**‘Fresh install of django in virtual env**

Mkdir djangoProject

Cd djangoProject

Pip install virtual env

Virtualenv env

env\Scripts\activate

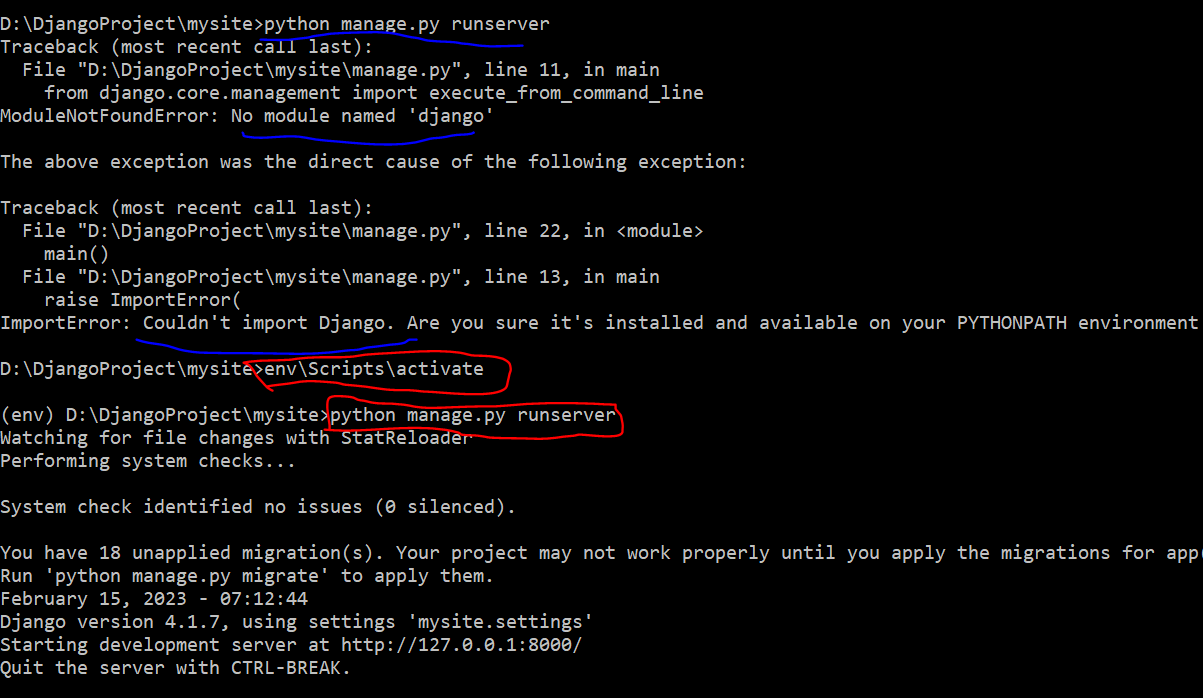
Pip install django

Django-admin

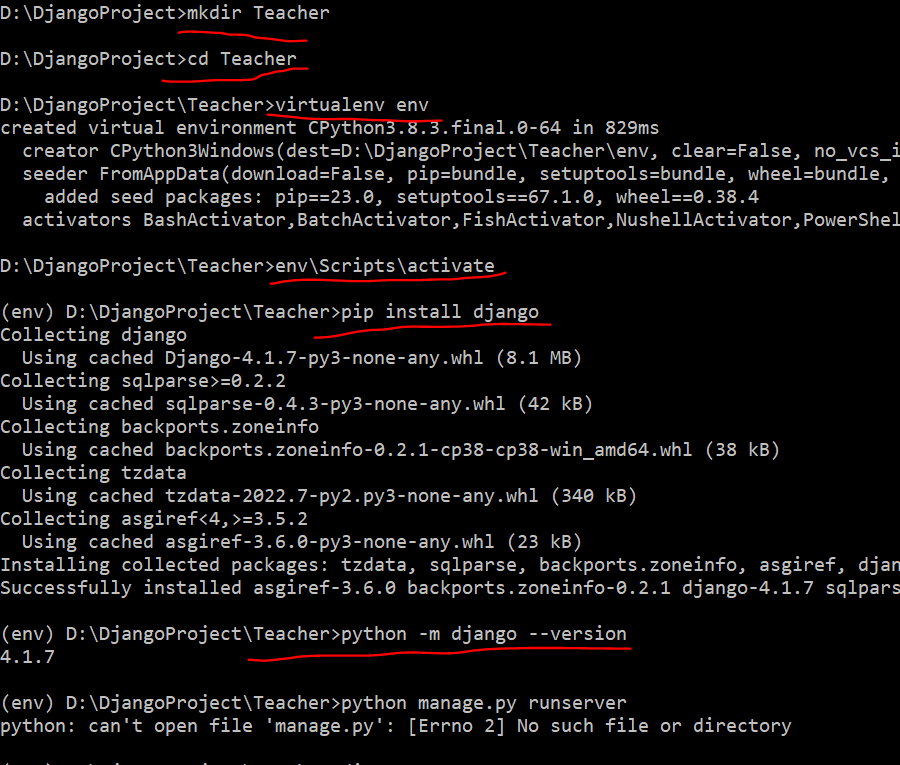
Python mange.py runserver

Python –m django --version

