***DJANGO NOTES***

To create a project :

In the terminal write

*django-admin* → (this command will give us multiple admin options)

*django-admin startproject projectname*

manage.py → it is a command line utility which will help us to interact with our Django project

The folder inside the directory with the same name is the package in which we will work

A package must contain \_\_init\_\_.py file

Urls.py → this contains our url declarations, mapping.

Wsgi.py → WSGI stands for web server gateway interface

Then change the directory by (in terminal if reauired)

*cd .\Foldername\*

Then to run the Django project write

*python manage.py runserver*

Press Ctrl + C to quit the server

To go back to previous folder write *cd..*

*<div class = “container”>* → This means elements middle mein hi rahenge and borders tak extend nahi hoge

*<div class = “container-fluid”>* → This means border tak extend hoga

*<body dir = ‘ltr’>*

It means the direction will be from left to right

To adjust the position of the components we can either use <br> tag or we can write inside class mt-2 (margin top)

*<h2 style="text-align: center">Center-aligned Text</h2>*

To center align your text

Default method is get

Django by default takes into consideration CSRF (Cross site request forgery)

Network mein new line character ko transport karne ke liye **\n and \r** dono daale jaate hai

ASGI and WSGI are our servers. By default Django uses WSGI server.

ASGI stands for asynchronous web server gateway interface.

If we want to use normal http requests we use WSGI but if we have to use web sockets as well as http requests then we use ASGI

Projects and apps are 2 different things in Django. A project can have multiple apps and an app can be in multiple projects.

An app is a pluggable web application

To create apps in Django project write this command in the terminal

*Python manage.py startapp appname*

CDN stands for content delivery network

The database schema will be in form of models (what all tables will be there, …. )

*python manage.py migrate*

migrate here means apply all migrations

migrations ka matlab hai jo change maine model mein kiya hai usko main store kar raha hoon

For example, maano meri table ka naam tha products, I am changing that name to products2, toh hum uska migration bana denge. Agar maine uska migration bana diya hai iska matlab humne database nahi change kiya hai, humne bas yeh information store kar di ki yeh change humara aane wala hai and if we apply the migration then we will apply the changes in our database

*Python mange.py makemigrations* will store migrations

To apply changes to our database write

*Python manage.py migrate*

To add data in our database, we need to make a superuser. We make a super user by writing this command in the terminal

*Python manage.py createsuperuser*

Agar hum koi bhi table bana rahe hai models.py mein, toh hume apne models ko register karna padta hai admin.py mein

Django works on MVT pattern (Model view template)

**Framework** – a set of rules, ideas or a structure is already there, and following this structure we will build your application

A framework is a set of rules, ideas or beliefs which we use in order to deal with problems or to decide what to do

Framework is there in all computer languages

Django is a high level python web application framework that enables the rapid development of web applications.

It is an open source python framework.

Follows the model view template architectural pattern. Most of the frameworks work on model view controller (MVC) pattern.

Most popular python framework

**Why to use Django?**

* It is fast and simple.
* It is secure.
* It suits any web application project.
* It is well established.
* MVT Support and object oriented.
* Built in authentication and authorization.
* Packaging system

