------------------------------------------------------------------

اول مره على الجهاز او الوندز لتحميل البيئه الافتراضيه

pip install virtualenv

=======================================================================

------------------------------------------------------------------

خطواط عند انشاء كل مشروع

------------------------------------------------------------------

1- نعمل بيئه افتراضيه خاصه بهاذا المشروع

<

نتجه الى الملف الى فيه المشروع

cd d:/code/learn/django

انشاء البيئه الافتراضيه

virtualenv test

طريقه اخرى لانشاء البيئه الفتراضيه

python -m venv test

>

2- تفعيل البيئة الافتراضيه

<

source test/Scripts/activate

cap كبتل Scripts فى s ملحوظه هامه حرف ال

>

3- هنحمل Django

<

pip install django

>

4- ندخل داخل البيئه الافتراضيه

<

cd test

>

5-انشاء المشروع

<

django-admin startproject first\_project

>

6- فتح المشروع

<

manage.py نخش جوه الملف الى فيه

cd first\_project

كود التشغيل

python manage.py runserver

ناخد الينك الى هيطلع نحطه فى البراوزر

ممكن يحصل خطا و يطلب منك تدخل كود معين مثل التالى

python manage.py migrate

ثم تقوم بكتابه كود التشغيل مره اخرى

نفتح محرر الاكواد

code .

ال . فى النهايه تعنى الملف الى انا فيه

بعد كده كل مره هفتح السيرفر ب نفس

python manage.py runserver

>

===================================================================

انشاء التطبيقات داخل المشروع

---------------------------------

manage.py اي تطبيق يتم انشائه من خلال

لازم نكون جوه صفحه المشروع و بمحازه ال manage.py

<

python manage.py startapp pages

>

settings.py بعد انشاء اى ابليكيشن لازم تسجله جوه المشروع فى ال

installed\_apps

<

'pages.apps.PagesConfig',

اسم التطبيق ثم دوت ابس ثم دوت اسم التطبيق اول حرف كبتل و نكتب وراه منغير فواصل كونفيج اول حرف كبتل ثم نغلق الكوتيشن ثم فصله

>

بتاعت المشروع urls.pyثم نقوم بتسجيله فى

<

نضيف على التالى

from django.urls import path

نجعله

from django.urls import path, include

ثم

نضيف على التالى

urlpatterns = [

path('admin/', admin.site.urls),

]

ليصبح

urlpatterns = [

path('admin/', admin.site.urls),

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

]

حيث علامه التنصيص الفضيه تجعل التطبيق يظهر عند اول صفحه للمشروع لاكن اذا ارت تغير المسار تكتب اسم الى عايز تضيفه على المسار

include و بنكتب بداخل الكوتيشن الى بعد

urls اسم الاب ثم دوت

ثم ننشئ بداخل الاب

urls.py ملف باسم

>

views.py ال

الخاصه ب الاب

from django.http import HttpResponse

ثم ننشئ فنكشن

def index(request):

return HttpResponse('hello world')

urls.py فى ال

الخاص بالتطبيق

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index')

]

======================================================================

templates انشاء التيمبليت

----------------------------------

انشاء ملف فى مستوى المشروع و الابليكيشن باسم

templates

مع مراعاة ان تكتب الاسم كما هو بالضبط

فى ملف المشروع نفتح

settings.py

و نعمل استدعاء لمكتبه

import os

و هذه المكتبه تساعد فى عدم حدوث اخطاء عند ربط التيمبلت

و فى نفس الصفحه هناك قائمه باسم

templates

بداخلها قاموس به مفتاح باسم

dirs.

