**Video file 65**

**Code file underconst project**

**One Click Website Under Maintenance Project in Django 5**

Middlewares.py:

from django.shortcuts import render

class UnderConstructionMiddleware:

    def \_\_init\_\_(self,get\_response):

        self.get\_response = get\_response

        print("One time Initialize")

    def \_\_call\_\_(self, request):

        print("THis is before view")

        response = render(request,'uc/underc.html')

        print("this is after response")

        return response

settings.py:

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

    'uc.middlewares.UnderConstructionMiddleware',

]

First go in to the settings disable our custom middlwware

We want , we can set the middleware on or off through admin panel so for that how we can do that

Than create super user

And install :

pip install python-decouple

after than go into the uc/models.py

uc/models.py:

from django.db import models

# Create your models here.

class UnderConstruction(models.Model):

    is\_under\_construction = models.BooleanField(default=False)

    uc\_note = models.TextField(blank=True,null=True,help\_text="Note for Under Construction mode")

    uc\_duration = models.DateTimeField(blank=True,null=True,help\_text="End data and time  for Under Construction mode")

    updated\_at = models.DateTimeField(auto\_now=True)

    def \_\_str\_\_(self):

        return f"Under construction: {self.is\_under\_construction}"

and than makemigrations

and go into the admin.py

admin.py:

from django.contrib import admin

from uc.models import UnderConstruction

# Register your models here.

@admin.register(UnderConstruction)

class UnderConstructionAdmin(admin.ModelAdmin):

    list\_display=['id','uc\_note','uc\_duration','is\_under\_construction']

    fields = ('uc\_note','uc\_duration','is\_under\_construction')

and again makemigrations and makemigrate

now here we need to write logic like if is\_under\_construction is true than show the under construction page else go into the view

from django.shortcuts import render

from uc.models import UnderConstruction

class UnderConstructionMiddleware:

    def \_\_init\_\_(self,get\_response):

        self.get\_response = get\_response

        print("One time Initialize")

    def \_\_call\_\_(self, request):

        try:

            uc = UnderConstruction.objects.first()

            if uc and uc.is\_under\_construction:

                response = render(request,'uc/underc.html')

        except:

            pass

but if we write this and active the is\_under\_construction so it cannot able any page or url eve admin

so .

    def \_\_call\_\_(self, request):

        if request.user.is\_staff:

            return self.get\_response(request)

        try:

            uc = UnderConstruction.objects.first()

            if uc and uc.is\_under\_construction:

                return render(request,'uc/underc.html')

        except:

            pass

        return self.get\_response(request)

and now how we show data :

in middlewares.py:

    def \_\_call\_\_(self, request):

        if request.user.is\_staff:

            return self.get\_response(request)

        try:

            uc = UnderConstruction.objects.first()

            context = {

                'uc\_note':uc.uc\_note,

                'uc\_duration':uc.uc\_duration

            }

            if uc and uc.is\_under\_construction:

                return render(request,'uc/underc.html',context)

        except:

            pass

        return self.get\_response(request)

and in template we can show that.

BUT THE ISSUE IN THAT FOR NEW USER DURING LOGIN EVEN THEY ARE ADMIN IT ALWAYS SHOW UNDERCONSTRUCTION PAGE BECAUSE EVERY USER CAN ACCESS THROUGH URL NOT A STAFF MEMBER:

SO FOR THAT WE SET THE KEY

MAKE .env file

Underconst/.env:

# Under Maintenance Key

MAINTENANCE\_BYPASS\_KEY=123

Uc/Middlewares.py:

from decouple import config

def \_\_call\_\_(self, request):

        if request.user.is\_staff:

            return self.get\_response(request)

        UC\_key = config("MAINTENANCE\_BYPASS\_KEY") # here pass maintainance key

        if 'u' in request.GET and request.GET['u'] == UC\_key:

            return self.get\_response(request)

        try:

            uc = UnderConstruction.objects.first()

            context = {

                'uc\_note':uc.uc\_note,

                'uc\_duration':uc.uc\_duration

            }

            if uc and uc.is\_under\_construction:

                return render(request,'uc/underc.html',context)

        except:

            pass

        return self.get\_response(request)

we can gave the pass key for some users for testing our website but one issue

user hits the url and go into another page they also shows the undercontruction page. They will compulsery to gave the passkey at any page url so how we resolve that

we use the sessions to overcome that :

middlewares.py:

    def \_\_call\_\_(self, request):

        if request.user.is\_staff:

            return self.get\_response(request)

        UC\_key = config("MAINTENANCE\_BYPASS\_KEY")

        if 'u' in request.GET and request.GET['u'] == UC\_key:

here we set the session

            request.session['bypass\_maintenance'] = True

            request.session.set\_expiry(0) # set expiery after page exit

    here we get the session

        if request.session.get('bypass\_maintenance',False):

            return self.get\_response(request)

        try:

            uc = UnderConstruction.objects.first()

            context = {

                'uc\_note':uc.uc\_note,

                'uc\_duration':uc.uc\_duration

            }

            if uc and uc.is\_under\_construction:

                return render(request,'uc/underc.html',context)

        except:

            pass

        return self.get\_response(request)