Pip is a dependency manager

Companies which use Django

* Youtube
* disqus
* Instagram
* spotify
* dropbox
* mozilla

**MVT:**

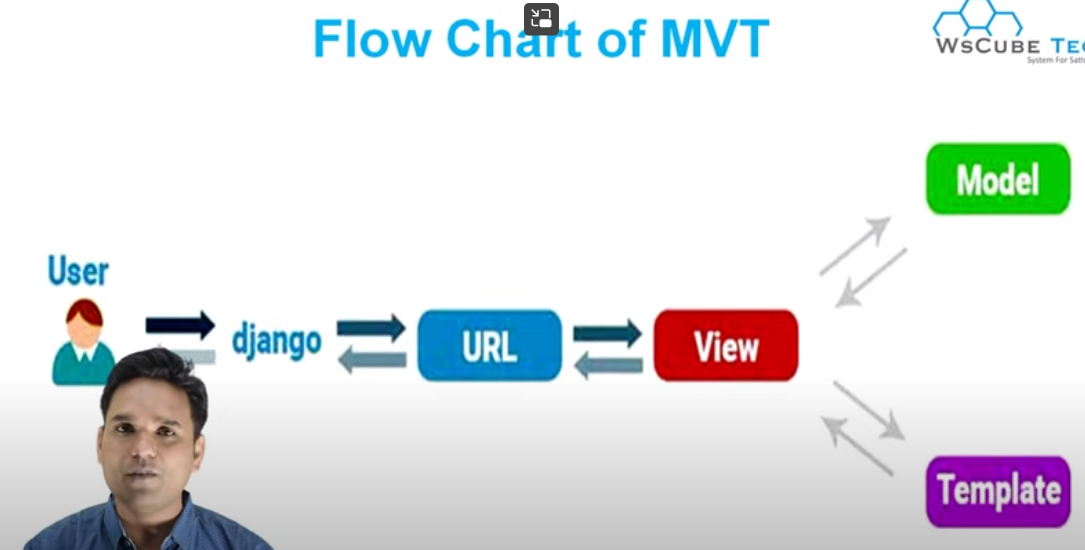
MVT is a software design pattern

Model : It is going to act as the interface of our data. Database se sara related kaam model ke processes ka hi hota hai

View : The view is the user interface – what you see in your browser when you render a website. *View function based bhi hota hai and class based bhi hota hai.* Example : ek particular URL pe ek particular view ko call karo

Template : A template consists of static parts of the desired HTML output as some special syntax describing how dynamic content will be inserted. Yahan pe hum data ko dynamic karenge, form banayenge waha se leke view pe bhejenge and view is connected to model.

All the three things Model, view and template are connected to each other.

Flowchart of MVT

User → URL → View → model → template

View pehle model se data ko lata hai and uska data laake template ko de deta hai. Model se data lana matlab product ka data laana, kisi category ka data laana.

Pip freeze (gives us the list of all the packages installed)

To change port number → *python manage.py runserver 4444* (Here 4444 is the port number). Also we may need to change the port number for cases lets say - when we are working in angular and Django and both of them are using the same port numbers. This can cause problem

Sqlite is the default database used by django

If we want an html page in our website then we will create a folder named template inside which all the html files will be stored.

We will have to create a folder named static also which will store all the static files (javascript files, css files, font files, images),

Another folder named media will also be created which will store all the dynamic images and files.

Middleware is used to implement restrictions like only the authorized person should open the admin panel.

Migrations – Kuch tables ka schema Django ke pass already rehta hai. Migrate karte hi tables database mein create ho jaayegi