ندخل قيمه لهاذا المفتاح

داخل ملف ال

templates

ننشأ فولدر لكل تطبيق و بداخله ننشا صفحه ال

html

ثم ننتقل الى الفيو

views.py

و نستدعى

from django.shortcuts import render

و نقوم بتعديل ال

function

او عمل واحد جديد

def index(request):

return render(request, 'pages/index.html')

==================================================================

dtl

-----

المتغيرات

-------

فى صفحه ال

views.py

للطتبيق

نقوم بتعديل ال

fanction

باضافه براميتر ثالث لل

return render

و يكون بها قاموس باسم و قيمه المتغير

سواء بادخال القاموس مباشره او كتابته بجسم المعادله و التعويض عنه بمتغير

<

def index(request):

the\_var = {'name':'abdo', 'age':25}

return render(request, 'pages/index.html', the\_var)

>

و فى صفحه ال

html

الخاصه بالتامبلت للتطبيق

templates - index.html

<

name = {{name}}

age = {{age}}

>

-------

filters الفلاتر

<

name = {{name|length}}

>

----------

tag تاج

+

extends الوراثه

----

ننشأ صفحه

html

فى الtemplates

و نعمل فيها الحجات الى انا عايز اورثها

و فى الصفحه الى انا شغال فيها و بتتعرض هضيف تاج التوريث

<

{% extends 'base.html' %}

>

لاكن عند الوراثه اى

html

مش هيبان غير لما احطه فى

block

عنطريق

الذهاب الى صفحه

html

الى بنورث منها و نضيف ال block

كالتالى

<

{% block any\_text %}

{% endblock any\_text %}

>

ثم الذهاب الى الصفحه الى بنشتغل فيها و بنعرضها و نضيف ال

html

جوه نفس التاج كالتالى

<

{% block any\_text %}

<h1>

about

</h1>

{% endblock any\_text %}

>

===============================================================================

التضمين

include

مثل

footer

و

nav\_bar

-----------

templates نعمل

و يستحسن جوه

folder

اسمه

parts

و نسميها

nav\_bar.html

footer.html

و نكتب جواهم اكواد ال

html

الى بنورث منها base.html بعد كده نذهب الى صفحه ال

ونحط

tag

ال

include

فى المكان المناسب فوق ال

block

الخاص بالمحتوى او تحته حسب هو

footer او nav\_bar

او حتى فى اى مكان يعجبنى

=========================================

if استخدام

----------------

{% if name == 'abdo' %}

<h1>

this message appears because if tag return true

</h1>

{% elif name == "ahmed" %}

<h1>

this message appears because elif tag return true

</h1>

{% else %}

<h1>

this message appears because if and elif tags is false

</h1>

{% endif %}

=================================================

for استخدام

-----------------

{% for x in name %}

<p>loob to print my name chr ( {{x}} )</p>

{% endfor %}

================================

الاستاتيك static

جوه فولدر المشروع نعمل فولدر ال static

و جوه فولدر static نعمل فولدر لكل غرض مثل css, html, js, image

ثم لازم تفعل فولدر ال static فى sittings.py

بداخل sittings.py

نكتب فوق STATIC\_URL

STATIC\_ROOT = os.path.join(BASE\_DIR, ‘static’)

مع ملاحظه ان ما تحته خط caps

و نكتب تحت STATIC\_URL

STATICFILES\_DIRS = [

مسار الفولدر

os.path.join(BASE\_DIR, ‘project\_name/static’)

]

ثم و انا داخل فيل المشروع نخش على ال terminal و نكتب

Py manage.py collectstatic

ثم فى الصفحه الى هربط بيها الفيل css الى انا عيزه هعمل

{% load static %}

بعد كده تاج html الخاص بالربط لل css او بالصوره لاكن فى خانه المسار بنحطه فى وسم tag {%%}

<link rel=”stylesheet” href=”{% static ‘css/style.css’ %}”>

التعليق comment

{% comment %}

نكتب اى كمنت هنا مش هيزهر

{% endcomment %}

اللينك <a>

زى ما بنعمله فى html عادى لاكن فى href بنعمله جوه tag

<a href=”{% url ‘name\_of\_the\_page %}”>

انشاء يوزر و باصورد

Create username and password

اقفل ال server عنطريق control + c

ثم كتابه داخل المشروع الى جواه manage.py

Python manage.py createsuperuser

الاسم اليوزر نيم abdoallrk username

الرقم السرى باسورد password 4526379924

Data البيانات

-----------------

فى صفحه ال models.py داخل التطبيق

ننشئ class و هذا ال class هيكون جدول للبيانات

class Car(models.Model):

    name = models.CharField(max\_length=100)

    price = models.DecimalField(max\_digits=6, decimal\_places=2)

    content = models.TextField()

    image = models.ImageField(upload\_to='photos/%y/%m/%d')

    active = models.BooleanField(default=True)