**First activate the virtual env then run mange.py**



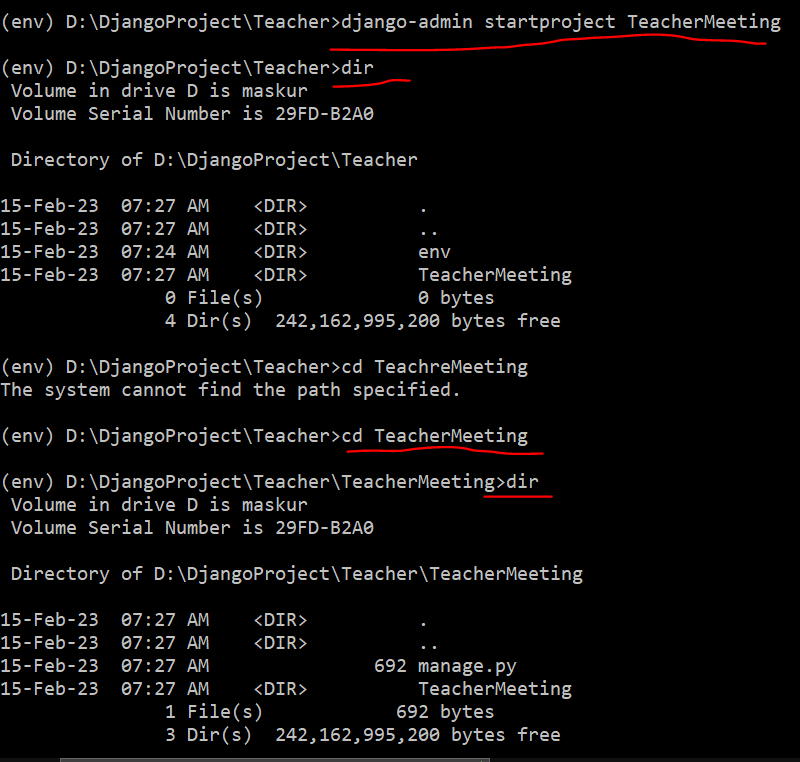
Complete guideline to install django

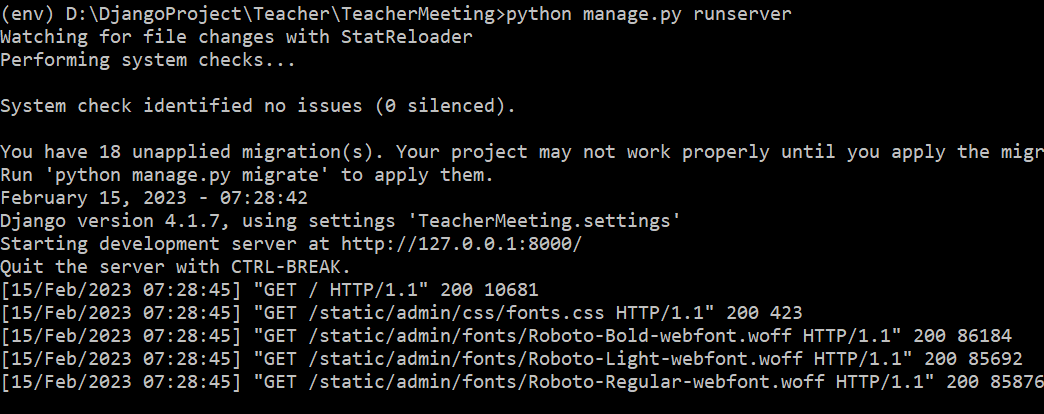




If the the app level template doesn’t find then check the project settings add the app name in installed apps