*Python manage.py make migrations*

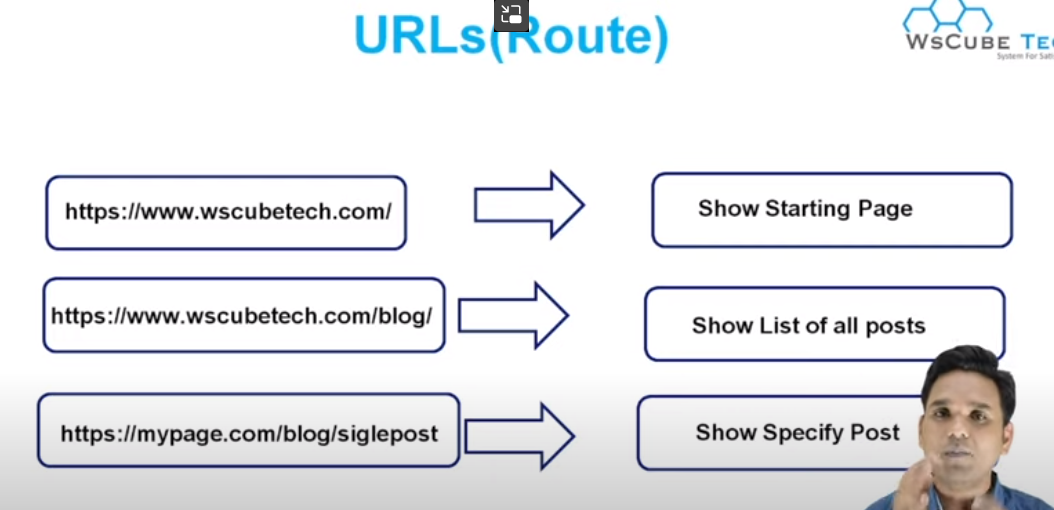
*Python manage.py migrate*

To view db.sqlite3 we need to download db browser (sqlitebrowser.org) because the data stored inside it is in binary form.

Regardless of the database used, the default tables will be migrated

To create a superuser write the command : *python manage.py createsuperuser. Then insert the username and password accordingly.*

The info related to super user is stored inside auth\_user table in db.sqlite3



All the URLs will be connected to some views

**Dynmaic URLs** : 1000 products ke liye 1000 pages nahi honge. Ek hi humara details page hoga who dynamic route ke help se manage hota hai

Routes integer, string, slug basis pe ban sakta hai

Slug – (combination of words and dashes)

Home page ka url blank rahega

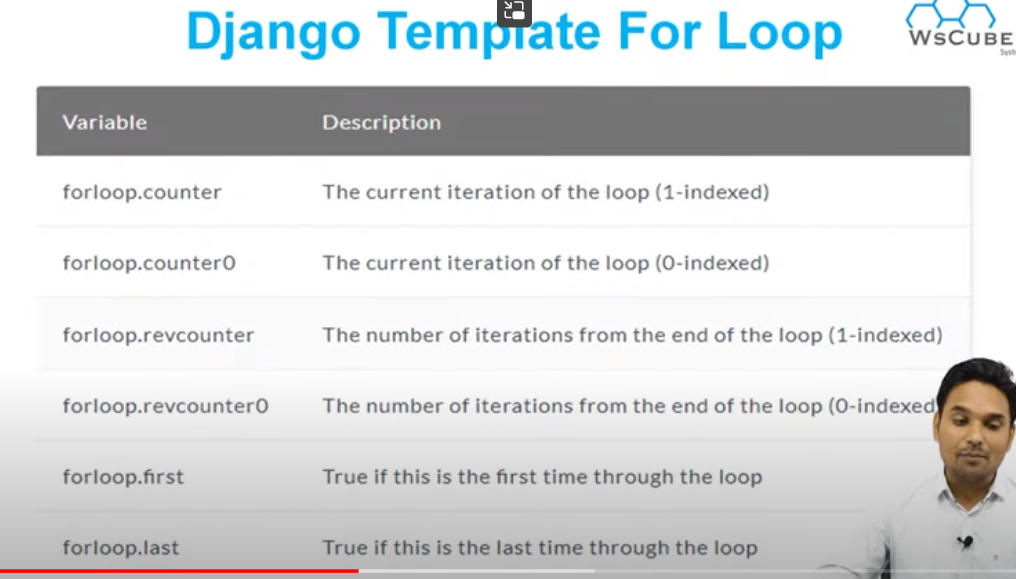
For loop in django template

*{% for i in list %}*

*<div> {{i}} </div>*

*{% endfor %}*

A picture containing text, screenshot, software, multimedia software

Description automatically generated

Fetching elements of the dictionary in a template

A screen shot of a computer code

Description automatically generated with low confidence

If we don’t write <div> tag inside for loop then all the elements will be displayed in a line.

If we want to print each of the elements on a new line then we use the <div> tag

If elif else statements in Django template

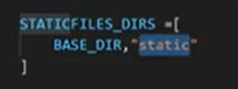
A picture containing text, font, number, screenshot

Description automatically generated

Agar humare pass ek .html page hai jisme humne html, css, javascript use kiya hai toh Django mein .html file ke contents hi display honge.

