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

Mkdir djangoProject

Cd djangoProject

Pip install virtual env

Virtualenv env

env\Scripts\activate

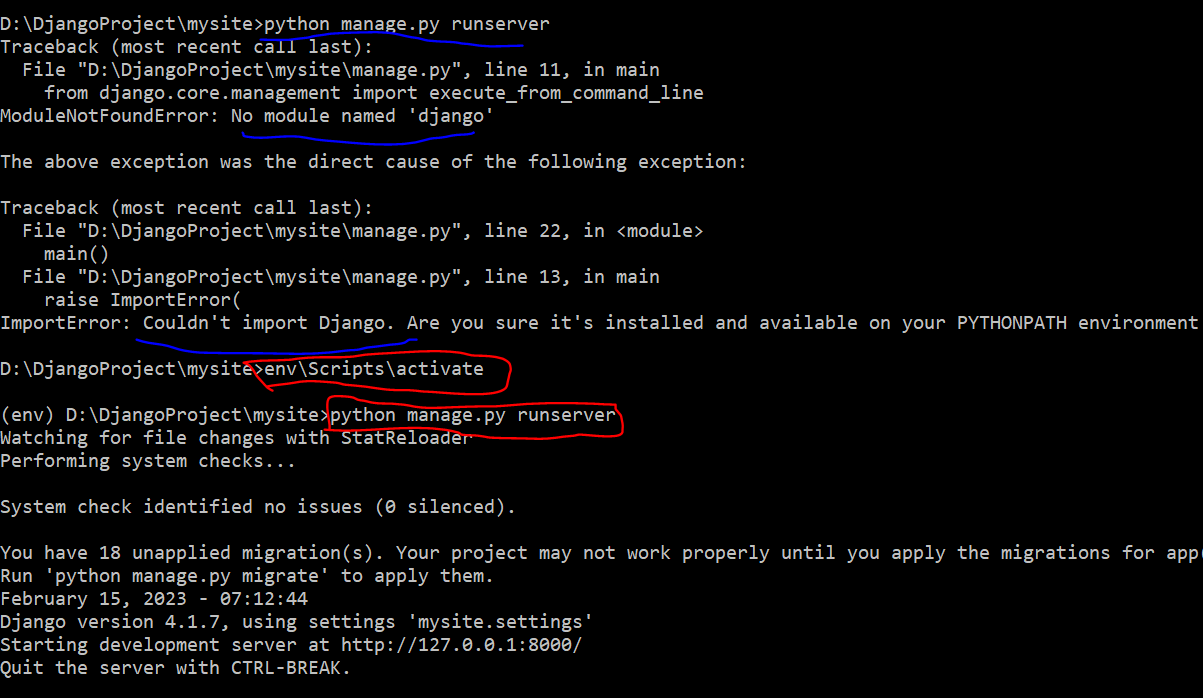
Pip install django

Django-admin

