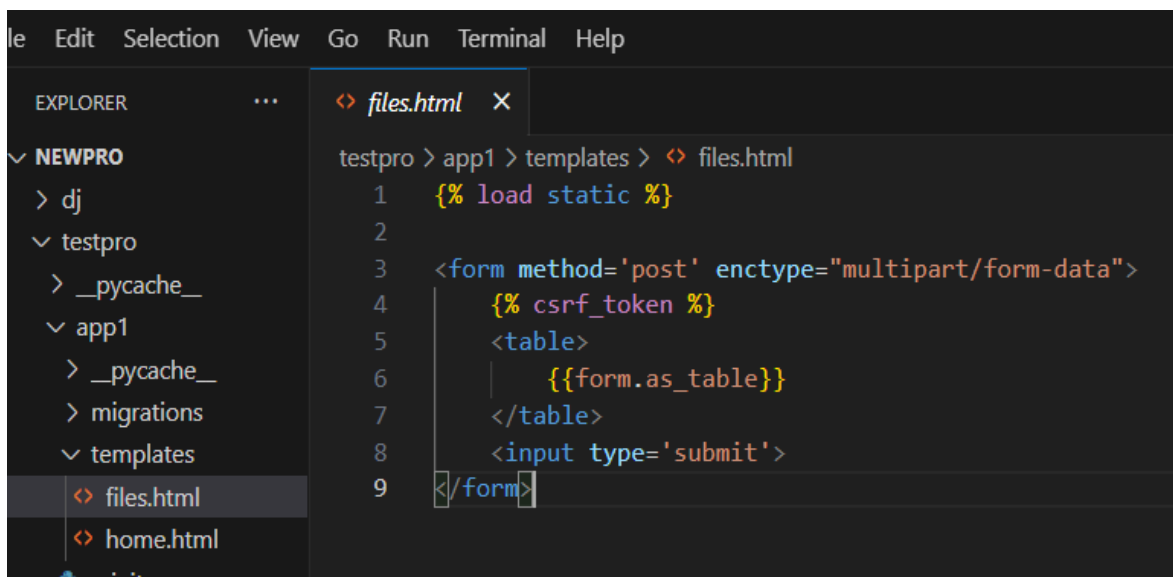


DJANGO IMAGE AND FILE UPLOAD:

Step-1:

Create a file.html page and create a form there with

- Method==post
- Enctype=multipart/form-data
- CsrfToken
- Then how to show the built in form(as table or as UL or as OL etc.)
- Then input method type = submit



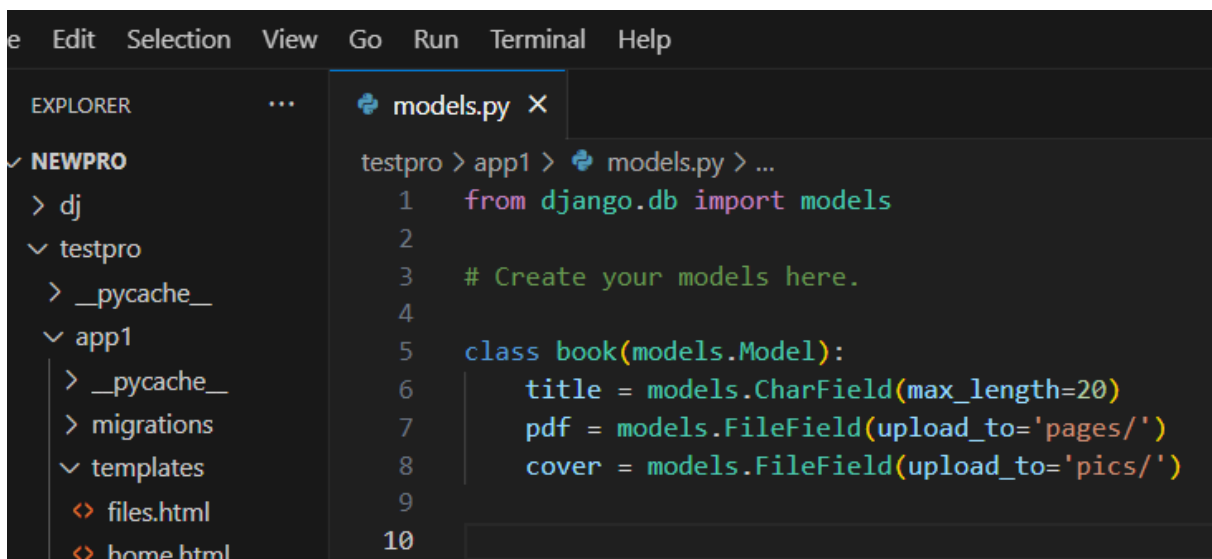
The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with folders 'dj', 'testpro', and 'app1'. Inside 'app1', there are files like '__pycache__', 'migrations', and 'templates'. The 'templates' folder is expanded, showing 'files.html' and 'home.html'. The main editor window displays the content of 'files.html'. The code in the editor is as follows:

```
testpro > app1 > templates > <> files.html
1  {% load static %}
2
3  <form method='post' enctype="multipart/form-data">
4      {% csrf_token %}
5      <table>
6          {{form.as_table}}
7      </table>
8      <input type='submit'>
9  </form>
```

