



Web Development using Python (Django)

www.hbpatel.in



Web Development using Python (Django)

www.hbpatel.in

django - Google Search

Google

django

All Books Images Videos News More Tools

About 19,70,00,000 results (0.39 seconds)

<https://www.djangoproject.com/> :
Django: The web framework for perfectionists with deadlines
Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of ...

[Documentation](#)
[Django at a glance](#) - Quick install guide - Getting started - Models

[Writing your first Django app](#)



The web framework for perfectionists with deadlines

[django](https://www.djangoproject.com/) The web framework for perfectionists with deadlines.

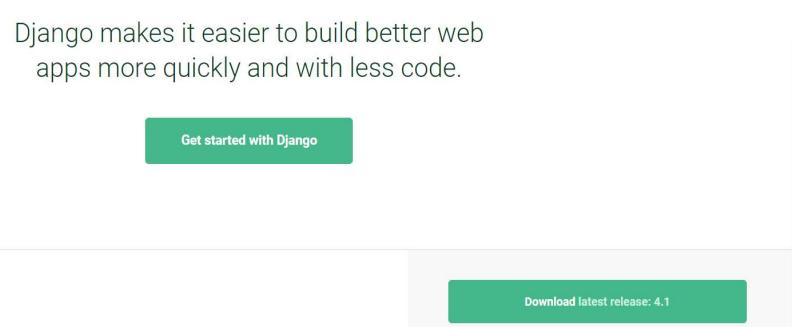
OVERVIEW DOWNLOAD DOCUMENTATION NEWS COMMUNITY CODE ISSUES ABOUT DONATE

Django makes it easier to build better web apps more quickly and with less code.

[Get started with Django](#)

Meet Django

Download latest release: 4.1



[Disqus](#)

[Pinterest](#)

[Facebook](#)

[Instagram](#)

[Spotify](#)

[Netflix](#)

[Youtube](#)

[The Washington Post](#)

[The Onion](#)

[National Geographic](#)

[Nasa](#)

[Dropbox](#)

[Opesta](#)

[Prezi](#)

[Eventbrite](#)

[Uber](#)

[Lyft](#)

[BitBucket](#)

[Mozilla](#)

[Summary](#)

Top websites build using Django Framework

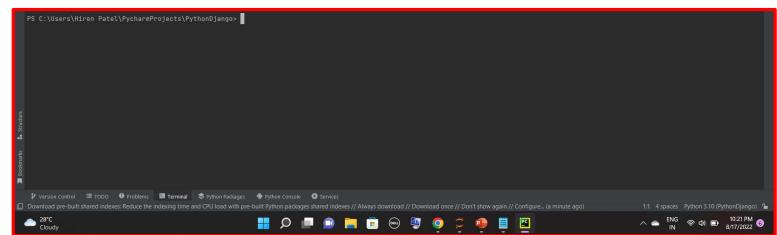
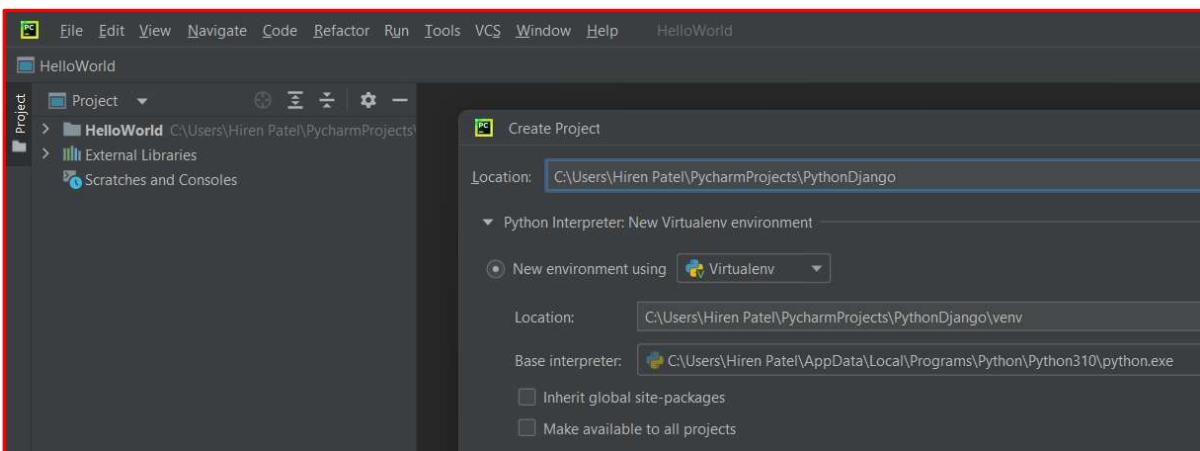
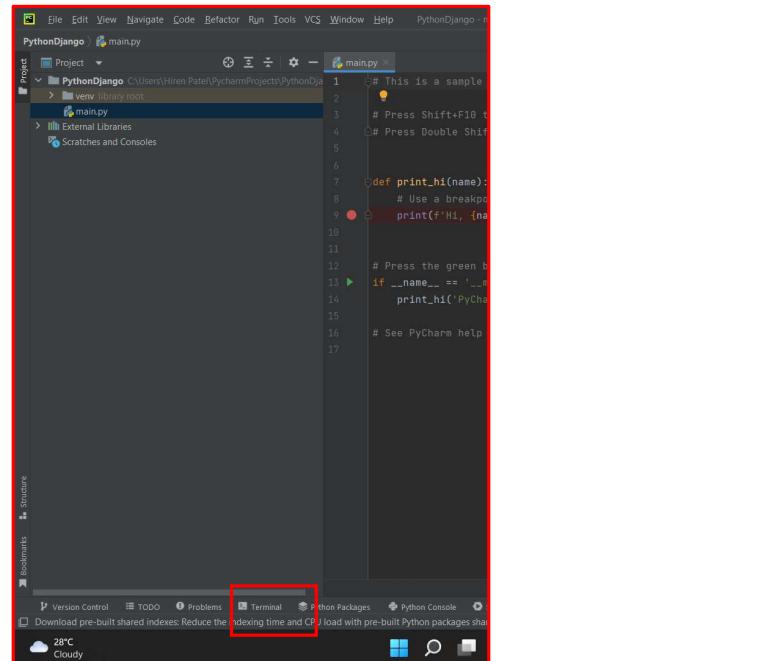
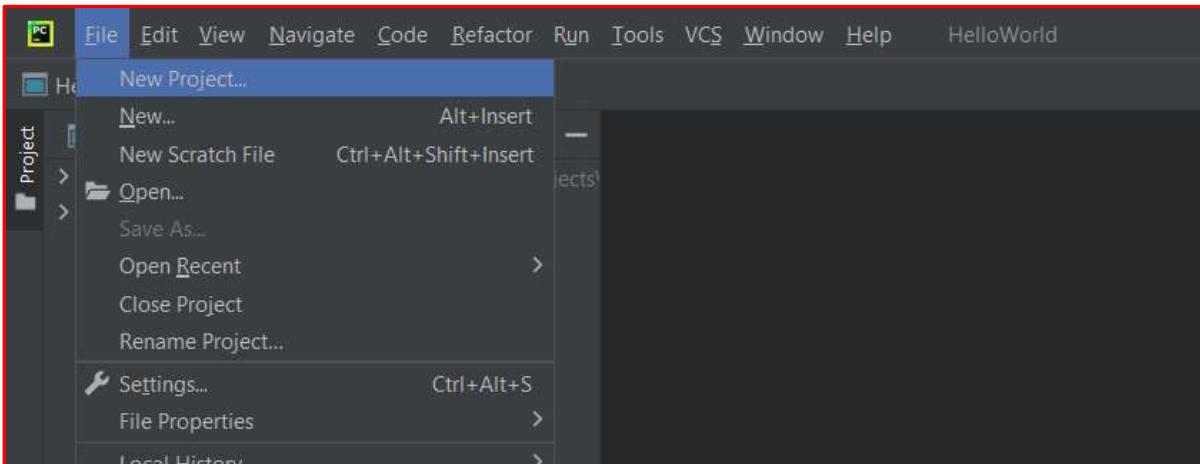
What is a Framework:

- Library of reusable modules
- Defines a structure for an application



Web Development using Python (Django)

www.hbpatel.in





Installing Django

www.hbpatel.in

The screenshot shows the PyCharm IDE interface with a red box highlighting the terminal window. The terminal output is as follows:

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> pip install django==4.1
Collecting django==4.1
  Downloading Django-4.1-py3-none-any.whl (8.1 MB)
    8.1/8.1 MB 234.4 kB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    42.3/42.3 kB 341.2 kB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting tzdata
  Downloading tzdata-2022.2-py2.py3-none-any.whl (336 kB)
    336.4/336.4 kB 260.9 kB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1 sqlparse-0.4.2 tzdata-2022.2

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

The terminal window is located in the bottom-left corner of the PyCharm interface. The PyCharm interface includes a Project tree on the left, a code editor with a main.py file, and various toolbars and status bars at the bottom.