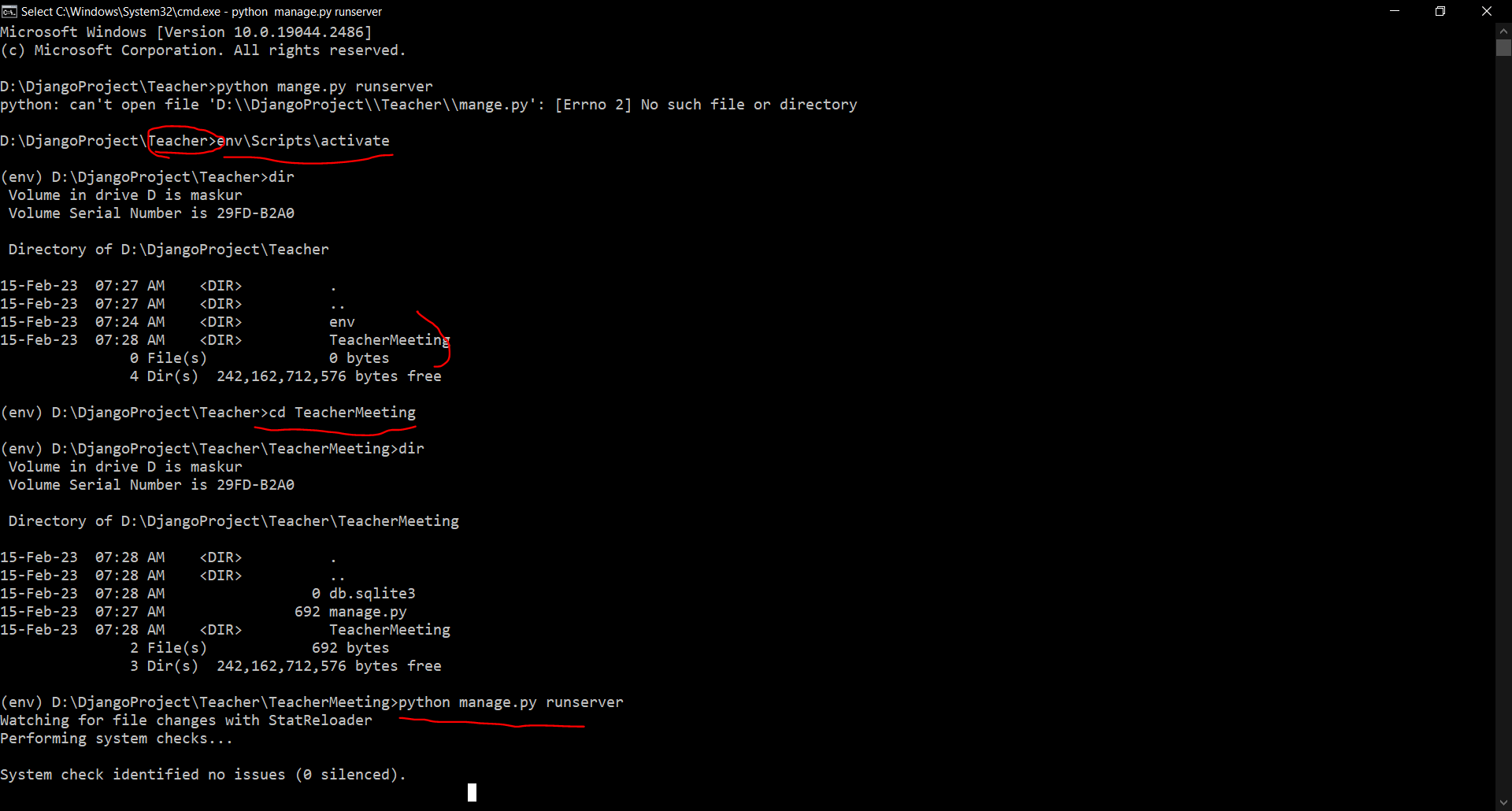
Project level template add -> go to settings then add os.path





Now you can go to browser : 127.0.0.1:8000

After closing this terminal and again to start server go to the Teacher folder and activate the env by (env\Scripts\activate)



