

Django Python

# إشراف / أ.د عايض الشباطي



Django Packages

إعداد الطالب :

عمرو محمد الحاج

عبدالرحمن الحبيشي

أنس فؤاد الأسود

# What is Django Python ?





هو إطار عمل لتطوير تطبيقات الويب بلغة البرمجة Python يتميز هذا الإطار بسرعة التطوير، والأمان، والقابلية للتوسيع، مما يجعله خياراً مثالياً للمطورين الذين يرغبون في بناء تطبيقات ويب فعالة وقوية.

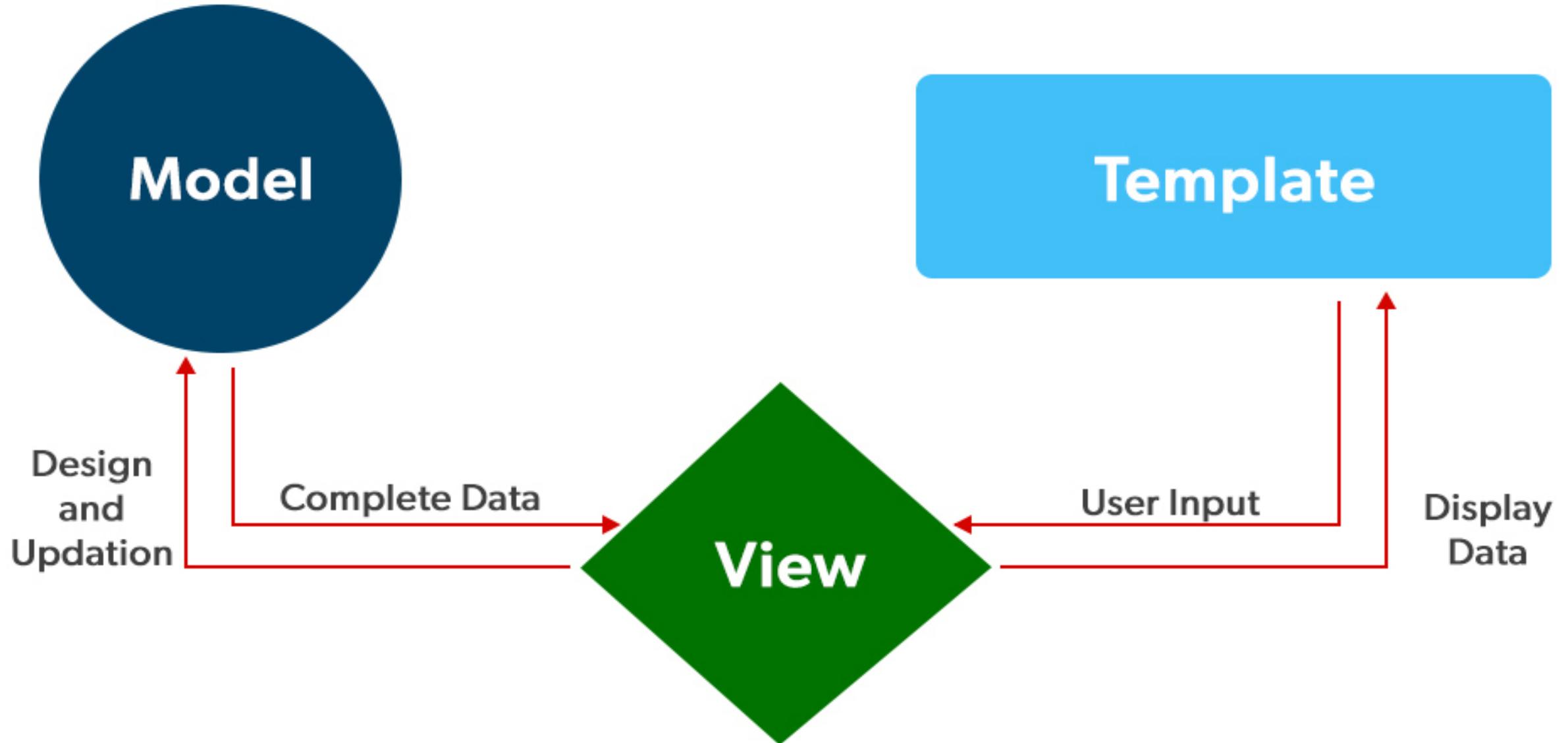
# تاریخ :Django

تم تطوير DJANGO في الأصل من قبل فريق من المطورين في صحيفة LAWRENCE JOURNAL-WORLD في عام ٢٠٠٣، وتم إصداره كمشروع مفتوح المصدر في يوليو ٢٠٠٥. سُمي الإطار نسبةً إلى الموسيقي DJANGO REINHARDT و من ذلك الوقت استمر DJANGO في النمو حتى أصبح أحد الإطارات الأكثر شعبية واستخداماً في تطوير تطبيقات الويب.