Start Django Project

www.hbpatel.in

The screenshot shows the PyCharm IDE interface. The project navigation bar on the left has 'PythonDjango' selected. The main code editor window contains a file named 'main.py' with the following content:

```
# This is a sample Python script.  
# Press Shift+F10 to execute it or replace it with your code.  
# Press Double Shift to search everywhere for classes, files, actions, and settings.  
#
```

The terminal window at the bottom shows the command being run:

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> django-admin startproject PythonDjango .
```

The screenshot shows the PyCharm IDE interface. The project navigation bar on the left has 'PythonDjango' selected. The main code editor window shows the 'settings.py' file from the 'PythonDjango' directory.

The terminal window on the right shows the command being run:

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py runserver
```

The output of the command is:

```
Watching for file changes with StatReloader  
Performing system checks...  
  
System check identified no issues (0 silenced).  
  
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.  
Run 'python manage.py migrate' to apply them.  
August 17, 2022 - 22:32:02  
Django version 4.1, using settings 'PythonDjango.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```



Start Django Project

www.hbpatel.in

The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

[Django Documentation](#)
Topics, references, & how-to's

[Tutorial: A Polling App](#)
Get Started with Django

[Django Community](#)
Connect, get help, or contribute

Cloudy 28°C

10:33 PM 8/17/2022



Django Project

www.hbpatel.in

Order Management (Django APP #1)	Customer Management (Django APP #2)
Product Management (Django APP #3)	Other (Django APP #4)



PythonDjango\urls.py

www.hbpatel.in

The screenshot shows the PyCharm IDE interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "PythonDjango".
 - products:** Contains migrations, admin.py, apps.py, models.py, tests.py, and urls.py.
 - PythonDjango:** Contains __init__.py, asgi.py, settings.py, and urls.py. The "urls.py" file is highlighted with a red box.
 - venv:** library root contains db.sqlite3, main.py, and manage.py.
 - External Libraries:** Shows various Python packages.
- Code Editor:** The right pane displays the content of "PythonDjango\urls.py".

```
"""PythonDjango URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.1/topics/http/urls/
Examples:
Function views
    1. Add an import: from my_app import views
    2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
    1. Add an import: from other_app.views import Home
    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'))
"""

from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('products/', include('products.urls')),
]
```



PythonDjango\products\migrations\..

www.hbpatel.in

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render

def index(request):
    return HttpResponseRedirect('Hello Hiren')
```

Project:

- PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - views.py
- PythonDjango
- venv library root

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango

PythonDjango > products

Project:

- PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
- products
 - migration
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - views.py
- PythonDjango
- venv library
- db.sqlite3

New

- Cut Ctrl+X
- Copy Ctrl+C
- Copy Path/Reference...
- Paste Ctrl+V
- Find Usages Alt+F7
- Find in Files... Ctrl+Shift+F
- Replace in Files... Ctrl+Shift+R
- Inspect Code...

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango

PythonDjango > products

Project:

- PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - views.py
- PythonDjango
- venv library root
- db.sqlite3

New Python file

- urls
- Python file
- Python unit test
- Python stub

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - products\urls.py

PythonDjango > products > urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('new', views.new)
```

Project:

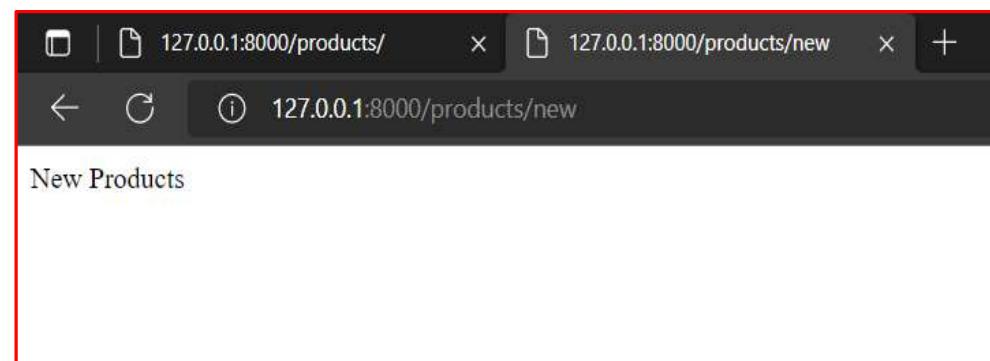
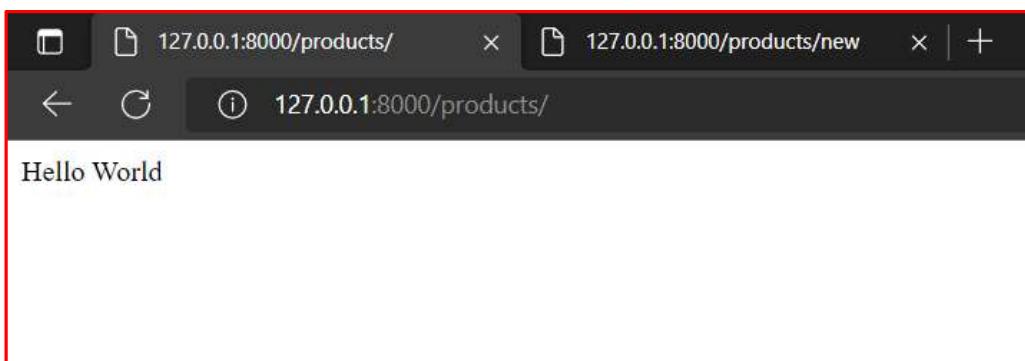
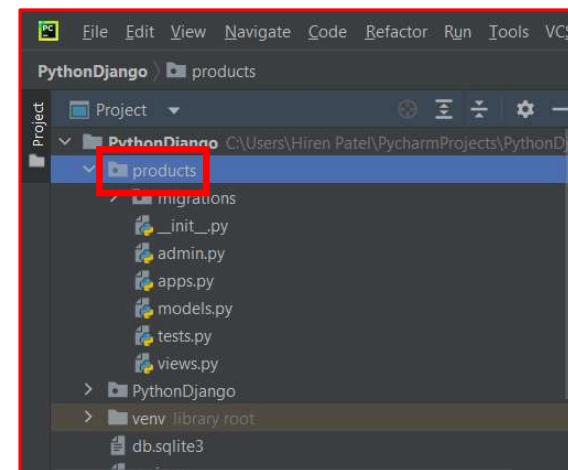
- PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - views.py
- PythonDjango
- venv library root
- db.sqlite3



Running Django Project

www.hbpatel.in

```
Terminal: C:\Windows\rshell.exe x C:\Windows\rshell.exe (2) x + 
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py startapp products
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```





Models in Python

www.hbpatel.in

The screenshot shows the PyCharm interface with the project 'PythonDjango' open. The 'products' app is selected in the project tree. The 'models.py' file is open in the editor, showing its contents:

```
from django.db import models

# Create your models here.
```

```
from django.db import models

class Product(models.Model):
    name = models.CharField(max_length=255)
    price = models.FloatField()
    stock = models.IntegerField()
    image_url = models.CharField(max_length=2083)
```

The screenshot shows the PyCharm interface with the project 'PythonDjango' open. The 'db.sqlite3' file is selected in the project tree.