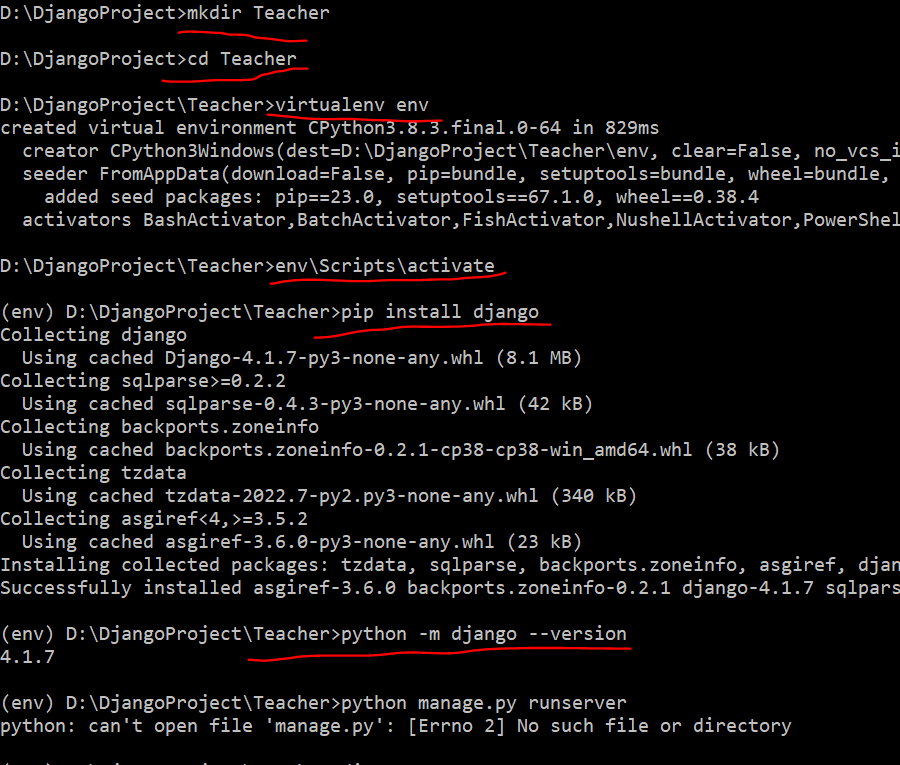
Python mange.py runserver

Python –m django --version

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



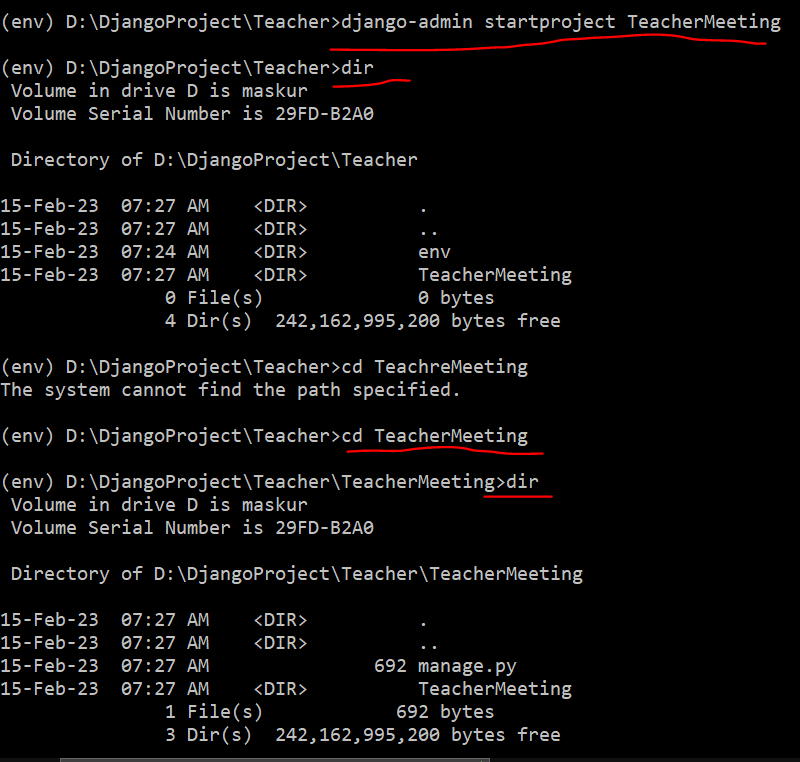
Complete guideline to install django