# المزايا الأساسية لـ Django

١. **سرعة التطوير:** يتيح DJANGO للمطورين كتابة تطبيقات الويب بسرعة وكفاءة عالية بفضل مجموعة الأدوات الشاملة التي يوفرها.
٢. **الأمان:** يتضمن DJANGO العديد من الميزات الأمنية المدمجة مثل الحماية من البرمجة عبر المواقع وحماية من الحقن XSS، مما يساعد على تطوير تطبيقات آمنة.
٣. **القابلية للتوسيع:** يمكن استخدام DJANGO لبناء تطبيقات صغيرة وكبيرة بنفس الكفاءة، يتميز بقدرته على التعامل مع كميات كبيرة من البيانات والمستخدمين.
٤. **ادارة قواعد البيانات:** يستخدم DJANGO نظام (OBJECT-RELATIONAL MAPPING) ORM الذي يسهل التعامل مع قواعد البيانات من خلال تحويل الكائنات البرمجية إلى جداول بيانات والعكس.
٥. **نظام إدارة الموقع:** يوفر DJANGO واجهة إدارة تلقائية يمكن استخدامها لإدارة محتوى الموقع بسهولة وبدون الحاجة إلى كتابة الكثير من التعليمات البرمجية.

# المكونات الأساسية لـ Django



# المكونات الأساسية لـ Django

١. النماذج **Models**  
تُستخدم لتعريف بنية البيانات في التطبيق، و تقوم بتحويل البيانات بين كائنات Python وجداول قواعد البيانات.

٢. العروض **Views**  
تُستخدم لمعالجة الطلبات وإرجاع الردود المناسبة للمستخدمين، يمكن أن تكون العروض وظائف بسيطة أو فئات أكثر تعقيداً.

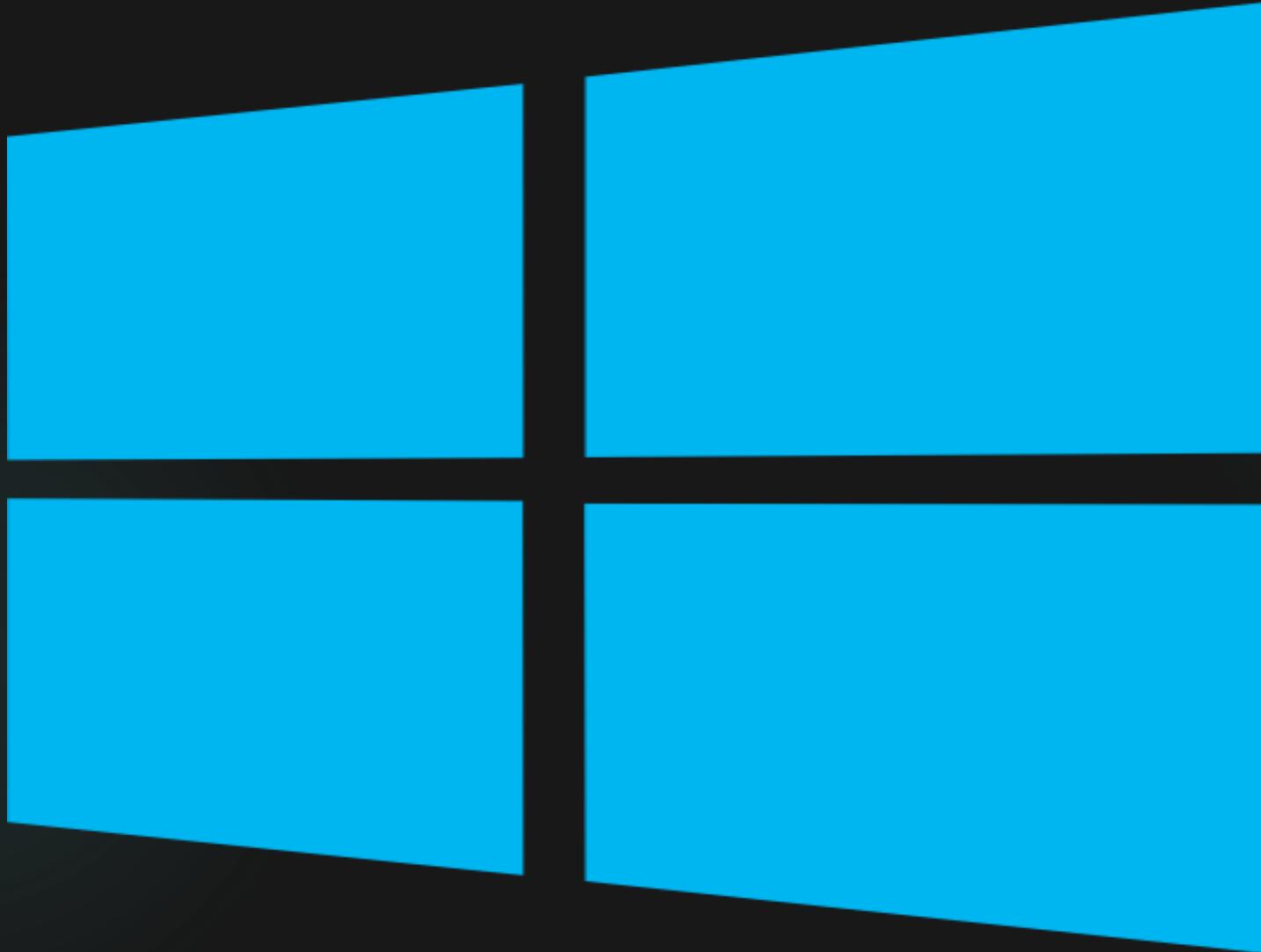
٣. القوالب **Templates**  
تُستخدم لتوليد صفحات HTML ديناميكية بناءً على البيانات التي تم تمريرها من العروض.