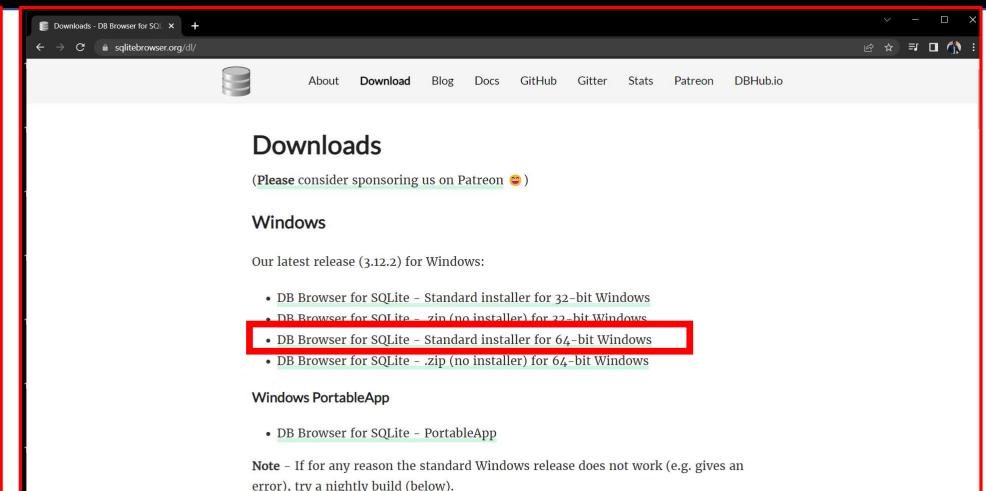
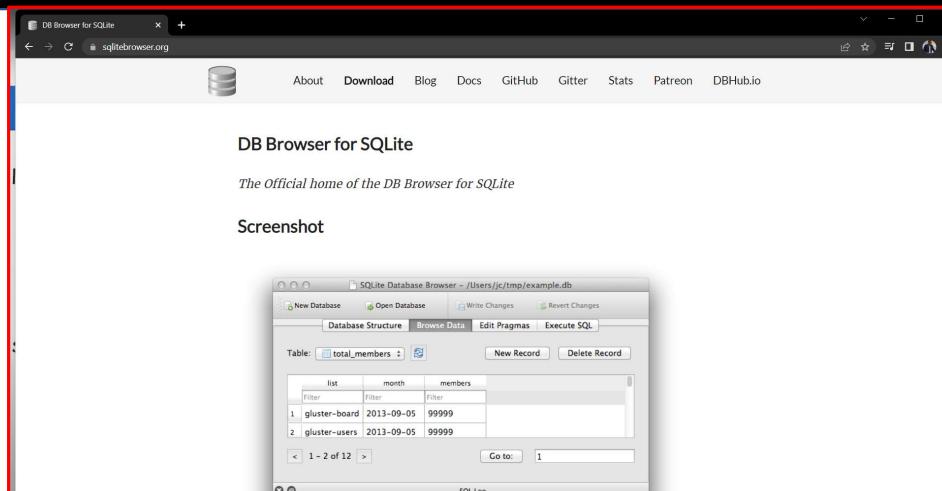
The screenshot shows a Google search results page for 'db server for sqlite'. The top result is 'DB Browser for SQLite' with the following description:

DB Browser for SQLite (DB4S) is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.



DB Browser for SQLite

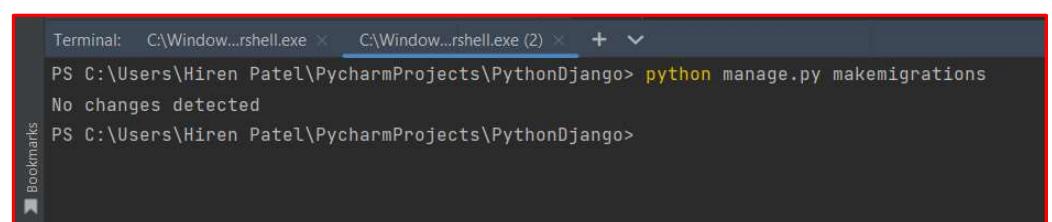
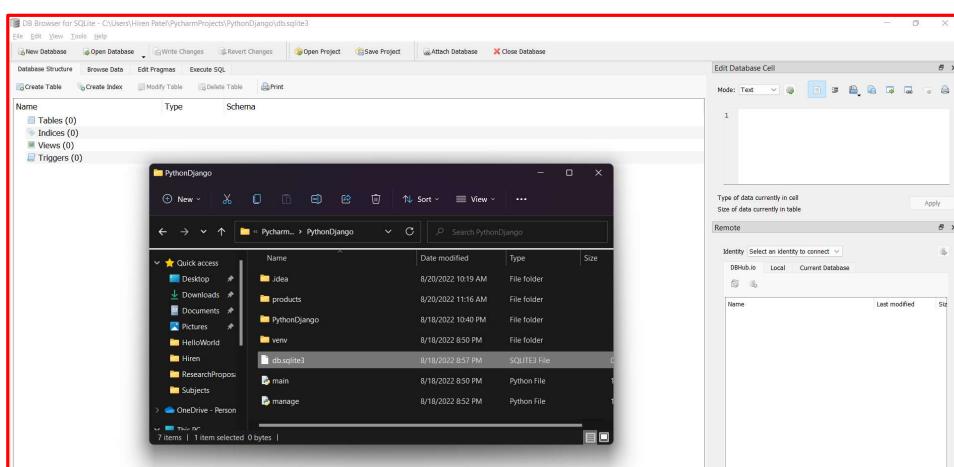
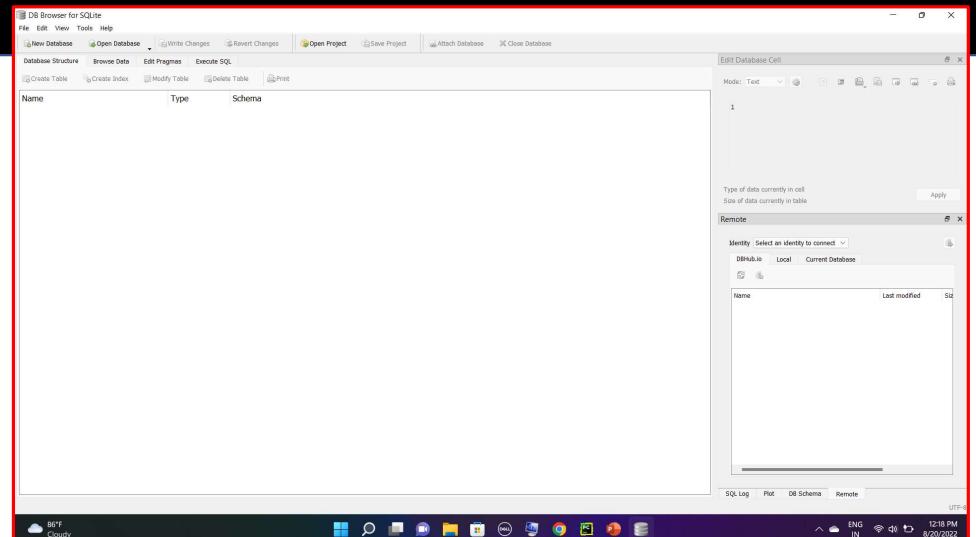
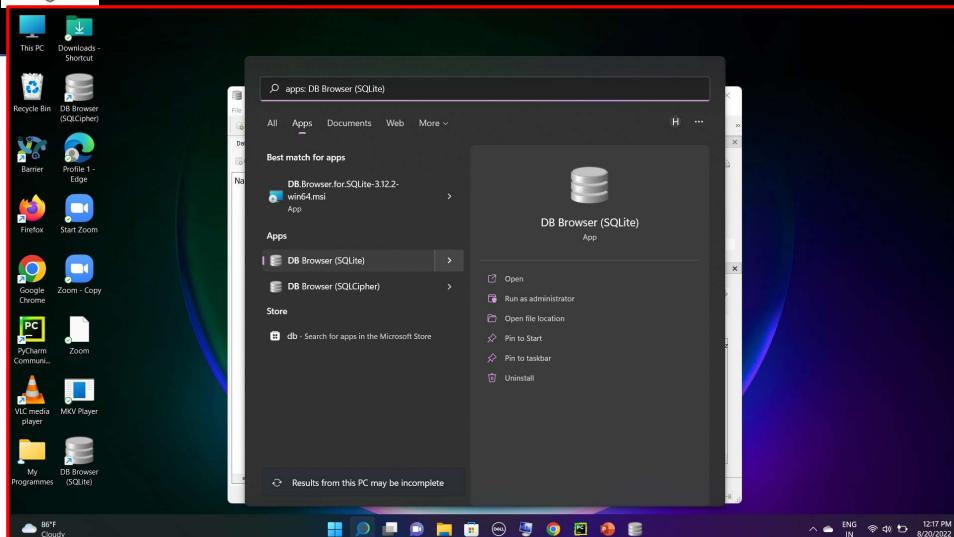
www.hbpatel.in





DB Browser for SQLite

www.hbpatel.in



DB Browser for SQLite

www.hbpatel.in



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - settings.py

PythonDjango > PythonDjango > settings.py

Project

```
24 # SECURITY WARNING: don't run with debug turned on in production!
25 DEBUG = True
26
27 ALLOWED_HOSTS = []
28
29
30 # Application definition
31
32 INSTALLED_APPS = [
33     'django.contrib.admin',
34     'django.contrib.auth',
35     'django.contrib.contenttypes',
36     'django.contrib.sessions',
37     'django.contrib.messages',
38     'django.contrib.staticfiles',
39 ]
40 
```

24 # SECURITY WARNING: don't run with debug turned on in production!
25 DEBUG = True
26
27 ALLOWED_HOSTS = []
28
29
30 # Application definition
31
32 INSTALLED_APPS = [
33 'django.contrib.admin',
34 'django.contrib.auth',
35 'django.contrib.contenttypes',
36 'django.contrib.sessions',
37 'django.contrib.messages',
38 'django.contrib.staticfiles',
39]
40

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - apps.py

PythonDjango > products > apps.py

Project

```
1 from django.apps import AppConfig
2
3 class ProductsConfig(AppConfig):
4     default_auto_field = 'django.db.models.BigAutoField'
5     name = 'products'
6 
```