/////////////////

بعد كده نضيف الدادا بتعتى على صفحه ال admin.py بتاعت نفس التطبيق الى فيه الدادا فى models.py

عنطريق 1- نعمل import لل class

from .models import Product

و ناخد بالنا من اسم ال class لازم يكون مطابق و اول حرف cap

2- نعمل register

admin.site.register(Product)

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

بعد كده نضيف الجدول بتاعى فى قواعد البيانات

جوه صفحه المشروع بمحازاه manage.py

1 – نوقف ال server

2 – نعمل migrations   
 < python manage.py makemigrations >

3- نعمل migrate

< python manage.py migrate >

عرض البيانات فى الموقع   
------------------------

ننشأ صفحه html خاصه بالمنتج و اخرى خاصه بعرض كل المنتجات و ذلك داخل فولدر خاص بهم داخل فولدر ال templates

* داخل صفحه ال html لا تنسى ان تقوم بتوريث الصفحه الام الى فيها الشكل العام الى انت عايز كل صفح الموقع تبقى شبهه
* داخل صفحه ال html لا تنسى ان تكتب اكواد html داخل وسم tag الخاص ب block ملحوظه اسم هذا ال block انت الى قومت باختياره مسبقاً
* {% extends 'base.html' %}
* {% block any\_text %}
* {% endblock any\_text %}
* بعد كده نعرض الصفحه html الى احنا عملناها عنطريق function داخل ال views.py الخاصه بالتطبيق بتاع المنتجات products

def product(request):

    return render(request, 'products/product.html')

* طبعا فى ال urls.py اخاصه بالمشروع احنا كنا ربطين فيها الينك الخاص ب urls.py الخاصه بال app
* from django.contrib import admin
* from django.urls import path, include
* urlpatterns = [
* path('admin/', admin.site.urls),
* path('pages/', include('pages.urls')),
* path('products/', include('products.urls')),
* ]
* دلوقتى urls.py الخاصه بال app هنربط فيها الينك الخاص بعرض الصفح الى فى ال views.py

from django.urls import path

from . import views

urlpatterns = [

    path('', views.products, name='products'),

    path('product', views.product, name='product'),

]

* بعد كده نبدأ بتحديد ما سوف يتم عرضه من البيانات data
* نروح لل views.py الخاصه بال app و نعمل استدعاء لل models الخاص بال app و نزود نعدل على الfunction الى بتعرض الصفح عنطريق اضافه argument ثالث لل render و يكون dictionary ليه key و value -- ال key اى كلمه معبره و ال value تكون ال class بتاع المنتجات مضاف اليه ال objects او object + باقى الاضافات
* from django.shortcuts import render
* from .models import Product
* # Create your views here.
* def products(request):
* return render(request, 'products/products.html', {'pro':Product.objects.all()})
* def product(request):
* return render(request, 'products/product.html')
* بعد كده نروح على templates و نختار ال html الخاصه بصفحه المنتج الى هعرضه و مبين tag ال block نكتب كود الى هنعرض بيه البيانات الى فى key الخاصه بال dictionary الى عملناها فى views
* {% extends 'base.html' %}
* {% block any\_text %}
* {% for x in pro %}
* <h1>{{x.name}}</h1>
* {% endfor %}
* {% endblock any\_text %}
* فى صفحه ال admin لو انا عايز كل منتج بدل ما يظهر class.object يظهر اسمه . نزهب الى صفحه ال models.py الخاصه لل app و نعدل على class المنتج ليصبح كالتالى

from django.db import models

class Product(models.Model):

    name = models.CharField(max\_length=100)

    price = models.DecimalField(max\_digits=6, decimal\_places=2)

    content = models.TextField()