٤. المسارات **URLs**  
تُستخدم لتحديد الروابط التي يمكن للمستخدمين الوصول إليها وربطها بالعروض المناسبة.

# كيفية تحميل و استخدام ? Django



# أولاً: على نظام windows



## ١. تحميل وتنصيب :Python

قم بزيارة الموقع الرسمي لـ [Python](#).

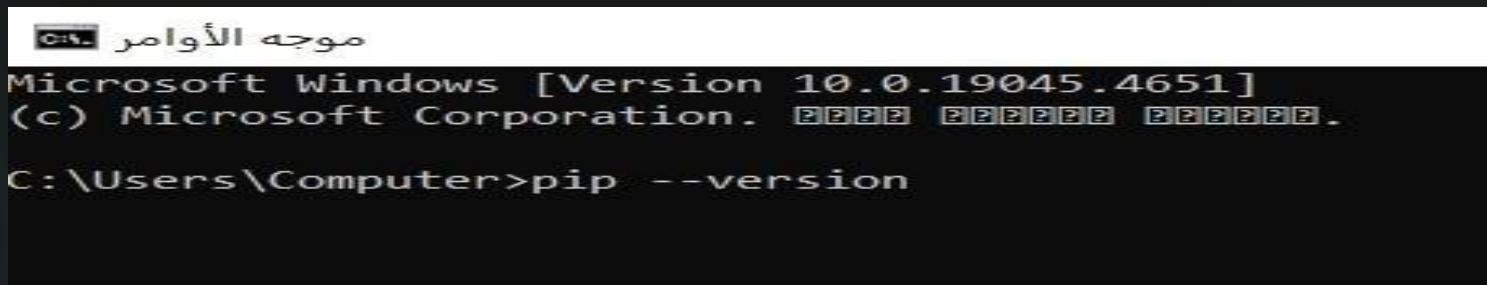
قم بتنزيل الإصدار الأحدث من [Python](#).

شغل مثبت Python وتأكد من تحديد الخيار Add Python to PATH قبل البدء في التثبيت.

**التحقق من التثبيت:**

بعد الانتهاء من التثبيت، افتح موجه الأوامر "Command Prompt".

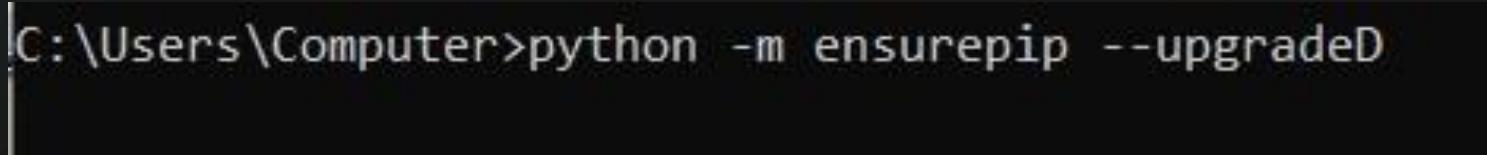
**تحقق من تثبيت pip:**



```
موجه الأوامر
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. 2021. برمجية Microsoft.

C:\Users\Computer>pip --version
```

إذا لم يكن مثبت ، يمكنك تنزيله باستخدام :



```
C:\Users\Computer>python -m ensurepip --upgrade
```

## ٢. تحميل و تثبيت :Django

تثبيت Django باستخدام :pip

```
C:\Users\Computer>pip install django
```

التحقق من تثبيت :Django

```
C:\Users\Computer>django-admin --version
```

### ٣. انشاء مشروع Django جديد:

لإنشاء مشروع جديد باسم **myproject** عبر احدى الأوامر التالية:

```
C:\Users\Computer>python -m django startproject myproject
```