To call the css and js files, copy all those files and paste it inside static folder

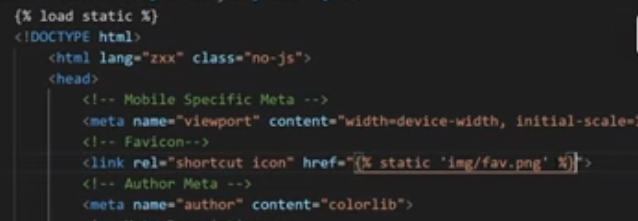
Then write this piece of code inside settings.py



Change the path in href tag since all the css and js files have been shifted inside static folder

Or else in the html page at the top line write

*{% load static%}* and to use it now, edit the href tag as follows



Header and footer part **include** in Django template:

If the header and footer part of each page should remain the same, one way to do is by copying that content and pasting it in each of the html pages. But if I want to change it I will have to change it in every page. So there is a better way to do it

Create a header.html inside templates. Copy that particular piece of code. Paste in inside header.html file

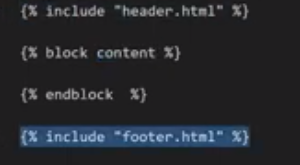
A screen shot of a computer

Description automatically generated with medium confidenceA screen shot of a computer code

Description automatically generated with low confidence

Extends – Django template tags

Create a base.html page and write this piece of code inside it



Now in index.html write :

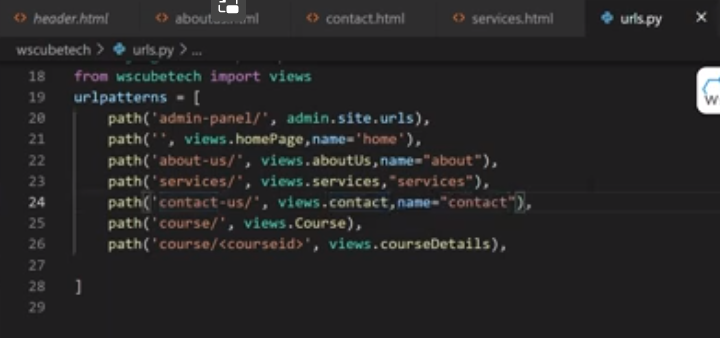
A screen shot of a computer screen

Description automatically generated with low confidence

At the end of the block content write {% endblock %}

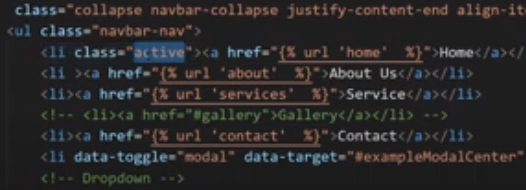
In urls.py the first parameter should match with the link/name in the href tag (href = “/about-us”).

Another way is

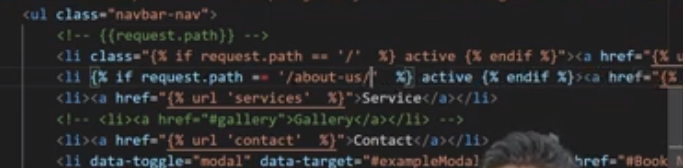
write name of parameter inside the path in urls.py