1 from django.apps import AppConfig
2
3 class ProductsConfig(AppConfig):
4 default_auto_field = 'django.db.models.BigAutoField'
5 name = 'products'
6

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

```
Terminal: C:\Windows\system32\cmd.exe x C:\Windows\system32\cmd.exe (2) x + v
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
No changes detected
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
Migrations for 'products':
  products\migrations\0001_initial.py
    - Create model Product
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

The screenshot shows the PyCharm IDE interface with a red border around the main content area. The top navigation bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - 0001_initial.py. The Project tool window on the left shows the directory structure: PythonDjango (containing venv, db.sqlite3, main.py, manage.py) and products (containing __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, views.py). A red box highlights the migrations folder within products, which contains 0001_initial.py. The code editor on the right displays the content of 0001_initial.py:

```
# Generated by Django 4.1 on 2022-08-20 07:10

from django.db import migrations, models

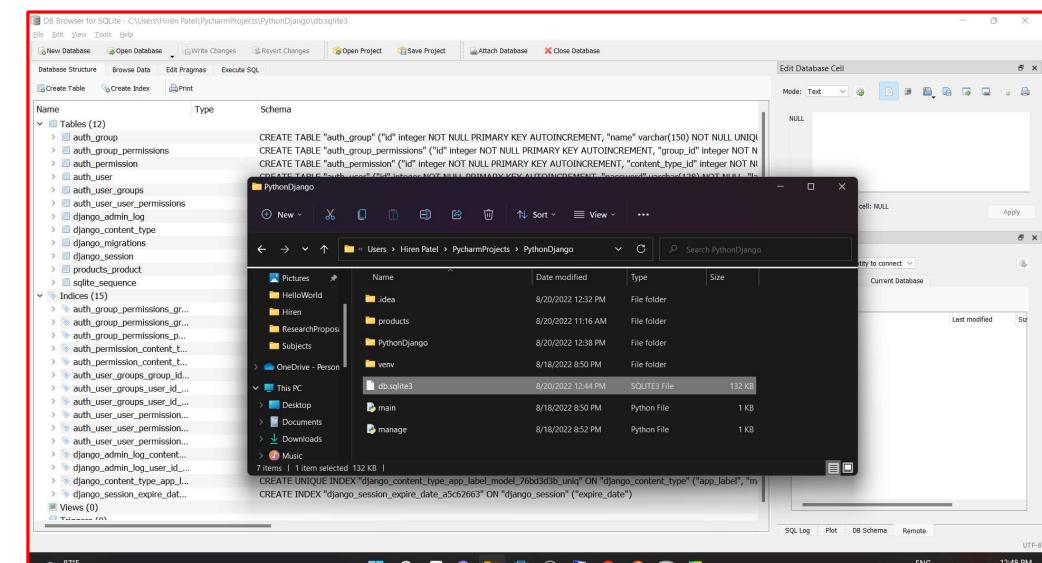
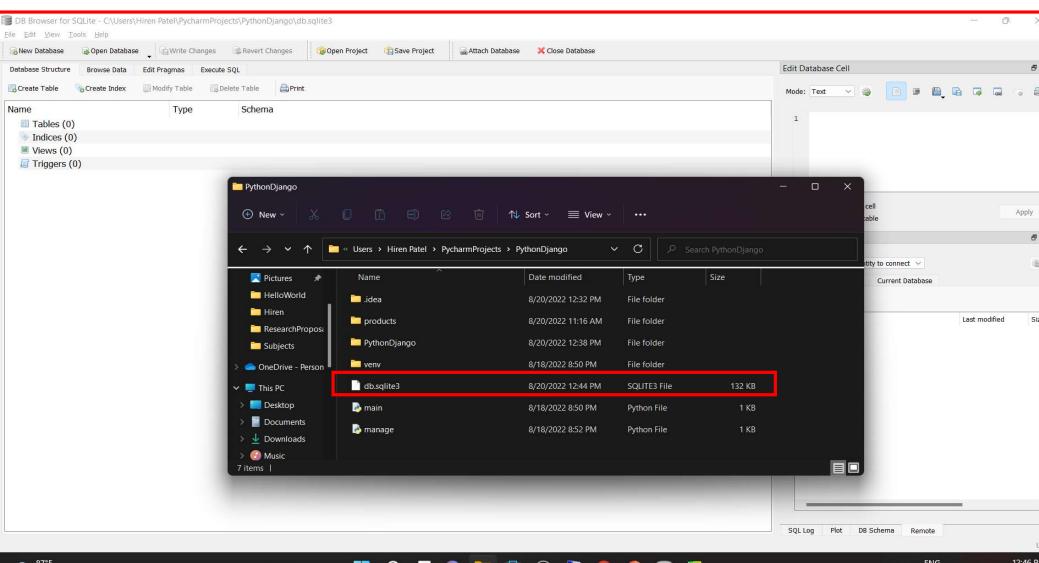
class Migration(migrations.Migration):
    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Product',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=255)),
                ('price', models.FloatField()),
                ('stock', models.IntegerField()),
                ('image_url', models.CharField(max_length=2083)),
            ],
        ),
    ],

```

```
Terminal: C:\Windows\rshell.exe x C:\Windows\rshell.exe (2) + ▾  
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate  
Operations to perform:  
  Applying auth.0007_alter_validators_add_error_messages... OK  
  Applying auth.0008_alter_user_username_max_length... OK  
  Applying auth.0009_alter_user_last_name_max_length... OK  
  Applying auth.0010_alter_group_name_max_length... OK  
  Applying auth.0011_update_proxy_permissions... OK  
  Applying auth.0012_alter_user_first_name_max_length... OK  
  Applying products.0001_initial... OK  
  Applying sessions.0001_initial... OK  
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```



Name	Type	Schema
Tables (12)		
auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL)
auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL)
auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL)
auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL)
auth_user_groups		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL)
auth_user_user_permissions		CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL)
django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL)
django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL)
django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL)
django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL, "session_data" longtext NOT NULL, "expire_date" datetime NOT NULL)
products_product		CREATE TABLE "products_product" ("id" integer NOT NULL, "name" varchar(255) NOT NULL, "price" real NOT NULL, "stock" integer NOT NULL, "image_url" varchar(2083) NOT NULL)
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, products, sessions
Running migrations:
  No migrations to apply.
```

The screenshot shows the PyCharm IDE interface with the following details:

- Code Editor:** The main window displays the `models.py` file for the `products` app. A red box highlights the class definition for `Offer`:

```
class Offer(models.Model):
    code = models.CharField(max_length=10)
    description = models.CharField(max_length=255)
    discount = models.FloatField()
```
- Project Explorer:** The left sidebar shows the project structure. A red box highlights the `models.py` file under the `products` app.
- Terminal:** The bottom panel shows a terminal window with the following command history:

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
Migrations for 'products':
  products\migrations\0002_offer.py
    - Create model Offer
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, products, sessions
Running migrations:
  Applying products.0002_offer... OK
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

A red box highlights the `python manage.py migrate` command.
- Status Bar:** The bottom right corner shows system information: 88°F, Cloudy, ENG IN, 1:44 PM, and 8/20/2022.

DB Browser for SQLite - C:\Users\Hiren Patel\PycharmProjects\PythonDjango\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Print

Name Type Schema

Tables (12)

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- products_product
 - id integer
 - name varchar(255)
 - price real
 - stock integer
 - image_url varchar(2083)
- sqlite_sequence

Indices (15)

- auth_group_permissions_gr...
- auth_group_permissions_gr...
- auth_group_permissions_p...
- auth_permission_content_t...
- auth_permission_content_t...
- auth_user_groups_group_id...
- auth_user_groups_user_id...
- auth_user_groups_user_id...
- auth_user_user_permission...
- auth_user_user_permission...
- auth_user_user_permission...
- django_admin_log_content...

CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(150) NOT NULL UNIQUE);
CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL, "permission_id" integer NOT NULL);
CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL, "codename" varchar(100) NOT NULL, "name" varchar(50) NOT NULL);
CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime, "is_superuser" integer NOT NULL, "username" varchar(150) NOT NULL, "first_name" varchar(30), "last_name" varchar(30), "email" varchar(254));
CREATE TABLE "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL, "user_id" integer NOT NULL);
CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "permission_id" integer NOT NULL, "user_id" integer NOT NULL);
CREATE INDEX "auth_user_user_permissions_permission_id_1fb5bf2c" ON "auth_user_user_permissions" ("permission_id");
CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id");
CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id");
CREATE INDEX "django_admin_log_content_type_id_e2fbc0ab" ON "django_admin_log" ("content_type_id");

Edit Database Cell

Mode: Text

NULL

cell: NULL Apply

PythonDjango

New |剪切 |复制 |粘贴 |插入 |删除 |上移 |下移 |排序 |视图 |更多

Search PythonDjango

Name	Date modified	Type	Size
Downloads	8/20/2022 12:32 PM	File folder	
Documents	8/20/2022 1:43 PM	File folder	
Pictures	8/20/2022 12:38 PM	File folder	
HelloWorld	8/20/2022 1:43 PM	File folder	
Hiren	8/20/2022 12:38 PM	File folder	
ResearchPropos...	8/20/2022 1:43 PM	File folder	
Subjects	8/20/2022 1:43 PM	File folder	
PythonDjango	8/20/2022 1:43 PM	File folder	
venv	8/18/2022 8:50 PM	File folder	
db.sqlite3	8/20/2022 1:44 PM	SQLITE3 File	136 KB
main	8/18/2022 8:50 PM	Python File	1 KB
manage	8/18/2022 8:52 PM	Python File	1 KB

7 items | 1 item selected 136 KB

CREATE INDEX "auth_user_user_permissions_permission_id_1fb5bf2c" ON "auth_user_user_permissions" ("permission_id");
CREATE INDEX "auth_user_user_permissions_user_id_a95ead1b" ON "auth_user_user_permissions" ("user_id");
CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_permission_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id");
CREATE INDEX "django_admin_log_content_type_id_e2fbc0ab" ON "django_admin_log" ("content_type_id");

SQL Log Plot DB Schema Remote

UTF-8

DB Browser for SQLite - C:\Users\Hiren Patel\PycharmProjects\PythonDjango\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database Close Database

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Print

Name Type Schema

Tables (13)

- auth_group
- auth_group_permissions
- auth_permission
- auth_user
- auth_user_groups
- auth_user_user_permissions
- django_admin_log
- django_content_type
- django_migrations
- django_session
- products_offer
- products_product
- sqlite_sequence

Products Offer Table Schema:

	Type
id	integer
code	varchar(10)
description	varchar(255)
discount	real

Indices (15)

- auth_group_permissions_group_id_b120cbf9
- auth_group_permissions_group_id_permission_id_0cd325b0_uniq
- auth_group_permissions_permission_id_84c5c92e
- auth_permission_content_type_id_2f476e4b
- auth_permission_content_type_id_codename_01ab375a_uniq
- auth_user_groups_group_id_97559544
- auth_user_groups_user_id_6a12ed8b
- auth_user_groups_user_id_group_id_94350c0c_uniq
- auth_user_user_permissions_permission_id_1ffb5f2c
- auth_user_user_permissions_user_id_a95ead1b
- auth_user_user_permissions_user_id_permission_id_14a6b632_uniq
- django_admin_log_content_type_id_e3fbc0ab

Edit Database Cell

Mode: Text

NULL

Type of data currently in cell: NULL
0 byte(s) Apply

Remote

Identity Select an identity to connect

DBHub.io Local Current Database

Name Last modified Size

SQL Log Plot DB Schema Remote

88°F Cloudy ENG IN 1:47 PM 8/20/2022



Admin Panel: 127.0.0.1:8000/admin

www.hbpatel.in

The screenshot shows a browser window with a red border containing the Django administration login page. The URL in the address bar is 127.0.0.1:8000/admin/login/?next=/admin/. Below the browser is a terminal window with a red border, also containing a red box highlighting its title bar. The terminal shows the command `python manage.py createsuperuser` being run in a Windows environment. It prompts for a username ('admin'), email ('hbpatel1976@gmail.com'), and password, which are rejected as being too short, common, or numeric. Finally, it bypasses validation and creates the user successfully.

```
Terminal: C:\Windows\system32\cmd.exe (4) PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py createsuperuser
Username (leave blank to use 'hirenpatel'): admin
Email address: hbpatel1976@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```



Admin Panel

www.hbpatel.in

The screenshot shows a browser window with two tabs. The top tab is titled "Log in | Django site admin" and has the URL "127.0.0.1:8000/admin/login/?next=/admin/". The bottom tab is titled "Site administration | Django site" and has the URL "127.0.0.1:8000/admin/".

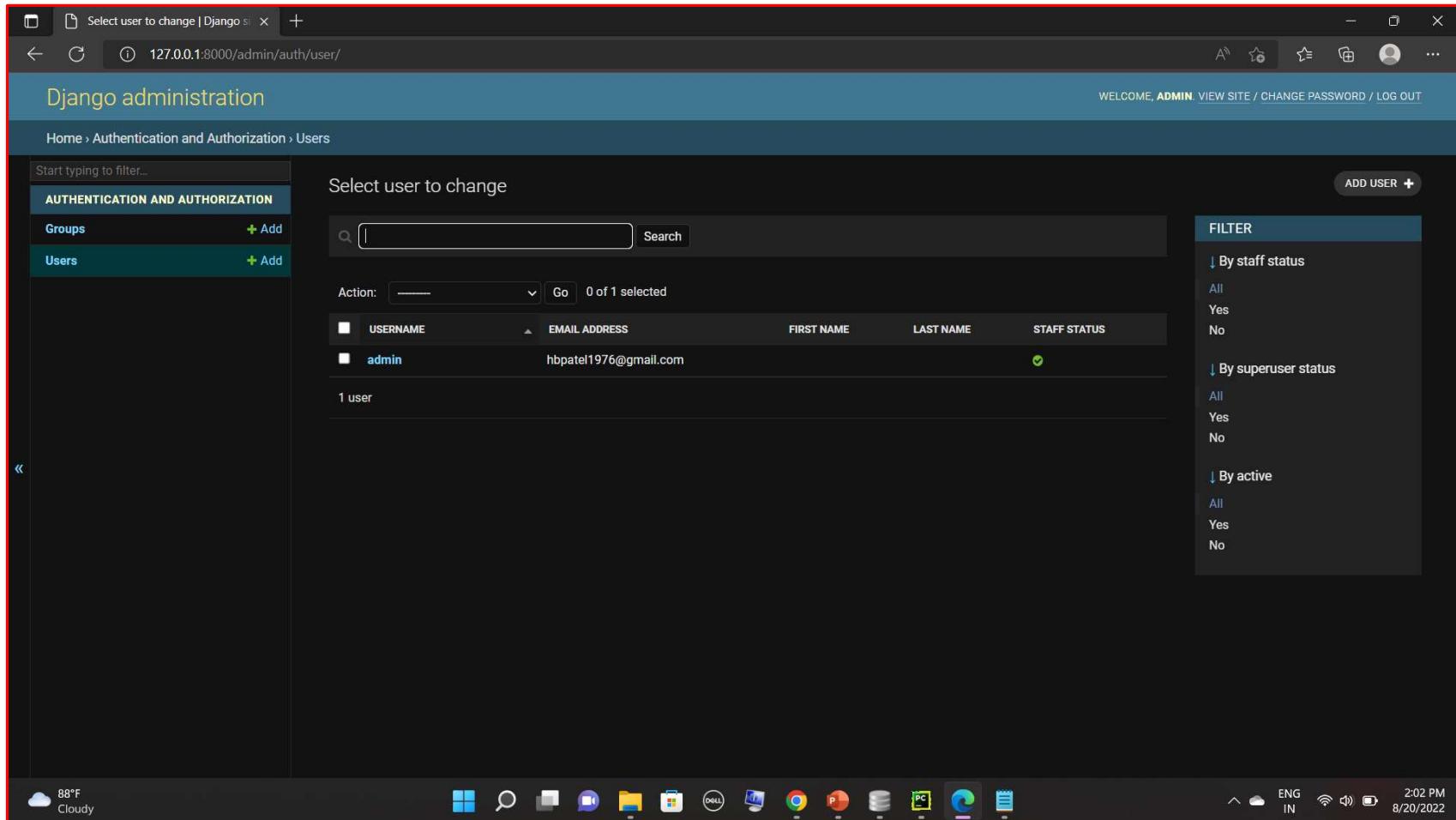
The login form in the top tab has a red box around it. Inside the box, the "Username" field contains "admin" and the "Password" field contains "1234".

The dashboard in the bottom tab also has a red box around it. It displays the "Django administration" header and the "Site administration" section. Under "AUTHENTICATION AND AUTHORIZATION", there are links for "Groups" and "Users", each with "+ Add" and "Change" buttons. On the right side of the dashboard, there are sections for "Recent actions" and "My actions", both of which say "None available".