أو

```
C:\Users\Computer>django-admin startproject myproject
```

أو

```
C:\Users\Computer>C:\Python312\Scripts\django-admin.exe startproject myproject
```



الانتقال الى المجلد المشروع الذي أنشئ:

```
C:\Users\Computer>cd myproject
```

٤. إنشاء تطبيق داخل المشروع:

```
C:\Users\Computer>python manage.py startapp myapp
```



## ٥. ادخل إلى المشروع ودخل الى إعداد التطبيق في الإعدادات:

قم بتسجيل تجذير التطبيق الذي أنشئت في `:settings.py`  
افتح `:INSTALLED_APPS` myproject/settings.py وأضف التطبيق إلى قائمة

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure under 'OPEN EDITORS'. It includes files for the main project ('myproject') and an application ('myapp'). The 'settings.py' file for the application is currently selected and open in the editor. The code in the editor shows the configuration for the application, including the list of installed apps. A red circle highlights the addition of the 'myapp' module to the 'INSTALLED\_APPS' list.

```
myproject > settings.py
27
28     ALLOWED_HOSTS = []
29
30
31     # Application definition
32
33     INSTALLED_APPS = [
34         'django.contrib.admin',
35         'django.contrib.auth',
36         'django.contrib.contenttypes',
37         'django.contrib.sessions',
38         'django.contrib.messages',
39         'django.contrib.staticfiles',
40         'myapp',
41     ]
```

## ٦. إعداد مجلدات الملفات الثابتة:

إنشاء مجلدات **templates** و **static**

```
C:\Users\Computer\myproject>mkdir myapp\templates\myapp
```

```
C:\Users\Computer\myproject>
```

## ٧. إضافة ملفات HTML و CSS

يتم إنشاء ملف `:myapp/static/css/style.css`

The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar:
  - OPEN EDITORS**: settings.py, # style.css
  - MYPROJECT**: myapp, migrations, static\css, # style.css, templates
- Editor Area**:
  - File tab: settings.py, # style.css (highlighted)
  - Content pane:

```
myapp > static > css > # style.css > body
1 <body> {
2   background-color: #f0f0f0;
3   font-family: Arial, sans-serif;
4 }
5 <h1> {
6   color: #333;
7   text-align: center;
8 }
```

: myapp/templates/myapp/index.html

EXPLORER    ...

> OPEN EDITORS

✓ MYPROJECT

- ✓ myapp
  - > migrations
  - ✓ static\css
    - # style.css
  - ✓ templates\myapp
    - ↳ index.html
- ↳ \_init\_.py
- ↳ admin.py
- ↳ apps.py
- ↳ models.py
- ↳ tests.py
- ↳ views.py

myapp > templates > myapp > ↳ index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>عبدالرحمن الحبيشي</title>
7      <link rel="stylesheet" href="{% static 'css/style.css' %}">
8  </head>
9  <body>
10     ^ مرحباً بك في موقعنا، نتشرف بزيارتكم</h1>
11     {% comment %} لاحقاً {% endcomment %}
12  </body>
13  </html>
```

تحديث **settings.py** للملفات الثابتة والقوالب:

EXPLORER ...

OPEN EDITORS

MYPRO... myproject > settings.py

```

54
55     # إعدادات ملفات الثابتة
56     STATIC_URL = '/static/'
57     STATICFILES_DIRS = [
58         os.path.join(BASE_DIR, 'myapp/static'),
59     ]
60
61     # إعدادات القوالب
62     {
63         'BACKEND': 'django.template.backends.django.DjangoTemplates',
64         'DIRS': [os.path.join(BASE_DIR, 'myapp/templates')],  