Step-2:

then create a table in models.py

- Pdf field for upload pdf files=path set to pages folder
- Cover field for uploading images=path set to pics folder



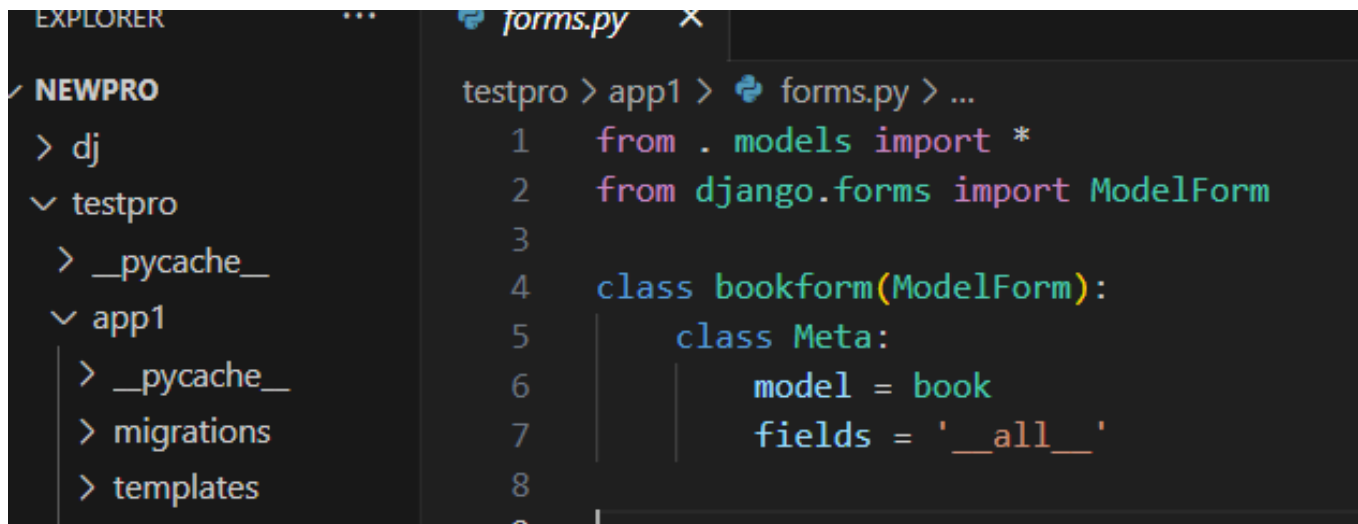
The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure as before. The main editor window displays the content of 'models.py'. The code in the editor is as follows:

```
testpro > app1 > models.py > ...
1  from django.db import models
2
3  # Create your models here.
4
5  class book(models.Model):
6      title = models.CharField(max_length=20)
7      pdf = models.FileField(upload_to='pages/')
8      cover = models.FileField(upload_to='pics/')
9
10
```

Step-3:

then create a forms.py file in app folder

- Import all tables from models
- Import model form from Django forms
- Create a new form which inherit from model form
- Create a meta class
- Model = our table name in models.py file
- Add all fields to fields variable