    image = models.ImageField(upload\_to='photos/%y/%m/%d')

    active = models.BooleanField(default=True)

    def \_\_str\_\_(self):

        return self.name

* فى صفحه ال admin لو انا عايز ال class بتاعى يظهر باسم معين . نزهب الى صفحه ال models.py الخاصه لل app و نعدل على class المنتج ليصبح كالتالى
* from django.db import models
* class Product(models.Model):
* name = models.CharField(max\_length=100)
* price = models.DecimalField(max\_digits=6, decimal\_places=2)
* content = models.TextField()
* image = models.ImageField(upload\_to='photos/%y/%m/%d')
* active = models.BooleanField(default=True)
* def \_\_str\_\_(self):
* return self.name
* class Meta:
* verbose\_name = 'phone'
* لو عايز ارتب المنتجات الى معروظه وفقاً للسعر على سبيل الميثال
* فى نفس الصفحه الى اشتغلنا عليها فوق هنعدل كالتالى

from django.db import models

class Product(models.Model):

    name = models.CharField(max\_length=100)

    price = models.DecimalField(max\_digits=6, decimal\_places=2)

    content = models.TextField()

    image = models.ImageField(upload\_to='photos/%y/%m/%d')

    active = models.BooleanField(default=True)

    def \_\_str\_\_(self):

        return self.name

    class Meta:

        verbose\_name = 'phone'

        ordering = ['price']

لو عايز الترتيب يكون تنازلى نحط سالب ( - ) قبل الكلمه

لو عايز قيمه افتراضيه default او خانه ليست اجبارى ان يتم ملئها او اغير اسم العنوان للبيانات الى هتخش منغير ما تتغير فى قاعده البيانات

from django.db import models

class Product(models.Model):

    name = models.CharField(max\_length=100, default='the phone name', verbose\_name='title' )

    price = models.DecimalField(max\_digits=6, decimal\_places=2)

    content = models.TextField(null=True, blank=True, verbose\_name="description")

    image = models.ImageField(upload\_to='photos/%y/%m/%d')

    active = models.BooleanField(default=True)

    def \_\_str\_\_(self):

        return self.name

    class Meta:

        verbose\_name = 'phone'

        ordering = ['-price']

* لعمل قائمه منسدله و نختار منها
* جوه ال class نعمل list بالاختيارات
* كل قيمه فى list هتكون عباره عن table () مكون من قمتين الاولى الى هتظهر و الثانيه الى هتكون فى الداتا
* ثم فى ال class نضيف عنصر زى ال name كده و اديه اسم معبر و نحط داخل الارجيو منتس بتاعته choices= name of list كالتالى و خد بالك بعد ما تضيف fild جديد فى الجدول اى class لازم تعمل ماى جراشن + ماى جراد
* from django.db import models
* class Product(models.Model):
* x = [
* ('phone','phone')
* ('computer','computer')
* ]
* name = models.CharField(max\_length=100, default='the phone name', verbose\_name='title' )
* price = models.DecimalField(max\_digits=6, decimal\_places=2)
* content = models.TextField(null=True, blank=True, verbose\_name="description")
* image = models.ImageField(upload\_to='photos/%y/%m/%d')
* active = models.BooleanField(default=True)
* category = models.CharField(max\_length=50, null=True, blank=True, choices=x)
* def \_\_str\_\_(self):
* return self.name
* class Meta:
* verbose\_name = 'phone'
* ordering = ['-price']
* فى ال terminal
* python manage.py makemigrations
* python manage.py migrate
* بعد كده runserver
* لو عايز اعمل الوقت و التاريخ الحالى بنروح لل models.py و نعمل import كالتالى ثم نستخدمه كالتالى
* from datetime import datetime
* class Test(models.Model):
* date = models.DateField()
* time = models.TimeField(null=True)
* date\_and\_time = models.DateTimeField(null=True)
* on\_created = models.DateTimeField(default=datetime.now)