65         'APP_DIRS': True,  

66         'OPTIONS': {
67             'context_processors': [
68                 'django.template.context_processors.debug',
69                 'django.template.context_processors.request',
70                 'django.contrib.auth.context_processors.auth',
71                 'django.contrib.messages.context_processors.messages',
72             ],
73         },
74     },
75 ]

```

## ٩. إعداد Views و URL

كتابة في View : myapp/views.py

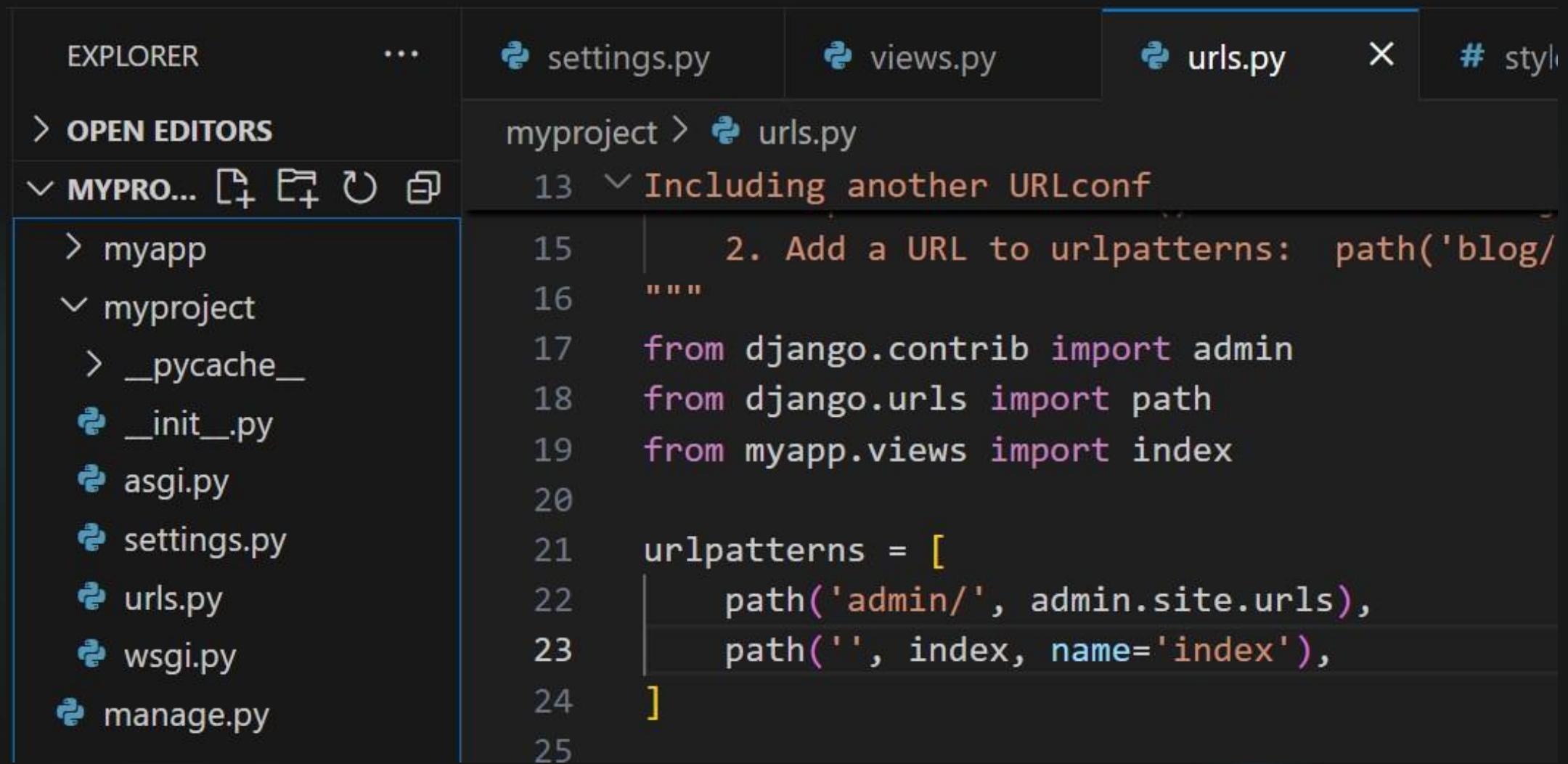
The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar showing the project structure:

- > OPEN EDITORS
- ▽ MYPROJECT
  - ▽ myapp
    - > migrations
    - > static
    - > templates
    - views.py
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
- ▽ myproject

The main editor area shows the contents of views.py:

```
myapp > views.py
1  from django.shortcuts import render
2
3  # Create your views here.
4  def index(request):
5      return render(request, 'myapp/index.html')
```

## تحديث URL في myproject/urls.py



EXPLORER    ...

> OPEN EDITORS

▽ MYPRO...

- > myapp
- ▽ myproject
  - > \_\_pycache\_\_
  - \_\_init\_\_.py
  - asgi.py
  - settings.py
  - urls.py
  - wsgi.py
- manage.py

settings.py views.py urls.py # style

myproject > urls.py

```
13 13     ▼ Including another URLconf
14 14         2. Add a URL to urlpatterns: path('blog/',
15 15             "''"
16 16
17 17     from django.contrib import admin
18 18     from django.urls import path
19 19     from myapp.views import index
20 20
21 21     urlpatterns = [
22 22         path('admin/', admin.site.urls),
23 23         path('', index, name='index'),
24 24     ]
25 25
```

1 file

## ١٠. تشغيل السيرفر:

تشغيل السيرفر المحلي:

```
python manage.py runserver
```

افتح المتصفح على <http://172.0.0.1:8000>  
لتشاهد الموقع مع القالب الذي أنشأته.

بهذه الخطوات، تكون قد أنشأت مشروع Django جديد مع واجهة  
HTML بسيطة. يمكنك الآن تعديل القوالب وإضافة المزيد من الميزات  
كما تشاء.



# طريقة لترتيب المجلدات والملفات في : Django

```
myproject/
|   myapp/
|   |   static/
|   |   |   images/
|   |   |   |   .الصور و الايقونات.jpg
|   |   templates/
|   |       myapp/
|   |           index.html
|
|   myproject/
|       settings.py
```



وهذا مثال بسيط لإضافة محتوى لنفس الصفحة في نفس القالب الذي تم إنشائه معاً في ما سبق:

The screenshot shows a web browser window with the following details:

- Page Title:** عبدالرحمن الحبيشي
- Page URL:** 127.0.0.1:8000/#
- Page Content:** The page displays the logo "Aikhtisar" with the tagline "اختصرناها لك" and a teal navigation bar with Arabic links: "من نحن", "الفنادق", "البرامج السياحية الخاصة", "الوجهات", "خدماتنا", and "القائمة الرئيسية". Below the navigation bar is a large question "ما هي الوجهة التي تريدها؟" followed by four images representing travel destinations: a resort pool, a city skyline, pyramids, and a waterfall.
- System Status Bar:** The bottom bar shows system icons and status information, including a date/time stamp (٣١/٠١/٢٠١٩ ٥:٤٤ PM), language (ENG), and developer tools like VS Code and Chrome.

ثانياً: على نظام لينكس



## أولاً: نقوم بتنبيه المتطلبات:

A screenshot of a Kali Linux terminal window titled "Player 6" in VMware Workstation 17 Player. The terminal shows the user "kali" at the prompt "kali@kali: ~". The command "\$ sudo su apt install python3" is being typed. The background of the window features a large, stylized Kali Linux logo. The top bar includes standard window controls and a toolbar with icons for file operations like copy, paste, and search.

## ٢. نقوم بتنصيب : Django

Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player 6 | II | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X | X</

### ٣. انشاء مشروع والدخول للمجلد:

The screenshot shows a Kali Linux terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal session is running as user "kali" at the prompt "kali@kali: ~/anas". The user has run the command "django-admin startproject anas", which creates a new Django project directory. The terminal then changes directory to the project folder using "cd anas". Finally, the user is back at the root directory of the project with the command "\$".

```
(kali㉿kali)-[~]
$ django-admin startproject anas
(kali㉿kali)-[~]
$ cd anas
(kali㉿kali)-[~/anas]
$
```

#### ٤. انشاء تطبيق داخل مشروعنا :

The screenshot shows a Kali Linux desktop environment running in a VMware Workstation window. The desktop has a dark blue theme with a red vertical bar on the right side. A terminal window is open, showing the command:

```
(kali㉿kali)-[~/anas]$ python3 manage.py startapp myapp
```

The terminal window title is "Player 6". The desktop interface includes a dock with icons for terminal, file manager, browser, and others, and a taskbar at the bottom.

## ٥. اعداد التطبيق في المشروع نفتح ملف `set.py` ونضيف اسم التطبيق إلى داخلة :

The screenshot shows a terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal is running on a Kali Linux system with the command line interface. The user is in the directory `~/anas/anas`. The terminal shows the following session:

```
(kali㉿kali)-[~/anas]
$ ls
anas  manage.py  myapp

File System
(kali㉿kali)-[~/anas]
$ cd anas

Home
(kali㉿kali)-[~/anas/anas]
$ ls
asgi.py  __init__.py  __pycache__  settings.py  urls.py  wsgi.py

(kali㉿kali)-[~/anas/anas]
$ nano settings.py
```

The terminal window has a dark blue header bar with various icons for file operations, a title bar, and a status bar at the bottom. The main area of the terminal shows the command history and current directory contents.

## نقوم بإضافة البرنامج :

Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player 6 | || ⌂ ⌃ ⌄ ⌅ | 1 2 3 4 | ⌆

kali@kali: ~/anas/anas

File Actions Edit View Help

GNU nano 6.2 settings.py \*

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp',
]
```

New Folder

```
MIDDLEWARE = [
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location  
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

كتب هنا للبحث

## ٦. اعداد عرض صفحة html بسيطة :

نقوم بإنشاء ملف جديد داخل المجلد myapp باسم views.py

ونضيف الكود التالي:

The screenshot shows a terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal session is running under the user "anas" on the host "kali". The command history shows the following steps:

- cd myapp
- ls
- nano views.py

The "views.py" file has been opened in nano editor. The terminal prompt "(kali㉿kali)-[~/anas/myapp]" is visible at the top right.