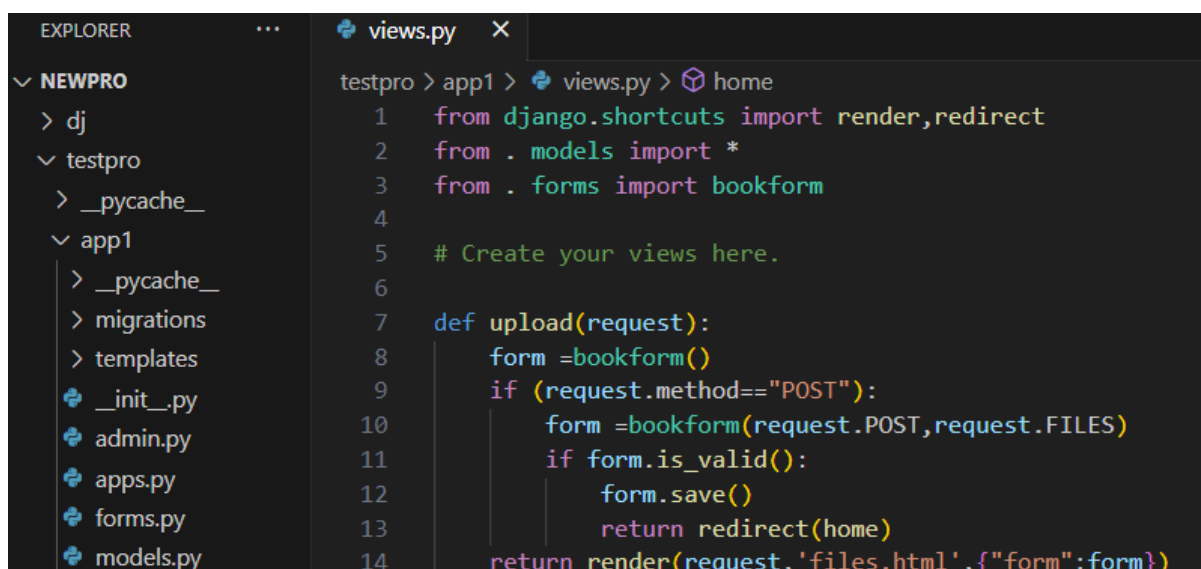


```
testpro > app1 > forms.py > ...
1  from . models import *
2  from django.forms import ModelForm
3
4  class bookform(ModelForm):
5      class Meta:
6          model = book
7          fields = '__all__'
8
9
```

Step-4:

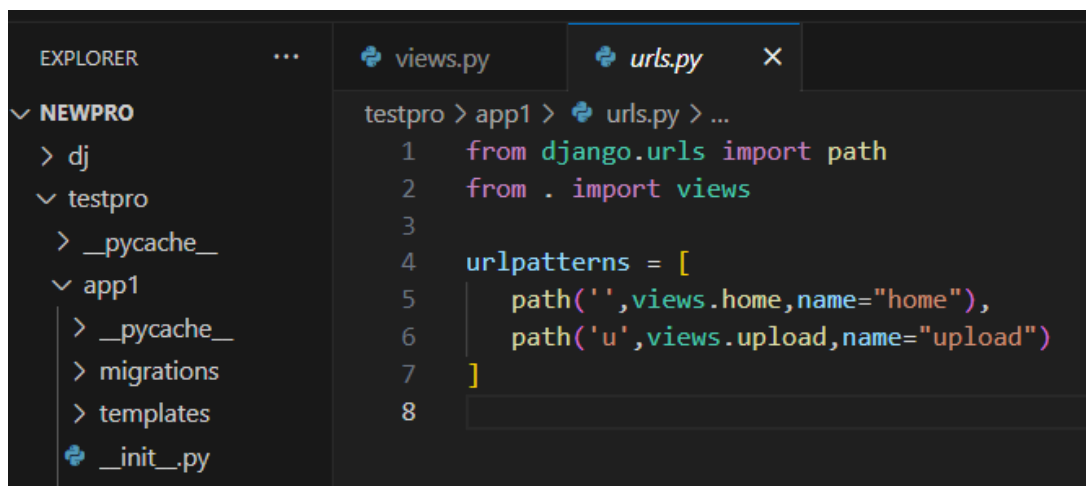
then create a view function to get the file and images in **views.py**

- Import models here
- Import our form from forms.py
- Create a upload function
- Assign form = created form in forms.py
- Check condition method is post
- Assign again: our created form(post method,get that files)
- If our form is valid then save it and go to homepage
- Pass form variable as dictionary in render function
{"form": form }



```
testpro > app1 > views.py > home
1  from django.shortcuts import render,redirect
2  from . models import *
3  from . forms import bookform
4
5  # Create your views here.
6
7  def upload(request):
8      form =bookform()
9      if (request.method=="POST"):
10         form =bookform(request.POST,request.FILES)
11         if form.is_valid():
12             form.save()
13             return redirect(home)
14         return render(request,'files.html',{'form':form})
```