then edit the href tag as

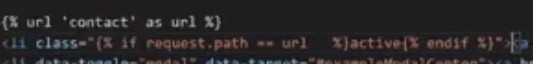
How to highlight link in Django



Use if condition

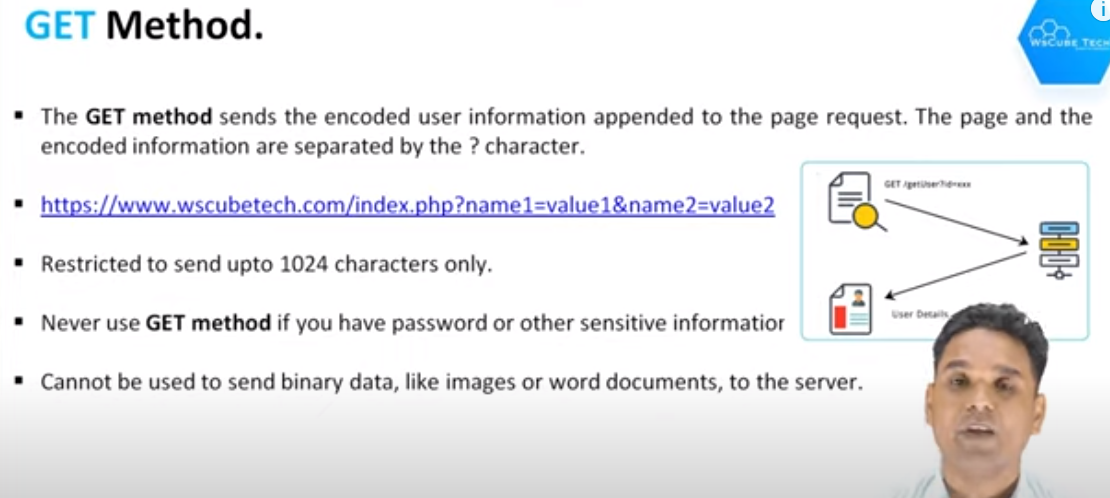


Another way :



Here href = {{url}}

**HTTP request methods**

1. GET method
2. POST method

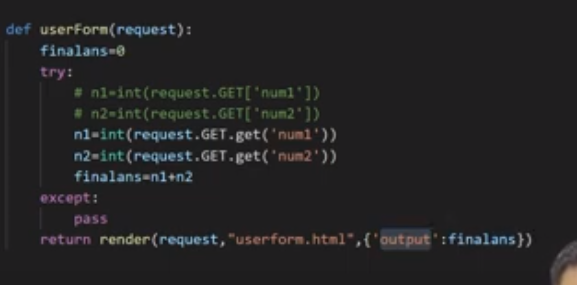
A screenshot of a computer

Description automatically generated with low confidence

With the help of post method we can handle images also

A screen shot of a computer screen

Description automatically generated with low confidence





**Page redirection in Django**

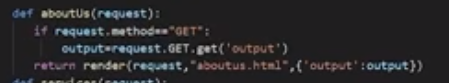
*From Django.http import HttpResponseRedirect*

…

*Return HttpResponseRedirect(‘/about-us’)*

A picture containing text, screenshot, font

Description automatically generated



**Django html form action URL**

Whenever we include action in form it means all the data of that form will be sent to the URL which is mentioned inside action



**Forms in Django**

Create a forms.py file in Django

Then for example write this piece of code inside it

*From django import forms*

*Class UsersForm(forms.Forms):*

*Num1 = forms.CharField(label = “value1”, required = False, widget = forms.TextInput(attrs = {‘class’ : “form-control”}))*

*Num2 = forms.CharField(label = “value2”, required = False, widget = forms.TextInput(attrs = {‘class’ : “form-control”}))*

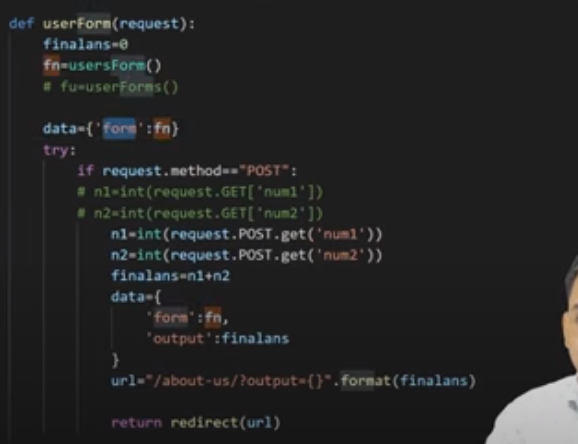
(**forms.Form):**

* It indicates that the UsersForm class is inheriting from another class called forms.Form.
* In Python, classes can inherit properties and behaviors from other classes, which is known as inheritance.
* In this case, UsersForm is inheriting from the forms.Form class, which suggests that it might be a form-related class defined in the forms module.