```
(kali㉿kali)-[~/anas]
$ cd myapp
(kali㉿kali)-[~/anas/myapp]
$ ls
admin.py  apps.py  __init__.py  migrations  models.py  tests.py  views.py
(kali㉿kali)-[~/anas/myapp]
$ nano views.py
```

نقوم بإضافة الشفرة التالية:

Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player 6 | || □ □ □ □ | 1 2 3 4 | □

kali@kali: ~/anas/myapp

File Actions Edit View Help

GNU nano 6.2 views.py \*

```
from django.shortcuts import render
def home (request):
    return render (request,'home.html')

# Create your views here.
```

File System

Help Write Out Where Is Cut Execute Location  
Exit Read File Replace Paste Justify Go To Line

كتب هنا للبحث

٧. أعداد URL تحتاج إلى التوجيه إلى العرض الذي أنشئ نشيء ملف جديد باسم urls.py داخل

## مجلد myapp ونضع فيه الكود التالي :

A screenshot of a Kali Linux terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal shows a file browser interface with a sidebar containing icons for Home, Desktop, Downloads, and others. The main area displays a file tree under the path "/home/anas/myapp". The terminal prompt is "kali@kali: ~/anas/myapp". The desktop background features a dark Kali Linux logo.

```
GNU nano 6.2                                     urls.py
from django.urls import path

from . import views
urlpatterns = [
    path("", views.home, name='home'),
```

]

[ Read 7 lines ]

**^G** Help      **^O** Write Out      **^W** Where Is      **^K** Cut      **^T** Execute      **^C** Location  
**^X** Exit      **^R** Read File      **^\\** Replace      **^U** Paste      **^J** Justify      **^/** Go To Line

ومن ثم مجلد **anas** ونضيف الكود التالي :

The screenshot shows a terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal is running on a Kali Linux system, indicated by the prompt "kali@kali: ~/anas/anas". The user has navigated to the directory `~/anas` and run the command `ls`, which lists the files: `asgi.py`, `__init__.py`, `__pycache__`, `settings.py`, `urls.py`, and `wsgi.py`. The user then runs `nano urls.py` to edit the file. The bottom of the screen shows the Kali Linux desktop environment with various icons and a taskbar.

```
(kali㉿kali)-[~/anas]
$ cd anas

(kali㉿kali)-[~/anas/anas]
$ ls
asgi.py __init__.py __pycache__ settings.py urls.py wsgi.py

(kali㉿kali)-[~/anas/anas]
$ nano urls.py
```

Player 6 | II □ □ □



&gt;&gt; □ ○ □ ○ □ ○

19:26



kali@kali: ~/anas/anas

File Actions Edit View Help

GNU nano 6.2

urls.py \*

```
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
```

### Including another URLconf

```
1. Import the include() function: from django.urls import include, path
```

```
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
```

```
"""\n\nfrom django.contrib import admin\nfrom django.urls import path, include
```

Home

```
urlpatterns = [\n    path('admin/', admin.site.urls),\n    path("", include('myapp.urls')),\n]
```

New Folder



^G Help  
^X Exit

^O Write Out  
^R Read File

^W Where Is  
^\\ Replace

^K Cut  
^U Paste

^T Execute  
^J Justify

^C Location  
^/ Go To Line

## ٧. انشاء قالب html:

في داخل مجلد myapp ننشي مجلد جديد باسم templaltes

وداخل المجلد ننشي ملف باسم home.html ونضف الكود التالي :

The screenshot shows a terminal window titled "Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal is running on a Kali Linux desktop environment. The command history is as follows:

```
(kali㉿kali)-[~/anas]
$ cd myapp
(kali㉿kali)-[~/anas/myapp]
$ mkdir templates
(kali㉿kali)-[~/anas/myapp]
$ cd templates
(kali㉿kali)-[~/anas/myapp/templates]
$ nano home.html
```

The terminal window has a dark blue header bar with various icons and a status bar at the bottom. The main area shows the command-line interface with syntax highlighting.

Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player 6 | || □ □ □ □

kali@kali: ~/anas/myapp/templates

File Actions Edit View Help

GNU nano 6.2 home.html \*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>
        heloo
    </h1>
</body>
</html>
```