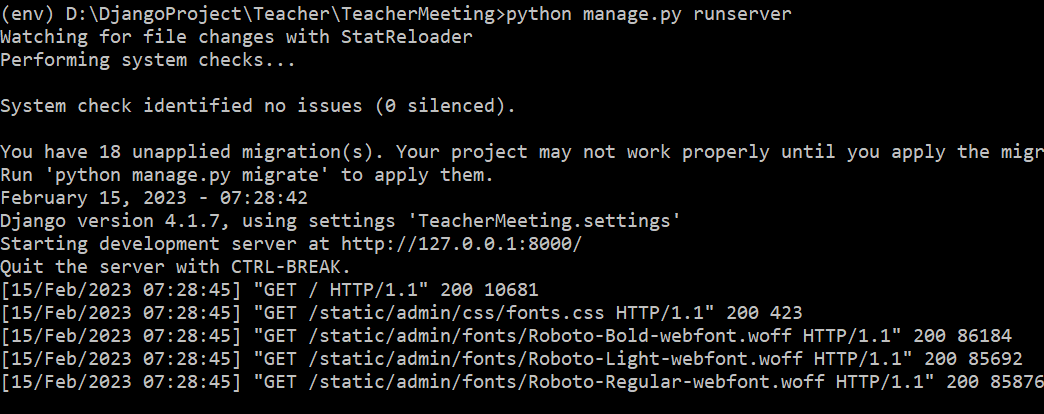




If the the app level template doesn’t find then check the project settings add the app name in installed apps

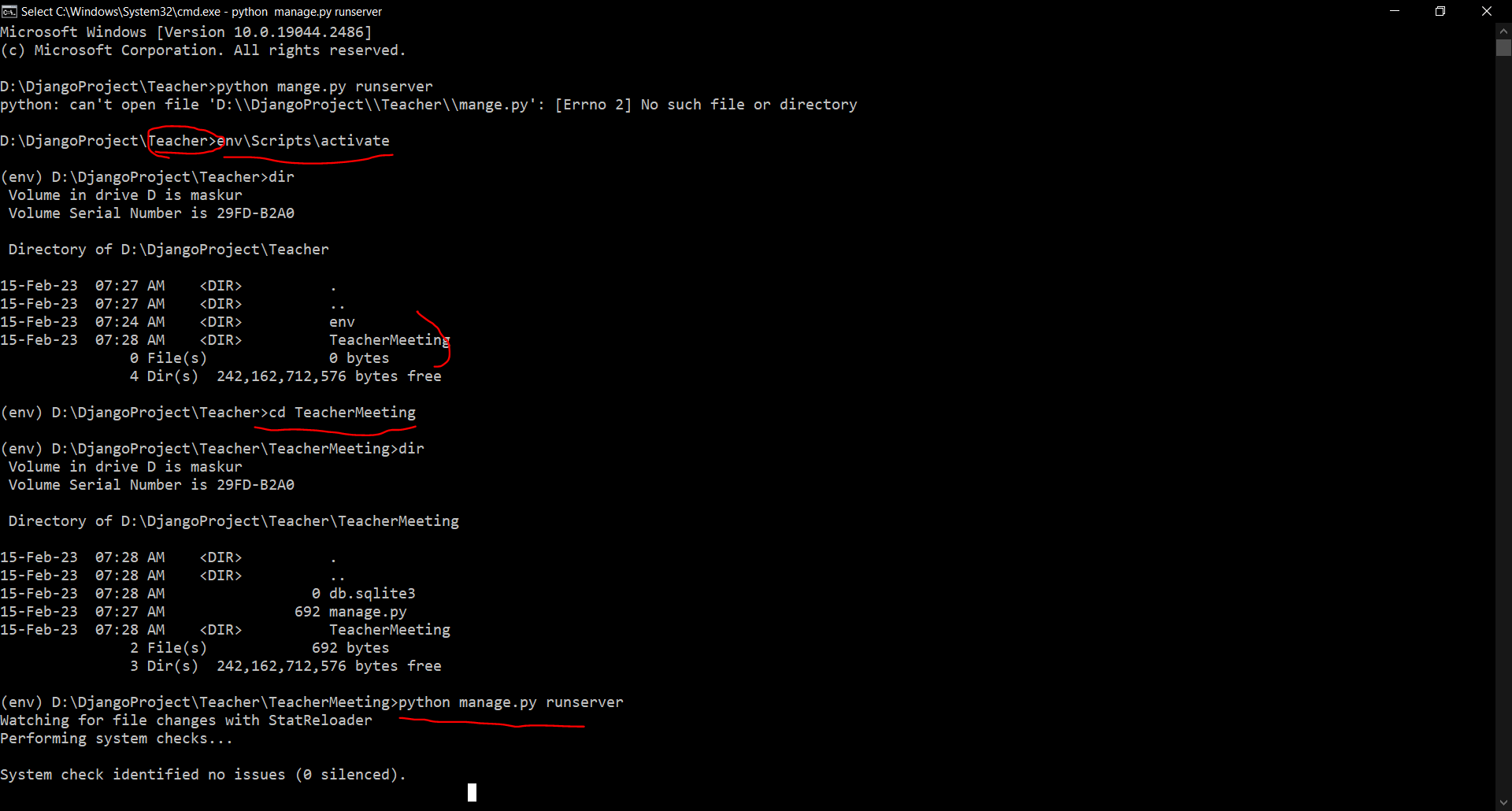
Project level template add -> go to settings then add os.path