We have used a CharField over here so that a textbox will appear

In views.py import this forms.py file by writing

*From .forms import UsersForm*



Then in html file call the form by writing {{form}}

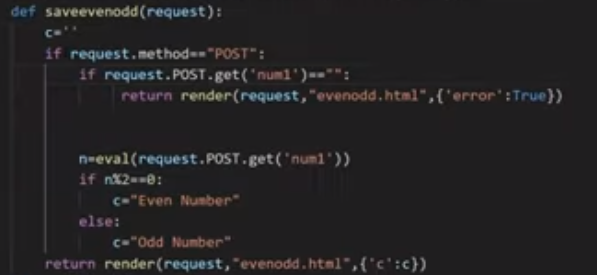
N1 = eval(request.POST.get(‘num1’) → here eval is used so that it can handle int as well as float data

**Manual Form Validation** (It means how to handle things when we submit an empty form)

i*f request.POST.get(‘num1’) == “”;*

*return render (request, ‘xyz.html’, {‘error’ : True})*

Then write if condition in .html file



A picture containing text, screenshot

Description automatically generated

**Models in our Django applicaiton**

*Python manage.py startapp xyz*

Startapp is for models. By executing this command it have a file inside named as models.py.

Xyz naam ki ek app (jo admin mein option de like add, view, delete sabka) banaya hai.

Admin.py mein hum log table ke konse features admin ko dikhaana chahte hai

And then once we are done with making the table then we will write these command in the terminal

*Python manage.py makemigrations*

*Python manage.py migrate*

Now inside models.py write this piece of code

Id will be automatically generated (it will be a primary key)

*From Django.db import models*

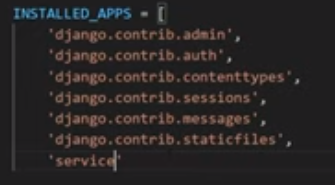
*Class Service(models.Model):*

*Service\_icon = models.CharField(max\_length = 50)*

*Service\_title = models.CharField(max\_length = 50) # max\_length is necessary to give in CharField*

*Service\_desc = models.TextField()*

Now go inside settings.py and inside installed\_apps write ‘service’



Now write the commands for makemigerations and migrate in terminal

Models abhi create hua hai (after executing the makemigrations command). Abhi who table mein convert nahi hua hai, to do so write migrate command

But after doing these we didn’t get an option in admin.

So go to admin.py and then write the following code

A picture containing text, screenshot, font

Description automatically generated

**Get all table data in Django**

Go to views and import our model by writing

*From service.models import Service*

*serviceData = Service.objects.all()*

to check the data which will be stored in serviceData we will use a for loop

*for a in serviceData:*

*print(a)*

A screen shot of a computer program

Description automatically generated with low confidence

A picture containing text, human face, person, person

Description automatically generatedA screen shot of a computer

Description automatically generated with low confidence

**Django order\_by query set, ascending and descending**

A screen shot of a computer code

Description automatically generated with low confidence

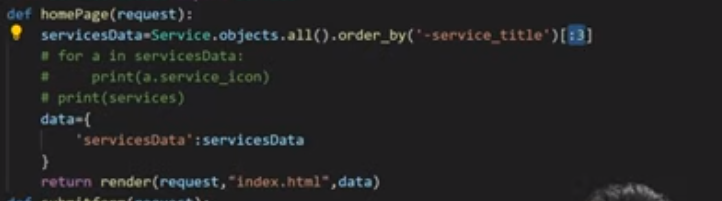
Here indicates for descending order (inside order\_by)

A screen shot of a computer code

Description automatically generated with low confidence

‘-’ Before column name means descending order and without ‘-’ means ascending order

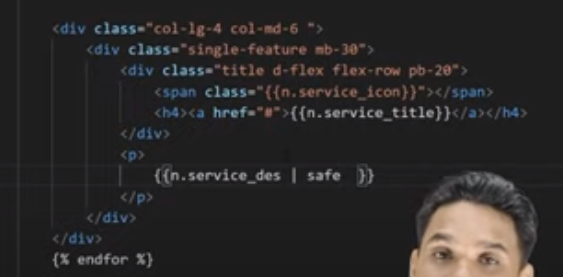
**Django limiting query results**

Use list slicing but it does not support negative indexes

**Template filters in Django**

Jaise humne table ke ek column mein lets say the name of that column is description and isme humne ek bold tag use kiya hai. So when displaying it on our html page we want the bold text to display properly. For these things we have template filters in Django