Base.html

<body>

{% block content %}

{% endblock content %}

</body>

Home.html

{% extends "base.html" %}

{% block content %}

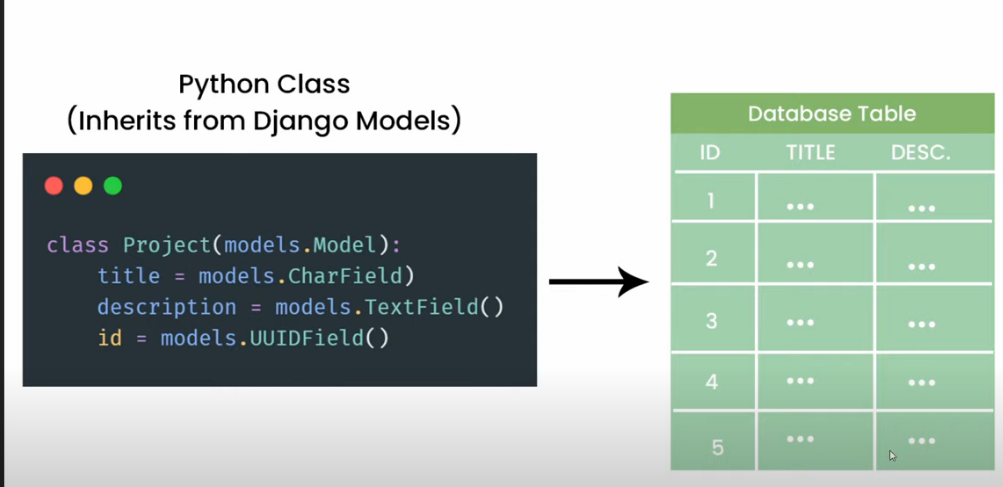
<h1>Welcome</h1>

<p>This is the home page</p>

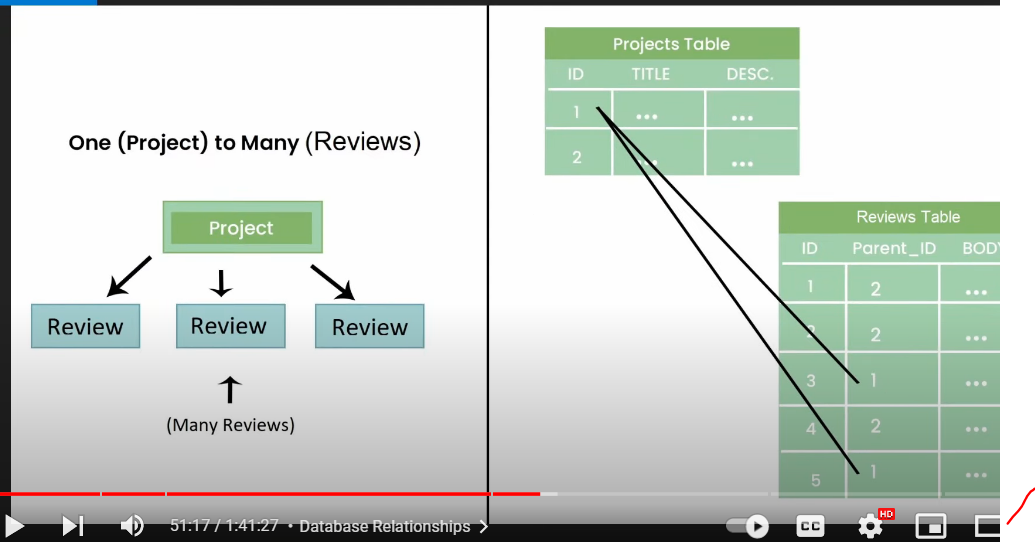
{% endblock content %}

Python manage.py migrate

Python manage.py createsuperuser



Python manage.py makemigrations



Linking url path in html page :

Two ways by url name or direct path

            <td> <a href="/project/{{project.id }}"> {{project.title }} </a></td>

Github:

Git status

Git add –a # it will add all file for commit

Git commit # this will commit all the file in local

Then there will be open a msg prompet in cmd to enter msg press (i) to close the editor press ctrl+c then type :wq

* Then to push the local commit to remote server send .
* Git push origin –main
* To add remote server use
* Git add remote “LINK of the project”

git checkout -b main # this will create a branch main locally and remotely

Add static file :

To do this create static folder in the project

Then add this folder in the setting.py

STATICFILES\_DIRS = [

   BASE\_DIR / 'static'

   #os.path.join(BASE\_DIR,'static')

]

To set the upload image location use media root in setting:

MEDIA\_ROOT = BASE\_DIR / 'static/images'

Use the media url for user upload which define the directory wehere the user upload content will be saved :

MEDIA\_URL = '/images/'

Configure the urls in main project urls.py

urlpatterns+=static(settings.MEDIA\_URL,document\_root = settings.MEDIA\_ROOT)

use a featured image field in database models

    featured\_image = models.ImageField(null=True,blank=True)

we can access the image in html page directly :

<img style="max-width: 200px;" src="{{book.featured\_image.url}}" alt="">

Or specifiy it in the function in models

define the function to return the file path of image to show in the html page

   @property

    def imageURL(self):

        try:

            img= self.featured\_image.url

        except:

            img=''

        return img

in html to show the form we can use the shortcut :

{{form.as\_p} or we can use custom way .

Use enctype=to submit pdf,image file

<form action="" method="POST" enctype="multipart/form-data">

    {% csrf\_token %}

     {% for field in form  %}

        <div>

            {{field.label}}

            {{field}}

        </div>

      {% endfor %}

    <input type="submit" value="Create">

</form>

And we need to change the views.py where we add request.files

       form = ProjectForm(request.POST,request.FILES)

production :

set debug =False

add the static root = base\_dir / ‘staticfiles’

python manage.py collectstatic

confifure the urls

urlpatterns+=static(settings.STATIC\_URL,document\_root = settings.STATIC\_ROOT)

whitenoise install and add to middleware