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 %}

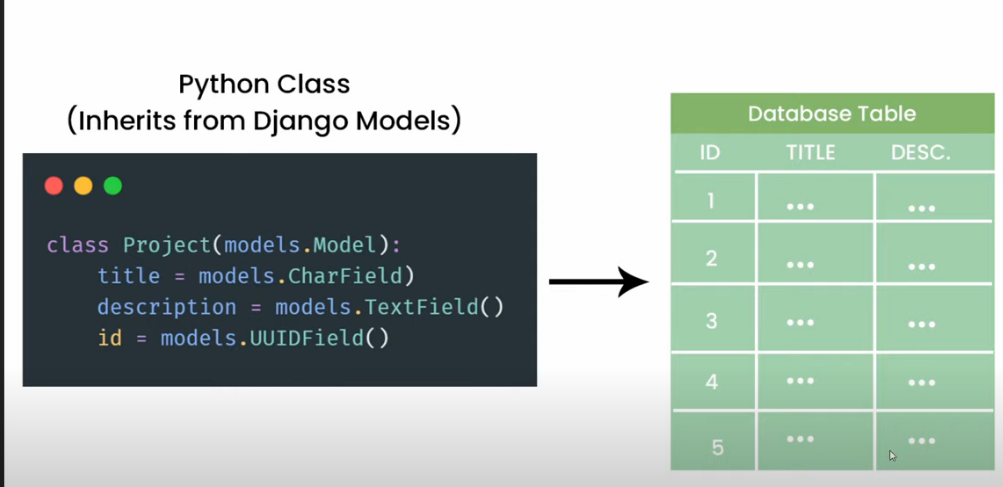
<h1>Welcome</h1>

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

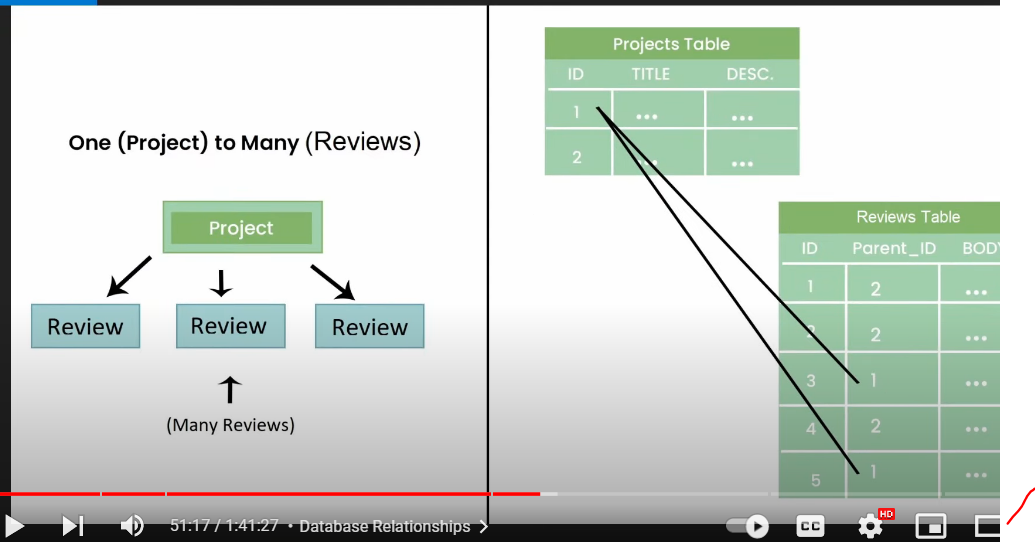
{% endblock content %}

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:first time

git init

git add -A

git commit -m "Initial commit"

git branch -M main #this wil change the baranch master to ma

git remote add origin https://github.com/your-username/your-repository.git

git push -u origin main

…………………….