Use *safe* filter

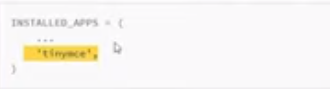


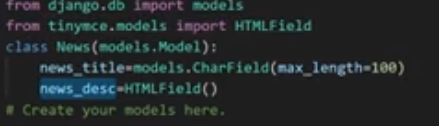
**Django TinyMCE Editor integration with news app**

We will have to install a python package for this:

*Pip install django-tinymce*

Add tinymce to INSTALLED\_APPS in settings.py





TinyMCE editor → Here for description an interface similar to word (where we can style our text and all) will be there and content written like that will be displayed in the same manner on our html page

**Display news in marquee tag**

&nbsp → used for spaces in html

A picture containing text, screenshot, font

Description automatically generated

*Onmouseover = “stop()”* and *onmouseout = “start()”* are functions of javascript (to be used alongside html)

A picture containing text, screenshot, font

Description automatically generated



A picture containing text, screenshot, font, line

Description automatically generated

**How to reset Django admin password**

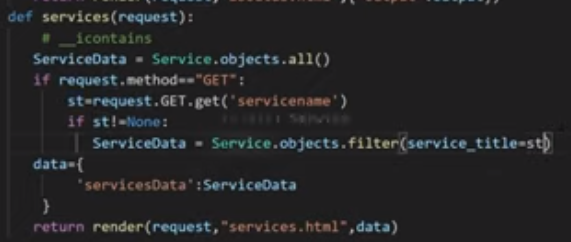
The username is stored in db.sqlite3. Use db browser and go to auth\_user. There we can find the username

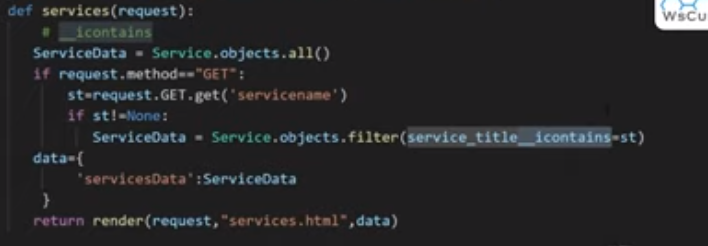
Now type this in terminal

*Python manage.py changepassword username*

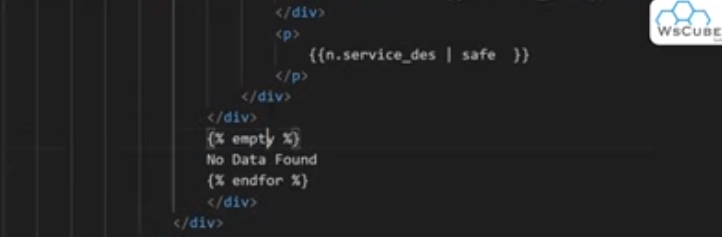
**How does filter work in Django**

By writing this piece of code the typed word should exactly match the title



Now if we want that the typed key should produce results even if it is not fully matched the service title then,

If the text does not match with any data, then no record found should be displayed. To do this:



**Django AutoSlugField**

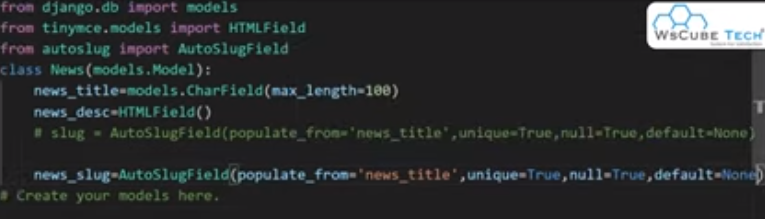
Jo bhi title, wagerah hai unko slug mein represent karna hai