Admin Panel

www.hbpatel.in



The screenshot shows the Django Admin interface for managing users. The URL in the browser is `127.0.0.1:8000/admin/auth/user/`. The page title is "Django administration" and the current view is "Authentication and Authorization > Users". A search bar at the top right allows filtering by "Start typing to filter...". On the left, a sidebar lists "Groups" and "Users" with "Add" buttons. The main content area is titled "Select user to change" and displays a table with one user entry:

Action	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	admin	hbpatel1976@gmail.com			<input checked="" type="checkbox"/>

The table shows 1 user. To the right of the table is a "FILTER" sidebar with three sections: "By staff status" (All, Yes, No), "By superuser status" (All, Yes, No), and "By active" (All, Yes, No). The bottom of the screen shows a Windows taskbar with various icons and system status information.



admin.py [product tab]

www.hbpatel.in

The screenshot shows a PyCharm IDE interface with a red box highlighting the project structure and the admin.py file. The admin.py file contains the following code:

```
from django.contrib import admin
from .models import Product
admin.site.register(Product)
```

The PyCharm interface includes tabs for File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - admin.py. The project structure on the left shows a PythonDjango project with a products directory containing migrations, admin.py (highlighted), apps.py, models.py, tests.py, urls.py, and views.py.

Below the PyCharm window is a browser window showing the Django administration site at 127.0.0.1:8000/admin/. The browser title is "Site administration | Django site". The main content area displays the "Django administration" header and "Site administration" section. Under "AUTHENTICATION AND AUTHORIZATION", there are links for "Groups" and "Users", each with "Add" and "Change" buttons. A red box highlights the "Products" link under the "PRODUCTS" heading. To the right, there are sections for "Recent actions" and "My actions", both currently showing "None available".

Adding a product

www.hbpatel.in

127.0.0.1:8000/admin/products/product/

Django administration

Home > Products > Products

Select product to change

0 products

ADD PRODUCT +

Start typing to filter...

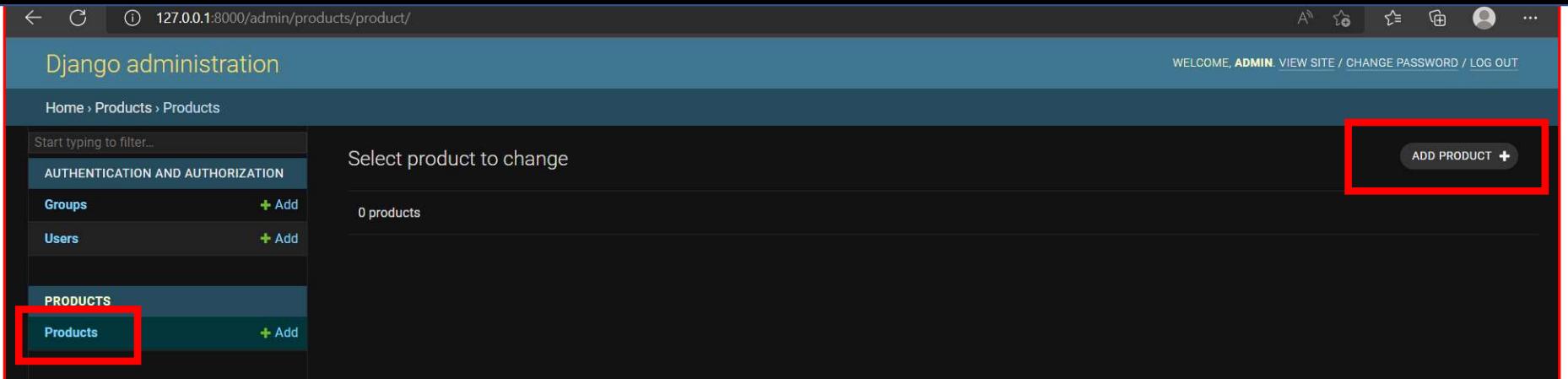
AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Products + Add



Add product | Django site admin

127.0.0.1:8000/admin/products/product/add/

Django administration

Welcome, ADMIN. View site / Change password / Log out

Home > Products > Products > Add product

Add product

Name: Orage

Price: 70

Stock: 100

Image url: <https://www.freefoodphotos.com/imagelibra>

Save and add another

Save and continue editing

SAVE

Start typing to filter...

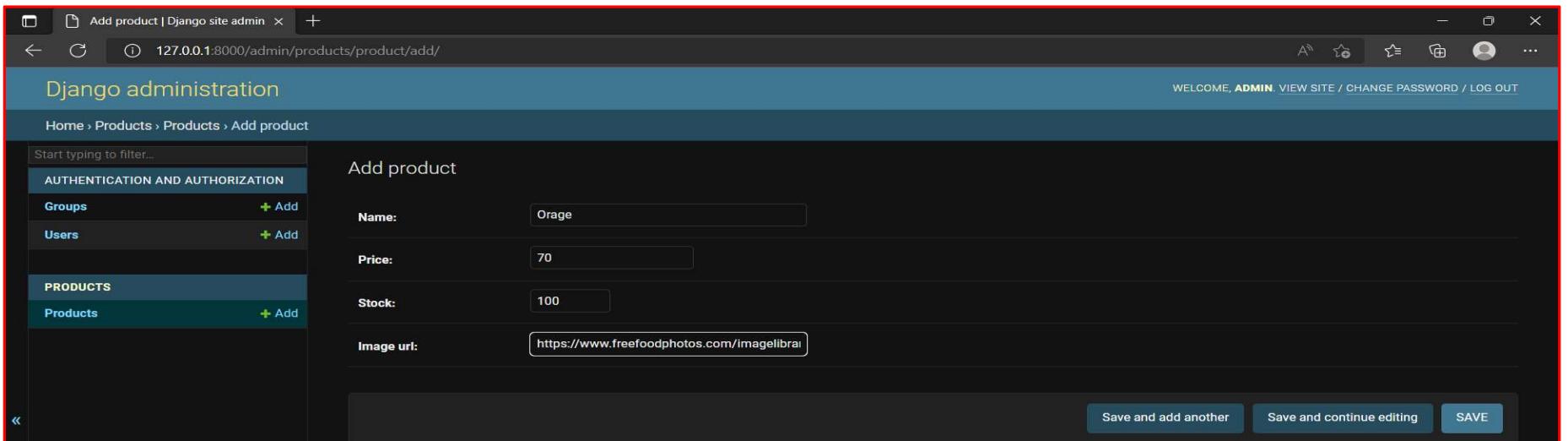
AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Products + Add



Viewing the product

www.hbpatel.in

The screenshot shows the Django admin interface with a dark theme. The top navigation bar includes a logo on the left, a search bar with the URL '127.0.0.1:8000/admin/products/product/', and various browser control icons on the right.

The main title 'Django administration' is displayed above the breadcrumb 'Home > Products > Products'. On the left, a sidebar lists 'Start typing to filter...', 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users), and 'PRODUCTS' (Products). The 'Products' section is highlighted in green.

The central area is titled 'Select product to change' and shows a single item: 'Orage' (Product object (1)). The 'Action' dropdown is set to '—' and shows '0 of 1 selected'. There are two checkboxes: one for 'PRODUCT' and one for 'Product object (1)', both of which are checked. A message '1 product' is displayed below.

A red box highlights the entire top-level product list screen.

The bottom part of the screenshot shows a detailed view of the selected product 'Orage'. The title is 'Product object (1) | Change prod'. The 'Change product' form contains the following fields:

- Name: Orage
- Price: 70.0
- Stock: 100
- Image url: <https://www.freefoodphotos.com/imagelibra>

At the bottom are three buttons: 'Delete' (red), 'Save and add another', 'Save and continue editing', and 'SAVE' (blue).



Viewing the product

www.hbpatel.in

The screenshot illustrates the development and administration process for a Django application.

PyCharm IDE: The top window shows the PythonDjango project structure. The `admin.py` file in the `products` app is open, defining a `ProductAdmin` class that registers the `Product` model with `list_display = ('name', 'price', 'stock')`.

```
from django.contrib import admin
from .models import Product

class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'price', 'stock')

admin.site.register(Product, ProductAdmin)
```

Django Administration: The bottom window shows the Django admin interface at `127.0.0.1:8000/admin/products/product/`. It displays a list of products, with one entry for "Orage". The "ADD PRODUCT" button is highlighted with a red box.

	NAME	PRICE	STOCK
<input checked="" type="checkbox"/>	Orage	70.0	100



Adding more product

www.hbpatel.in

Select product to change | Django × +

127.0.0.1:8000/admin/products/product/

Django administration

Welcome, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Products › Products

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Products + Add

The product "Product object (5)" was added successfully.