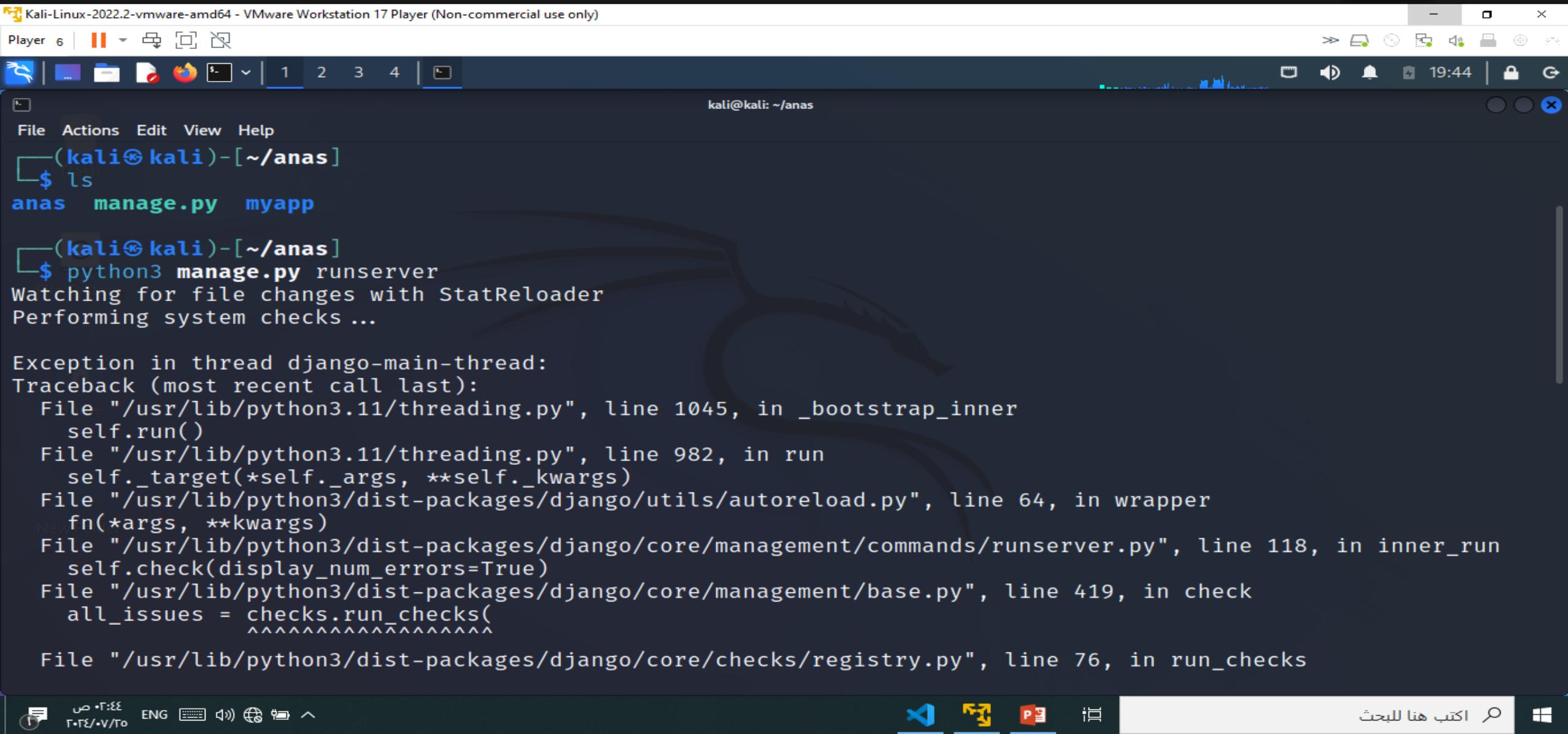
New Folder

^G Help   ^O Write Out   ^W Where Is   ^K Cut   ^T Execute  
^X Exit   ^R Read File   ^\ Replace   ^U Paste   ^J Justify   ^C Location  
^/ Go To Line   M-U Undo   M-E Redo

اكتب هنا للبحث

## ٩. تشغيل الخادم :

ثم نفتح متصفح الويب وانتقل الى <http://172.0.0.1:8000>.



Kali-Linux-2022.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)

Player 6 | || ⌂ ⌃ ⌄ ⌅ | 1 2 3 4 | ⌆

kali@kali: ~/anas

```
File Actions Edit View Help
(kali㉿kali)-[~/anas]
$ ls
anas manage.py myapp

(kali㉿kali)-[~/anas]
$ python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks ...

Exception in thread django-main-thread:
Traceback (most recent call last):
  File "/usr/lib/python3.11/threading.py", line 1045, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.11/threading.py", line 982, in run
    self._target(*self._args, **self._kwargs)
  File "/usr/lib/python3/dist-packages/django/utils/autoreload.py", line 64, in wrapper
    fn(*args, **kwargs)
  File "/usr/lib/python3/dist-packages/django/core/management/commands/runserver.py", line 118, in inner_run
    self.check(display_num_errors=True)
  File "/usr/lib/python3/dist-packages/django/core/management/base.py", line 419, in check
    all_issues = checks.run_checks(
                  ^^^^^^^^^^
  File "/usr/lib/python3/dist-packages/django/core/checks/registry.py", line 76, in run_checks
    for check in self.all_checks:

```

كتب هنا للبحث

وشكراً جزيلاً  
على حسن استماعكم