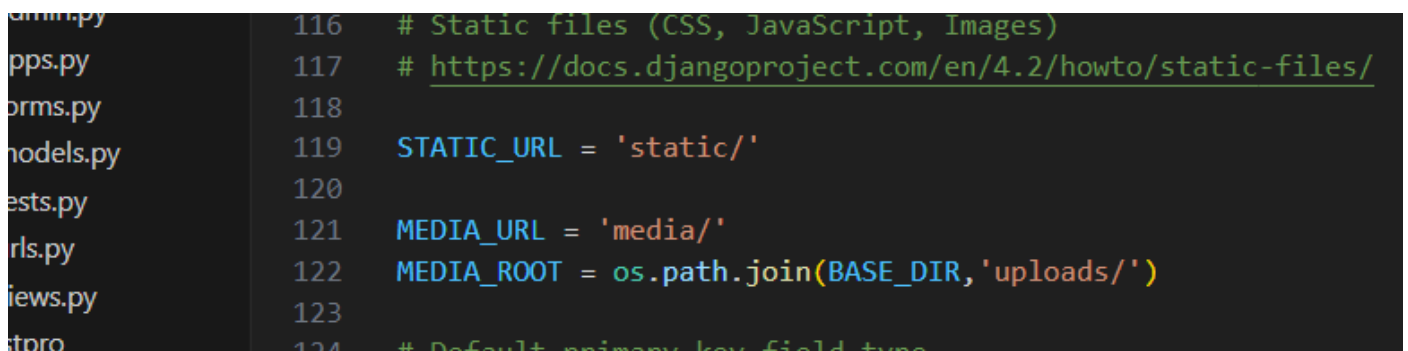
Add path for the view functions in urls.py file in app folder:



```
testpro > app1 > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.home, name="home"),
6      path('u', views.upload, name="upload")
7  ]
8
```

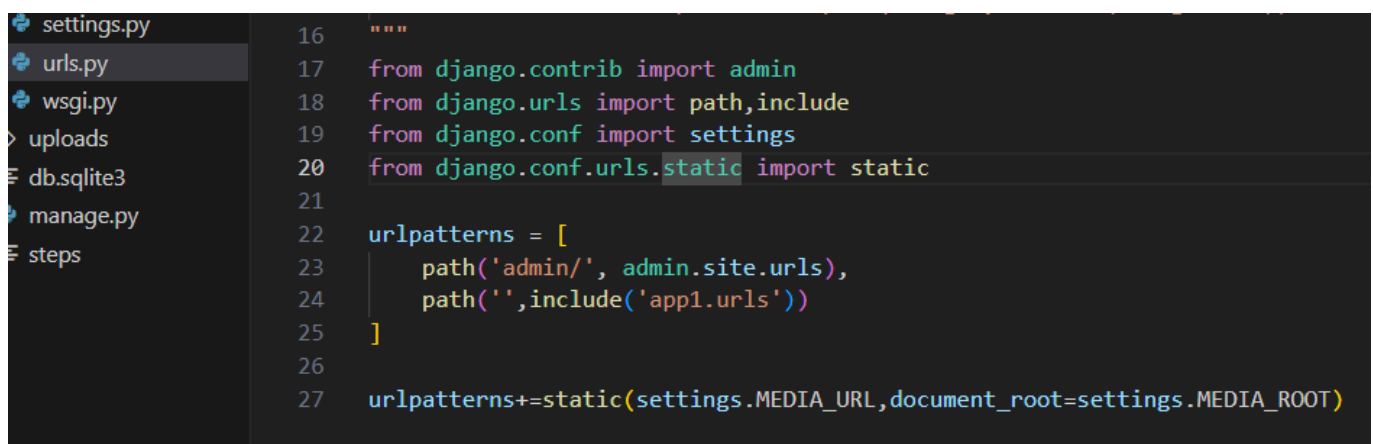
Step-5

Add media url path and root in **settings.py** file in project folder



```
116  # Static files (CSS, JavaScript, Images)
117  # https://docs.djangoproject.com/en/4.2/howto/static-files/
118
119  STATIC_URL = 'static/'
120
121  MEDIA_URL = 'media/'
122  MEDIA_ROOT = os.path.join(BASE_DIR, 'uploads/')
123
124  # Default primary key field type
```