عرض عنصر معين filter

* فى صفحه ال views.py الخاصه ب ال app نخش على ال function الخاصه بعرض الصفحه و نعدل عليها باضافه بعد Product.objects.all() نحط .filter(price=100) على سبيل المثال التالى
* def products(request):
* return render(request, 'products/products.html', {'pro':Product.objects.all().filter(price=5)})

البحث عن منتج اسمه به حرف معين .filter(name\_\_contains=’s’)

def products(request):

    return render(request, products/products.html',{'pro':Product.objects.all().filter(name\_\_contains='s')})

البحث عن سعر من الاسعار التاليه 100 و 500 و 800على سبيل المثال

.filter(price\_\_in=[100, 500, 800]

def products(request):

    return render(request, 'products/products.html', {'pro':Product.objects.all().filter(price\_\_in=[5, 2000])})

البحث عن الاسعر بين الاسعار التاليه من 100 الى 500

.filter(price\_\_range=(100,500)

def products(request):

    return render(request, 'products/products.html', {'pro':Product.objects.all().filter(price\_\_range=[5, 1500])})

الترتيب

سابقاً كنا اتعلمنا اذاى نرتب لاكن عنطريق صفحه models.py بس دى مش افضل طريقه لانه بيعمل الترتيب على كل الصفح مره وحده

ممكن نرتب كل صفحه على حدى من خلال صفحه ال views.py عنطريق تعديل ال function الى Product.objects.all().order\_by(price)

كما هو بالمثال التالى

def products(request):

    return render(request, 'products/products.html', {'pro':Product.objects.all().order\_by('price')})

و ممكن نعد عدد المنتجات عنطريق .count()لاكن لازم نحول الناتج الى string

لاكن فى هذه الحاله سوف نعرض قيمه ال key مباشره بدون loop

def products(request):

    return render(request, 'products/products.html', {'pro':str(Product.objects.all().count())})

{{pro}}

الاستثناء

عرض كل حاجه معادى شئ معين

.exclude(price=5)

def products(request):

    return render(request, 'products/products.html', {'pro':Product.objects.all().exclude(price=5)})

عرض الصور

----\\\\\

1. لازم نحط الصوره فى تاج ال html الخاص بالصور كالتالى

<img src="{{x.image.url}}” alt=””>

2 - لازم نفعل ال media - عنطريق الذهاب الى ملف ال settings.py جوه فولدر المشروع – ثم تحت عند الكمنت الخاص ب static files (css, javascript, images)

3 – نقوم بكتابه الاكواد التاليه

# Media

MEDIA\_ROOT = os.path.join(BASE\_DIR, ‘media’)

MEDIA\_URL = ‘/media/’

4 – ثم فى نفس الفولدر الخاص بالمشروع نذهب الى ملف urls.py و نقوم باستدعاء المكاتب التاليه

from jango.conf import settings

from jango.conf.urls.static import static

5- ثم فى نفس الملف نقوم بالضافه الى ال list المسمى urlpatterns التالى

urlpatterns = [

    path(‘admin/’, admin.site.urls),

    path(‘pages/’, include(‘pages.urls’)),

    path(‘products/’, include(‘products.urls’)),

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

6 – الصور الى هتظهر هتكون بتاعت المنتجات الجديده فقط التى تم انشائوها بعد كل هذه الخطوات و ليس القديمه

ناخد البيانات من المستخدم

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\\\

**الطريقه الاولى**

1. نقوم بتكوين ال templates

* و نعمل فورم عادى خالص ب html باستخدام <form>
* فى حاله حمايه البيانات بواسطه ال post نقوم باضافه تاج للحمايه خاص ب Django كالتالى {% csrf\_token %}
* و يكون الشكل النهائى
* {% extends 'base.html' %}
* {% block any\_text %}
* <form method="post">
* {% csrf\_token %}
* <br>
* <input type="text" name="username" >
* <br>
* <input type="password" name="password" >
* <br>
* <input type="submit" value="save">
* </form>
* {% endblock any\_text %}