Select product to change

Action: — Go 0 of 5 selected

ADD PRODUCT +

<input type="checkbox"/>	NAME	PRICE	STOCK
<input type="checkbox"/>	Grapes	130.0	30
<input type="checkbox"/>	Apple	120.0	80
<input type="checkbox"/>	Banana	20.0	100
<input type="checkbox"/>	Strawberry	120.0	50
<input type="checkbox"/>	Orage	70.0	100

5 products

Select product to change | Django × +

127.0.0.1:8000/products/

127.0.0.1:8000/products/

Hello World

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - admin.py

PythonDjango › products › admin.py

```
from django.contrib import admin
from .models import Product, Offer

class OfferAdmin(admin.ModelAdmin):
    list_display = ('code', 'discount')

class ProductAdmin(admin.ModelAdmin):
    list_display = ('name', 'price', 'stock')

admin.site.register(Offer, OfferAdmin)
admin.site.register(Product, ProductAdmin)
```

Site administration | Django site x 127.0.0.1:8000/products/ 127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#) [Change](#)
- Users [+ Add](#) [Change](#)

PRODUCTS

- Offers [+ Add](#) [Change](#)
- Products [+ Add](#) [Change](#)

Select offer to change | Django s x 127.0.0.1:8000/products/ 127.0.0.1:8000/admin/products/offer/

Django administration

WELCOME, ADMIN | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Offers

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

PRODUCTS

- Offers [+ Add](#)
- Products [+ Add](#)

The offer "Offer object (1)" was changed successfully.

Select offer to change

Action: 0 of 2 selected

	CODE	DISCOUNT
<input type="checkbox"/>	DIWALI15	0.15
<input type="checkbox"/>	INDEPEN10	0.1

2 offers [ADD OFFER +](#)

The screenshot shows the PyCharm IDE interface with a project named "PythonDjango". The "products" app is selected in the project tree. A context menu is open over the "templates" folder, with the "New" option highlighted. A submenu is displayed, showing options like "File", "New Scratch File", "Directory", "Python Package", etc.

```
from django.http import HttpResponse
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return HttpResponse('Hello World')

def new(request):
    return HttpResponse('New Products')
```

The screenshot shows the PyCharm IDE interface with a project named "PythonDjango". The "products" app is selected in the project tree. A context menu is open over the "index.htm" file, with the "New" option highlighted. A submenu is displayed, showing options like "File", "New Scratch File", "Directory", "Python Package", etc.

```
from django.http import HttpResponse
from django.shortcuts import render
from .models import Product
```

The screenshot shows the PyCharm IDE interface with a project named "PythonDjango". The "products" app is selected in the project tree. A context menu is open over the "index.htm" file, with the "New" option highlighted. A submenu is displayed, showing options like "File", "New Scratch File", "Directory", "Python Package", etc.

```
from django.http import HttpResponse
from django.shortcuts import render
from .models import Product
```



Viewing product

www.hbpatel.in

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango > products > templates > index.html

Project

```
<h1>Products</h1>
<ul>
    <li> Item 1 </li>
    <li> Item 2 </li>
    <li> Item 2 </li>
    <li> Item 4 </li>
</ul>
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

Project

```
from django.http import HttpResponse
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return render(request, 'index.html')

def new(request):
    return HttpResponse('New Products')
```

Select offer to change | Django s x 127.0.0.1:8000/products/ x +

← ⌂ ⓘ 127.0.0.1:8000/products/

Products

- Item 1
- Item 2
- Item 3
- Item 4

Viewing product

www.hbpatel.in



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango › products › views.py

```
Project
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  + products
    + migrations
    + templates
      + index.html
      + __init__.py
      + admin.py
      + apps.py
      + models.py
      + tests.py
      + urls.py
      + views.py
  + PythonDjango
  + venv library root
    + db.sqlite3
    + main.py
```

index.html x views.py x Press Esc to exit full screen

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return render(request,
                  'index.html',
                  {'products': products})

def new(request):
    return HttpResponseRedirect('New Products')
```

Select offer to change | Django s x 127.0.0.1:8000/products/

← ⌂ ⓘ 127.0.0.1:8000/products/

Products

- Orage
- Strawberry
- Banana
- Apple
- Grapes

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango › products › templates › index.html

```
Project
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  + products
    + migrations
    + templates
      + index.html
      + __init__.py
      + admin.py
      + apps.py
      + models.py
```

index.html x views.py x

```
<h1>Products</h1>
<ul>
  {% for product in products %}
    <li> {{ product.name }} </li>
  {% endfor %}
</ul>
```



Viewing product

www.hbpatel.in

The screenshot shows the PyCharm IDE interface. On the left is the project tree with 'PythonDjango' selected, containing 'products', 'migrations', 'templates' (with 'index.html'), and other files like '_init_.py', 'admin.py', and 'apps.py'. The main editor window displays the 'index.html' template code:

```
<h1>Products</h1>
<ul>
    {% for product in products %}
        <li>{{ product.name }} (Rs. {{ product.price }})</li>
    {% endfor %}
</ul>
```

The screenshot shows a web browser window with the URL '127.0.0.1:8000/products/'. The page title is 'Products'. The content is a list of fruit items with their prices:

- Orage (Rs. 70.0)
- Strawberry (Rs. 120.0)
- Banana (Rs. 20.0)
- Apple (Rs. 120.0)
- Grapes (Rs. 130.0)

The screenshot shows a Google search results page for 'bootstrap framework'. The search bar contains 'bootstrap framework'. Below it, there are filters for 'All', 'Images', 'Videos', 'News', 'Shopping', and 'More'. The text 'About 8,86,00,000 results (0.50 seconds)' is displayed. The first result is a link to 'https://getbootstrap.com' with the text 'Bootstrap · The most popular HTML, CSS, and JS library in the ...' and a description 'Bootstrap utilizes Sass for a modular and customizable architecture. Import only the components you need, enable global options like gradients and shadows, and ...'

The screenshot shows the Bootstrap website at 'getbootstrap.com'. The header features a purple navigation bar with tabs for 'Docs' (which is highlighted with a red box), 'Examples', 'Icons', 'Themes', and 'Blog'. A yellow banner in the top right corner says 'New in v5.2 CSS variables, resp...'. The main content area shows a large blue background image.

Get started with Bootstrap · Bootstrap

getbootstrap.com/docs/5.2/getting-started/introduction/

Docs Examples Icons Themes Blog Search v5.2

Getting started

- Introduction
- Download
- Contents
- Browsers & devices
- JavaScript
- Webpack
- Parcel
- Vite
- Accessibility
- RFS
- RTL
- Contribute

Customize

- Overview
- Sass
- Options
- Color
- Components
- CSS variables
- Optimize

Get started with Bootstrap

View on GitHub

Bootstrap is a powerful, feature-packed frontend toolkit. Build anything—from prototype to production—in minutes.

Join us for demos, and best practices to modernize, unify, and innovate with data - Register now

ads via Carbon

Quick start

Get started by including Bootstrap's production-ready CSS and JavaScript via CDN without the need for any build steps. See it in practice with this [Bootstrap CodePen demo](#).

1. Create a new `index.html` file in your project root. Include the `<meta name="viewport">` tag as well for proper responsive behavior in mobile devices.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
```

On this page

- Quick start
- CDN links
- Next steps
- JS components
- Important globals
- HTML5 doctype
- Responsive meta tag
- Box-sizing
- Reboot
- Community

90°F Cloudy ENG IN 3:32 PM 8/20/2022

B Get started with Bootstrap · Boot x +

getbootstrap.com/docs/5.2/getting-started/introduction/

Docs Examples Icons Themes Blog Search CTRL K v5.2

Getting started

- Introduction
- Download
- Contents
- Browsers & devices
- JavaScript
- Webpack
- Parcel
- Vite
- Accessibility
- RFS
- RTL
- Contribute

Customize

- Overview
- Sass
- Options
- Color
- Components
- CSS variables
- Optimize

Quick start

Get started by including Bootstrap's production-ready CSS and JavaScript via CDN without the need for any build steps. See it in practice with this [Bootstrap CodePen demo](#).

1. Create a new `index.html` file in your project root. Include the `<meta name="viewport">` tag as well for proper responsive behavior in mobile devices.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

Copy to clipboard

On this page

- Quick start
- CDN links
- Next steps
- JS components
- Important globals
- HTML5 doctype
- Responsive meta tag
- Box-sizing
- Reboot
- Community

90°F Cloudy 3:35 PM 8/20/2022 ENG IN

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.htm

PythonDjango > products > templates

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango 1

- products
- migrations
- templates
 - index.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py

index.html

```
<h1>Products</h1>
<ul>
    {% for product in pro

```

New

- Cut Ctrl+X
- Copy Ctrl+C
- Copy Path/Reference...
- Paste Ctrl+V
- Find Usages Alt+F7
- Find in Files... Ctrl+Shift+F

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.htm

PythonDjango > products > templates

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango 1

- products
- migrations
- templates
 - index.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py

index.html

```
<h1>Product
<ul>
    {% for product in

```

New File

base.html

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - base.html

PythonDjango > products > templates > base.html

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango 1

- products
- migrations
- templates
 - base.html
 - index.html

base.html

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Bootstrap demo</title>
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - base.html

PythonDjango > products > templates > base.html

Project

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Bootstrap demo</title>
    </head>
    <body>
        {% block content %}
        {% endblock %}
    </body>
</html>
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango > products > templates > index.html

Project