Git status

Git pull origin main

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

------Migrating a single app:

Python manage.py makemigrations app\_name;

Xxx\_migrations file will generate

Python manage.py sqlmigrate app\_name Xxx\_file

Python manage.py migrate

Django restful Api

Mkdir restCrud

Cd restCrud

Virtualenv env

Env\scripts\activate

Pip install django-admin

Django-admin startproject restfulApiCrud

Cd restfulApiCrud

Code .

Python manage.p startapp employeeApi

Pip install djangorestframework

Set the installed app: rest\_framework

For postgres

Pip install psycopg2

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'restfulapidb' ,

        'USER': 'postgres',

        'PASSWORD': 'postgres',

        'HOST' :'localhost'

    }

}

Changing superuser password:

Python manage.py changepassword user\_name

Broken pipe error

Use a try catch block to rescue any exception

Deploy app to heroku

>pip install psycopg2 gunicorn django-heroku whitenoise

Collecting psycopg2

Procfile

runtime

pip freeze > requirements.txt

Deploy on Render

Add the staticurlpatterns so that in server gunicorn can detect

from django.contrib.staticfiles.urls import staticfiles\_urlpatterns

urlpatterns = [

    path('admin/', admin.site.urls),

    path('', include('todoapp.urls')),

]

urlpatterns+=staticfiles\_urlpatterns()

To add django database url : pip install dj-database-url

Pip install gunicorn

pip install psycopg2-binary

Create a postgres database in render

Copy external database link

Go to settings and set this

DATABASES ={

    "default":dj\_database\_url.parse(os.environ.get("DATABASE\_URL"))

}

Got to project cmd and set the dabatase\_url

set DATABASE\_URL=postgres://tickler\_user:l4FO2axIW2qdZPQ8clUhvXsvbilrJ6cr@dpg-ckglbskldqrs73dkj8v0-a.oregon-postgres.render.com/tickler

echo %DATABASE\_URL%

python manage.py migrate

ALLOWED\_HOSTS = ['\*']

Type this to add all the apckeges nedded to

pip freeze > requirements.txt

then changes the requirenst.txt manually set crpytographay==2.8 if there there is a issue in the build process when deploy on render .

upload the project on github

init git inside the project and then upload

after uploading go to render website

**create webservices and select github then select the project**

**give the name of the webservice**

**in gunicorn app write gunicorn projectName.wsgi:application**

**go to advanced environment variable then add**

**PYTHON\_VERSION =3.8.3**

**DATABASE\_URL = PASTe THE INTERNAL DB URL FROM RENDER**

**THEN ITS WORK FINE**

**Static file handling**

**1.pip install whitenoise**

**2.** STATIC\_ROOT = BASE\_DIR / "staticfiles"

**Put this at the bottom of the settings file**

**3.**     "whitenoise.middleware.WhiteNoiseMiddleware",

**Put this whitenoise middleware just below the security middleware**

**4. python manage.py collectstatic**

**And also you must flow the django static file directory format in the app level todoap/statcic/tdoapp/css/style.css like this .**

**And use** <script src="{% static 'todoapp/js/script.js' %}"></script>

**Not the direct url path of the css .**

**Exception**

OperationalError at /register/

could not translate host name "dpg-ckglbskldqrs73dkj8v0-a" to address: Name or service not known

this is the issue of the database check the database url is set properly both external and internal in render

set DATABASE\_URL = “THE URL”

RestaApi

Pip install django

Pip install djangorestframework

Add the rest\_framework on installed app

Create serializer.py in the app