We will have to install the package

*Pip install django-autoslug*

Then in models.py write

From autoslug import AutoSlugField



Then write the 2 migration statements

A screen shot of a computer code

Description automatically generated with low confidenceA picture containing text, screenshot, font

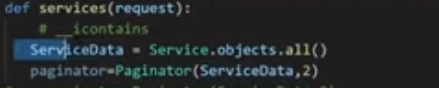
Description automatically generated

**Pagination in Django project**

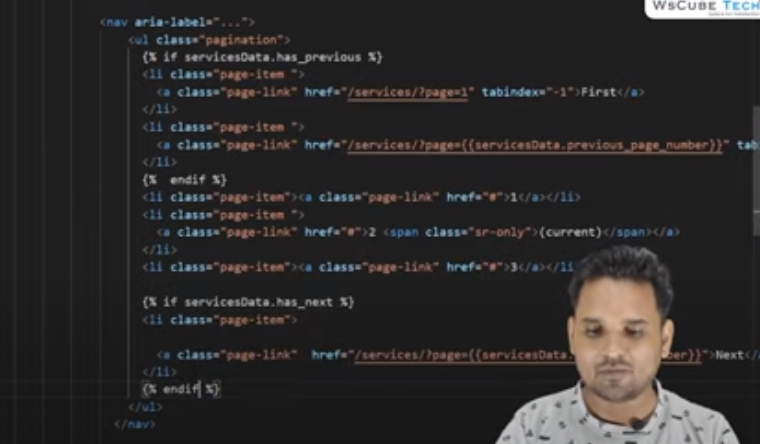
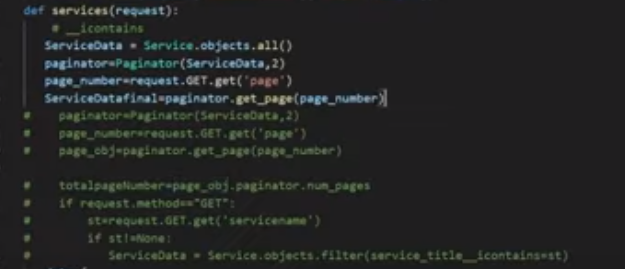
we will have to import a library for this

from Django.core.paginator import Paginator

Here 2 describes no of visible elements on a page



Copy pagination design from bootstrap



**Saving form data to the database (Model)**

**File upload in Django model in admin panel FileField()**

Write this in settings.py

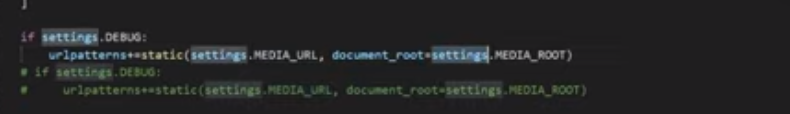


import these 2 libraries in urls.py

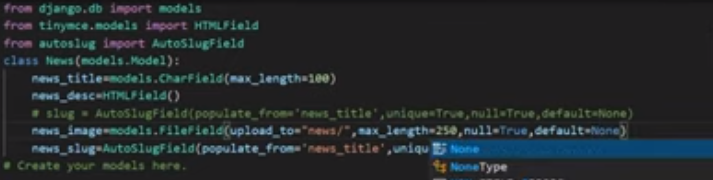
*from django.conf import settings*

*from django.conf.urls.static import static*

Here static is a function

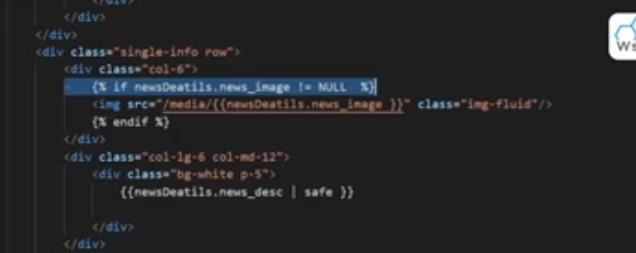
Now write this in urls.py

Now write this inside news ke models.py mein



Then write the migrations command

**Display uploaded image in template**



**Send email in Django project**