```
{% extends 'base.html' %}

{% block content %}
    <h1>Products</h1>
    <ul>
        {% for product in products %}
            <li>{{ product.name }} (Rs. {{ product.price }}) </li>
        {% endfor %}
    </ul>
    {% endblock %}
```

Select offer to change | Django s

127.0.0.1:8000/products/

Products

- Orage (Rs. 70.0)
- Strawberry (Rs. 120.0)
- Banana (Rs. 20.0)
- Apple (Rs. 120.0)
- Grapes (Rs. 130.0)

B Cards · Bootstrap v5.2 +

getbootstrap.com/docs/5.2/components/card/

Docs Examples Icons Themes Blog Search CTRL K v5.2

Floating labels Layout Validation Components Accordion Alerts Badge Breadcrumb Buttons Button group Card Carousel Close button Collapse Dropdowns List group Modal Navbar Navs & tabs Offcanvas Pagination Placeholders

Cards

Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.

View on GitHub

On this page

- About
- Example
- Content types
 - Body
 - Titles, text, and links
 - Images
 - List groups
 - Kitchen sink
 - Header and footer
- Sizing
 - Using grid markup
 - Using utilities
 - Using custom CSS
- Text alignment
- Navigation
 - Images
 - Image caps
 - Image overlays
- Horizontal
- Card styles
 - Background and color
 - Border
 - Mixins utilities
- Card layout

Join us for AWS Innovate and learn to scale your applications and innovate with AWS Global Infrastructure

ads via Carbon

About

A **card** is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. If you're familiar with Bootstrap 3, cards replace our old panels, wells, and thumbnails. Similar functionality to those components is available as modifier classes for cards.

Example

Cards are built with as little markup and styles as possible, but still manage to deliver a ton of control and customization. Built with flexbox, they offer easy alignment and mix well with other Bootstrap components. They have no `margin` by default, so use `spacing utilities` as needed.

90°F Cloudy

Cloud ENG IN 4:09 PM 8/20/2022

Copy to clipboard

HTML⚡📋

```
<div class="card" style="width: 18rem;">
    
    <div class="card-body">
        <h5 class="card-title">Card title</h5>
        <p class="card-text">Some quick example text to build on the card title and make up the bulk of
            <a href="#" class="btn btn-primary">Go somewhere</a>
        </div>
    </div>
```

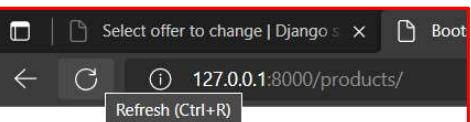
Project+-☰⚙️

index.html

```
{% extends 'base.html' %}

{% block content %}
    <h1>Products</h1>
    <div class="row">
        {% for product in products %}
            <div class="col">
                <div class="card" style="width: 18rem;">
                    
                    <div class="card-body">
                        <h5 class="card-title">Card title</h5>
                        <p class="card-text">Some quick example text to build on the card title and make up the bulk of the
                            card's content.</p>
                        <a href="#" class="btn btn-primary">Go somewhere</a>
                    </div>
                </div>
            </div>
        {% endfor %}
    </div>
    {% endblock %}
```

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjangoproductsmigrationstemplatesbase.htmlindex.html__init__.pyadmin.pyapps.pymodels.pytests.pyurls.pyviews.pyPythonDjangovenv library rootdb.sqlite3main.pymanage.pyExternal LibrariesScratches and Consoles



Products



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)

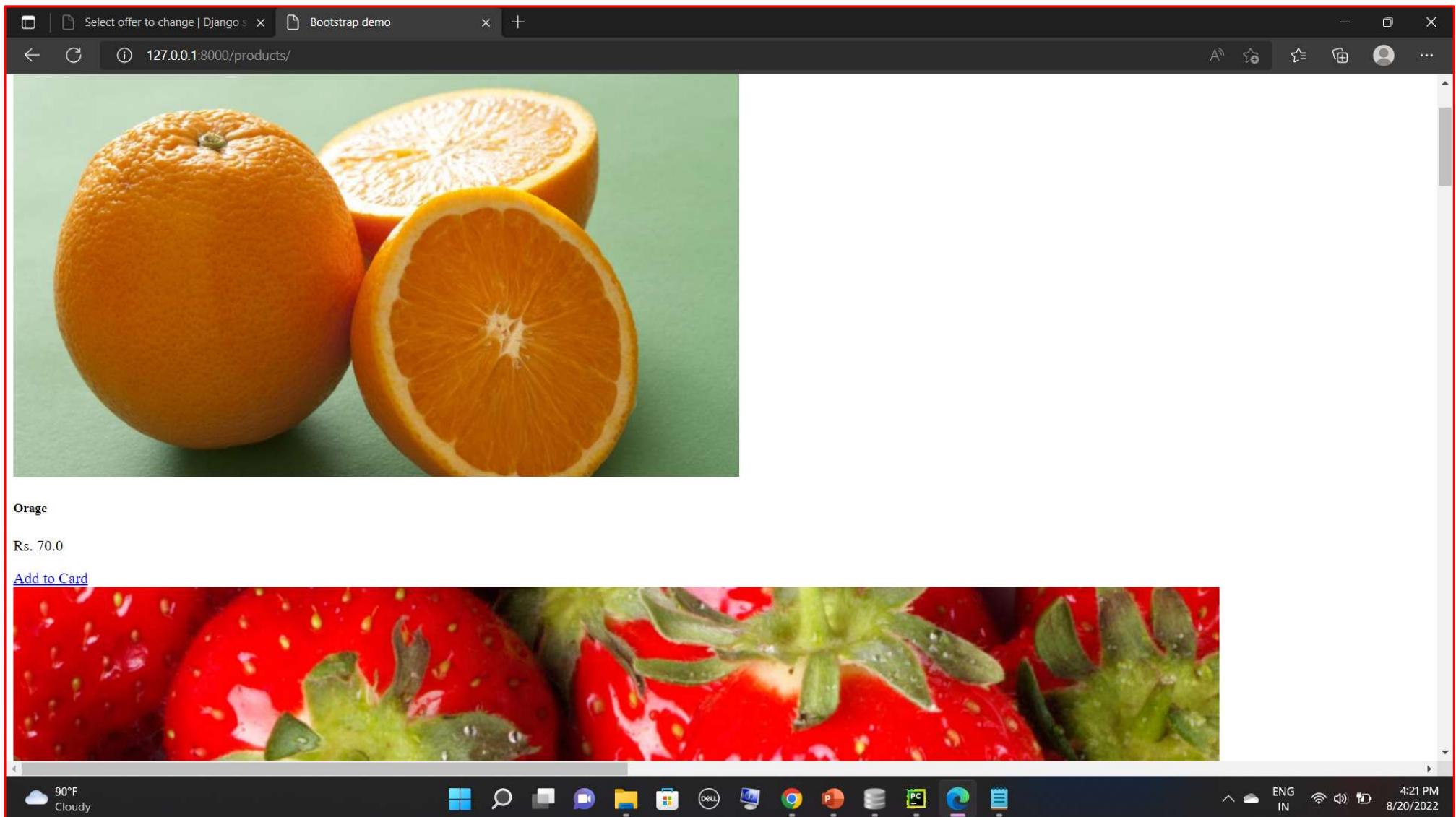


Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

```
{% extends 'base.html' %}

{% block content %}
    <h1>Products</h1>
    <div class="row">
        {% for product in products %}
            <div class="col">
                <div class="card" style="width: 18rem;">
                    
                    <div class="card-body">
                        <h5 class="card-title"> {{ product.name }} </h5>
                        <p class="card-text"> Rs. {{ product.price }} </p>
                        <a href="#" class="btn btn-primary"> Add to Card </a>
                    </div>
                </div>
            </div>
        {% endfor %}
    </div>
    {% endblock %}
```



Navbar · Bootstrap v5.2

getbootstrap.com/docs/5.2/components/navbar/

Docs Examples Icons Themes Blog Search CTRL K v5.2

Brand

The `.navbar-brand` can be applied to most elements, but an anchor works best, as some elements might require utility classes or custom styles.

Text

Add your text within an element with the `.navbar-brand` class.

Navbar

Navbar

HTML

```
<!-- As a link -->
<nav class="navbar bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
  </div>
</nav>

<!-- As a heading -->
<nav class="navbar bg-light">
  <div class="container-fluid">
    <span class="navbar-brand mb-0 h1">Navbar</span>
  </div>
```

On this page

How it works

Supported content

Brand

Text

Image

Image and text

Nav

Forms

Text

Color schemes

Containers

Placement

Scrolling

Responsive behaviors

Toggler

External content

Offcanvas

CSS

Variables

Sass variables

Sass loop

90°F Cloudy

4:26 PM 8/20/2022

```
ex.html x base.html x
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World</title>
  </head>
  <body>
    <!-- As a link -->
    <nav class="navbar bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Python Project</a>
      </div>
    </nav>
    {% block content %}
    {% endblock %}
  </body>
</html>
```