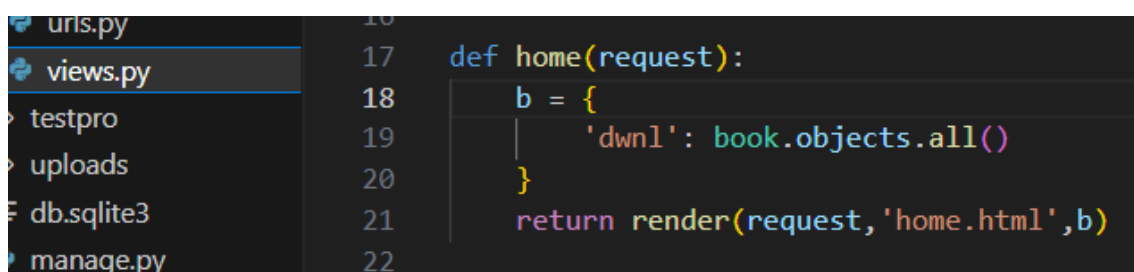
then add url in settings add path in **urls.py** file in project folder



```
16  """
17  from django.contrib import admin
18  from django.urls import path, include
19  from django.conf import settings
20  from django.conf.urls.static import static
21
22  urlpatterns = [
23      path('admin/', admin.site.urls),
24      path('', include('app1.urls'))
25  ]
26
27  urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Step-6:

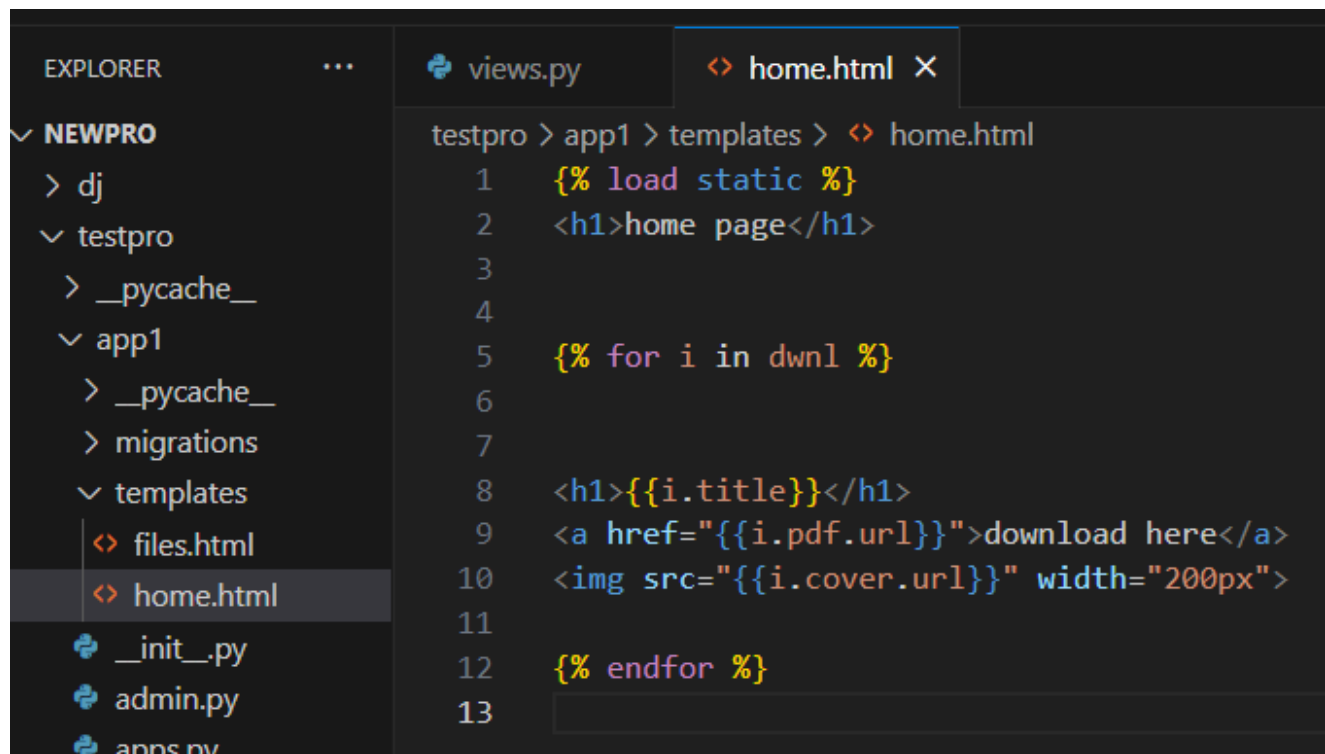
then create a dictionary for get the link of each file fields



```
17  def home(request):
18      b = {
19          'dwnl': book.objects.all()
20      }
21      return render(request, 'home.html', b)
22
```

Step-7

fetch link of each files in new html page home.html



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under 'NEWPRO'. The 'templates' folder is expanded, showing 'files.html' and 'home.html'. 'home.html' is selected. The main editor area shows the content of 'home.html' with the following code:

```
testpro > app1 > templates > <> home.html
1  {% load static %}
2  <h1>home page</h1>
3
4
5  {% for i in dwnl %}
6
7
8  <h1>{{i.title}}</h1>
9  <a href="{{i.pdf.url}}">download here</a>
10 
11
12 {% endfor %}
13
```