1. نقوم بتكوين ال models

* نستدعى from Django.db import models ثم نقوم بصنع ال class اى الجدول ثم makemigrations و migrate
* from django.db import models
* class Login(models.Model):
* username = models.CharField(max\_length=50)
* password = models.CharField(max\_length=50)
* python manage.py makemigrations
* python manage.py migrate

3 – ننتقل الى صفحه ال views.py الخاصه بال app

* يتم استلام البيانات المدخله من المستخدم عنطريق ال request الموجوده ك ارجيومنت لل function ثم type الخاص بال form ثم get للحصول على value الخاص ب key الى هنكتبه ثم حفظ الكلام ده فى var منطقى
* نريد اضافه تلك البيانات بالجدول ... نقوم باستدعاء الجدول from .models import Login ثم نقوم بصنع object لنضع به البيانات
* from django.shortcuts import render
* from .models import Login
* # from django.http import HttpResponse
* def about(request):
* username = request.POST.get('username')
* password = request.POST.get('password')
* data = Login(username = username, password = password)
* data.save()
* return render(request, 'pages/about.html')
* ثم نذهب الى ال admin.py ليظهر هذا الجدول فى صفحه ال admin بالويب

from django.contrib import admin

from .models import Login

# Register your models here.

admin.site.register(Login)

=====================\\\\\\\\\\\ \\\\\\\\

**الطريقه الثانيه**

**---\\\**

بتعتمد على ان ال Django ان هو الى هيعمل ال فورم

* هنروح ل folder ال app و نعمل file.py جديد نسميه على سبيل المثال form.py و جواه نقوم باستدعاء المكتبه بتاعت ال form ثم نقوم بصنع class لجدول البيانات
* from django import forms
* class LoginForm(forms.Form):
* username = forms.CharField(max\_length=50)
* password = forms.CharField(max\_length=50)
* احنا عيزين نرسل البيانات لل templates علشان كده هنروح لل views.py و هنعمل import لل class بتاع الجدول الى عملناه فى فى ال forms
* و فى ال views.py نظبط ال function الخاصه بعرض الصفحه عنطريق تعديل ال return render و نزود برامتر ثالث به key المناسب لك و value و هيكون ال class الى فيه اليانات الى هرجعها
* from django.shortcuts import render
* from .models import Login
* from .forms import LoginForm
* # from django.http import HttpResponse
* def about(request):
* username = request.POST.get('username')
* password = request.POST.get('password')
* data = Login(username = username, password = password)
* data.save()
* return render(request, 'pages/about.html', {'lf':LoginForm})
* و فى ال templates مش هنكون محتاجين نعمل input text و هنعمل مكنهم ال متغير الى هو ال key الى عملناه فى function الخاص ب ال views.py
* {% extends 'base.html' %}
* {% block any\_text %}
* <form method="post">
* {% csrf\_token %}
* {{lf}}
* <br>
* <input type="submit" value="save">
* </form>
* {% endblock any\_text %}

**\\\\===============**

**الطريقه الثالثه**

**------\\\\\\\\**

بدل مكنا بندخل فى ال forms كل البيانات واحد واحد هناخد البيانات الى فى ال models.py على طول نعمل import لل class الموجود فى models ثم ننشئ ال class فى ال forms

from django import forms

from .models import Login

class LoginForm(forms.ModelForm):

    class Meta:

        model = Login

        fields = '\_\_all\_\_'

* و فى ال view.py مش بنحتاج نعمل فى ال function غير return
* from django.shortcuts import render
* from .models import Login
* from .forms import LoginForm
* # from django.http import HttpResponse
* def about(request):
* LoginForm(request.POST).save()
* return render(request, 'pages/about.html', {'lf':LoginForm})

\\\======\\\

حل مشكله ال save و الحمايه من الهكر hucker

فى صفحه ال views.py داخل ال function هنحط if من اجل تجنب تلك المشاكل

from django.shortcuts import render

from .models import Login

from .forms import LoginForm

# from django.http import HttpResponse

def about(request):

    if request.method == 'POST':

        dataform = LoginForm(request.POST)

        if dataform.is\_valid():

            dataform.save()

    return render(request, 'pages/about.html', {'lf':LoginForm})