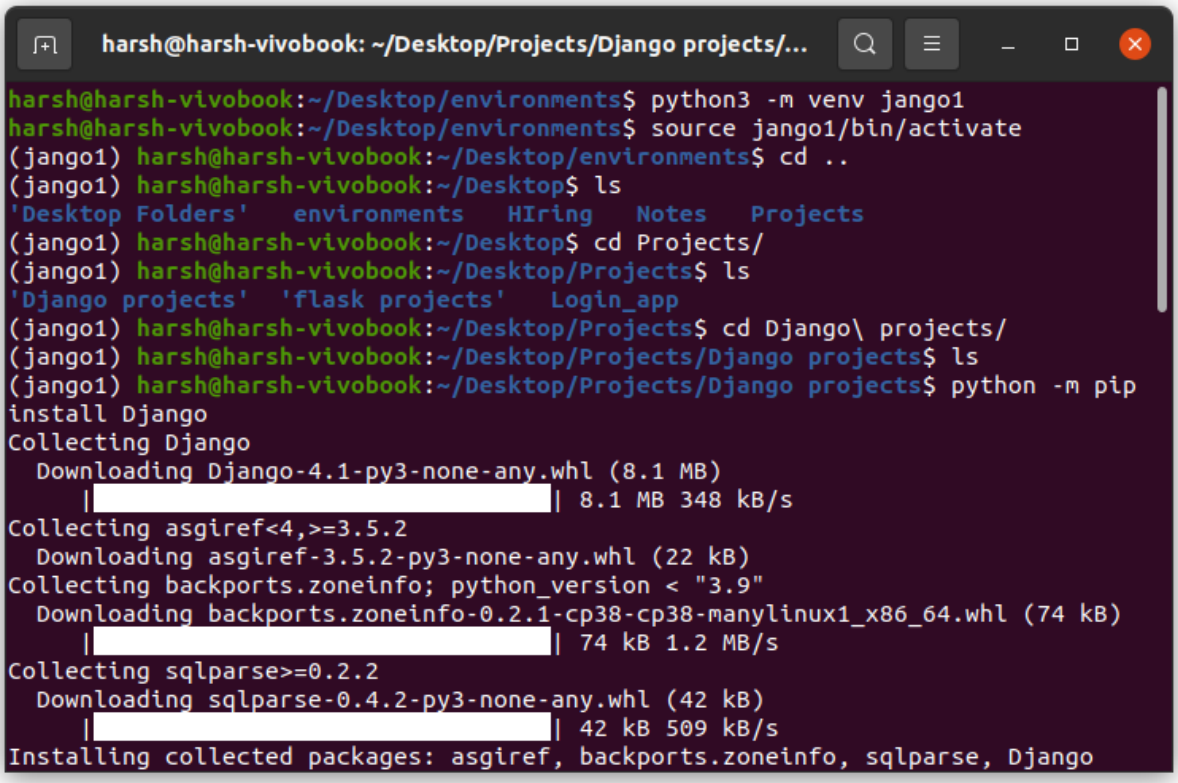
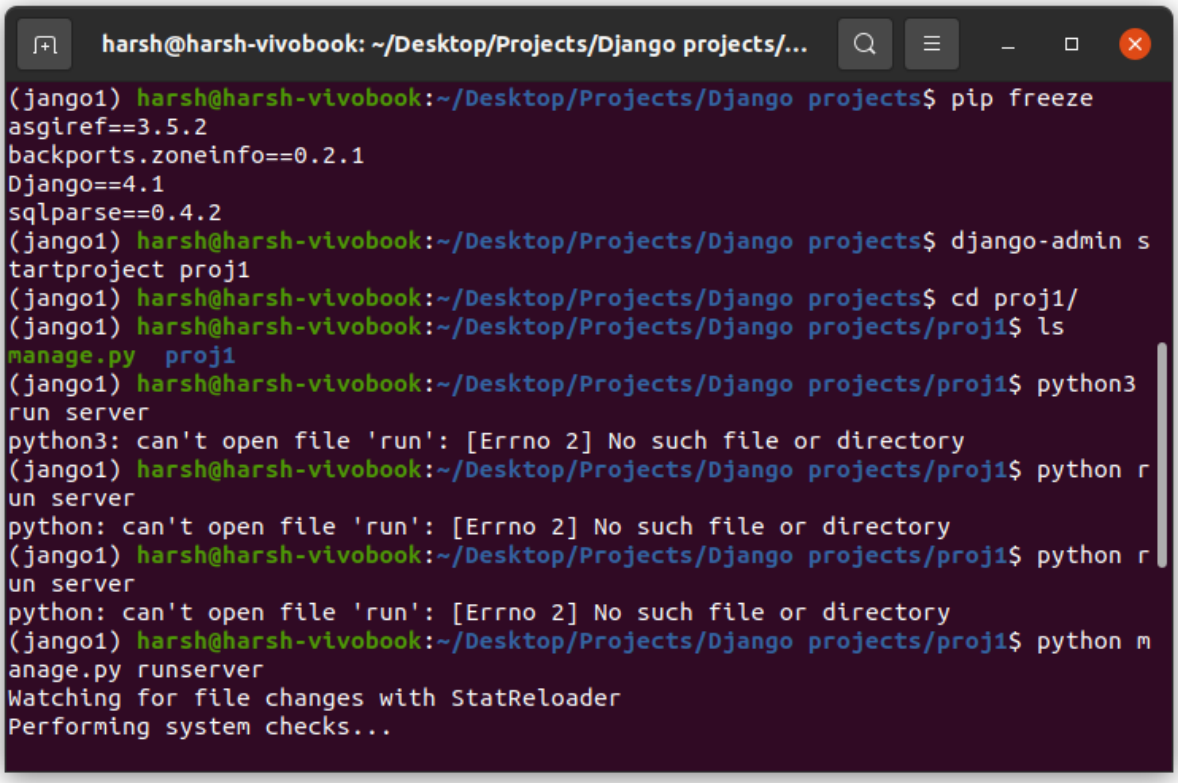
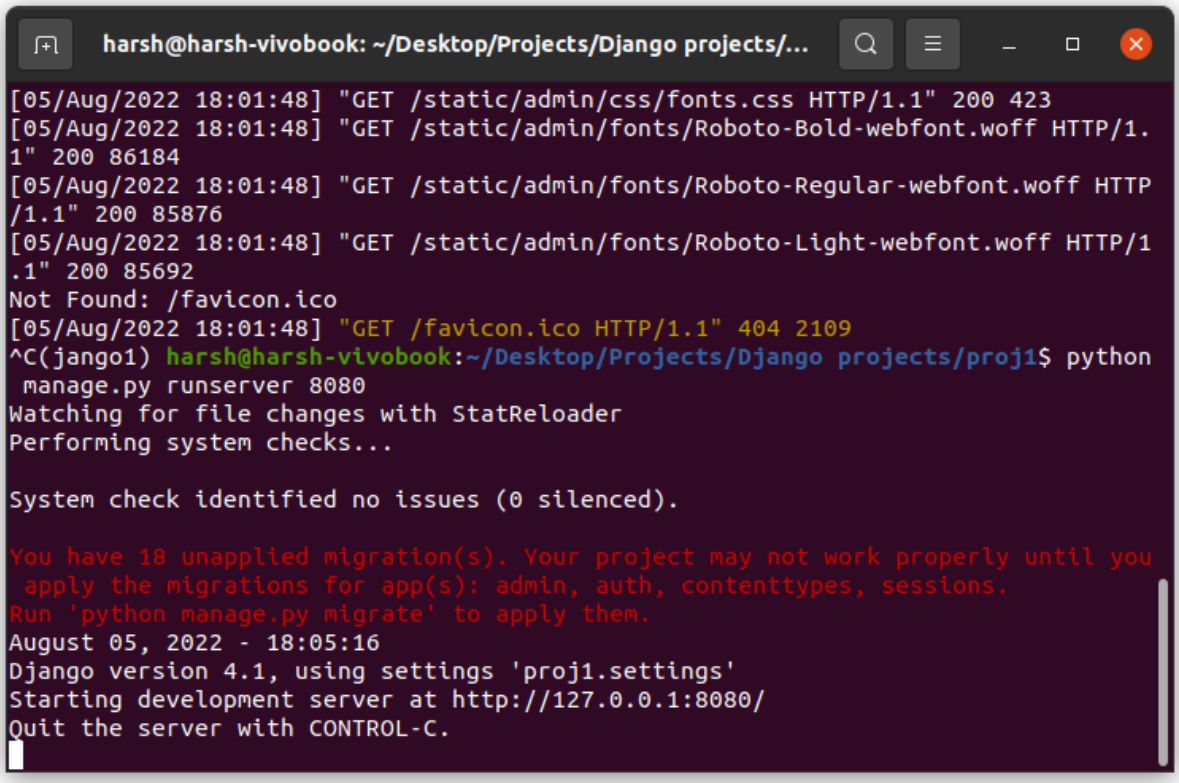
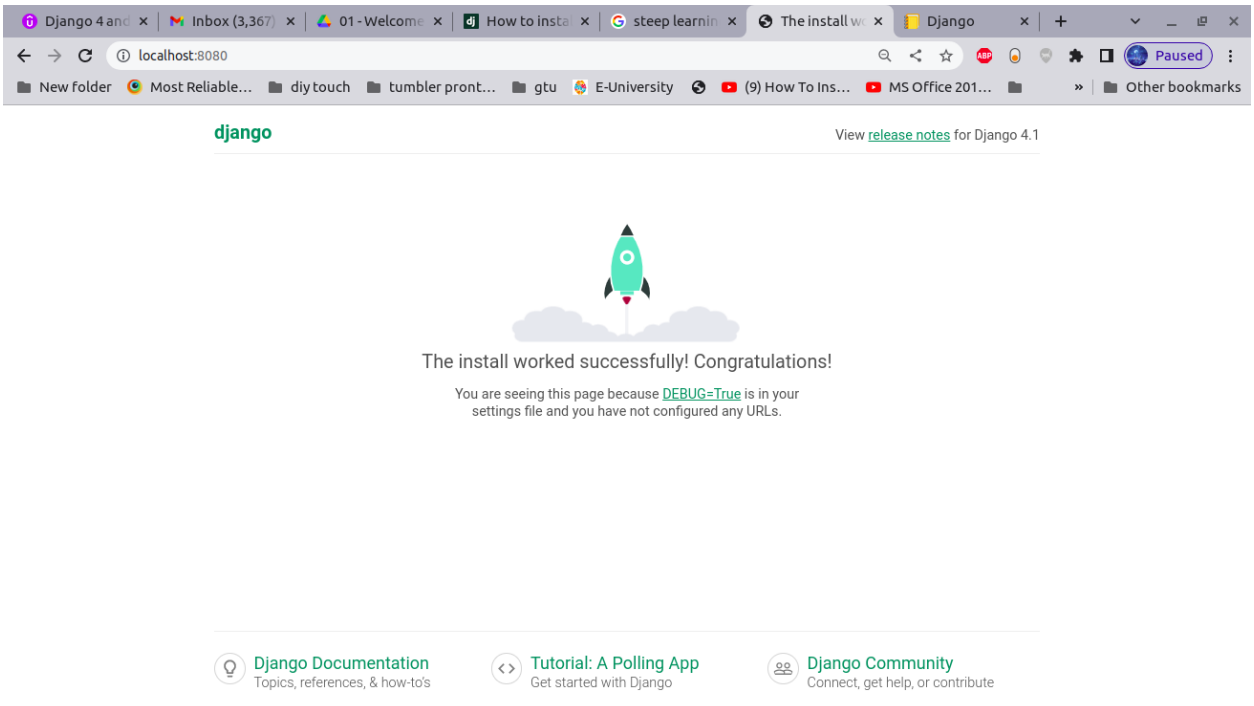
Django


harsh@harsh-vivobook: ~/Desktop/Projects/Django projects/...
harsh@harsh-vivobook:~/Desktop/environments$ python3 -m venv jango1
harsh@harsh-vivobook:~/Desktop/environments$ source jango1/bin/activate
(jango1) harsh@harsh-vivobook:~/Desktop/environments$ cd ..
(jango1) harsh@harsh-vivobook:~/Desktop$ ls
'Desktop Folders' environments HIring Notes Projects
(jango1) harsh@harsh-vivobook:~/Desktop$ cd Projects/
(jango1) harsh@harsh-vivobook:~/Desktop/Projects$ ls
'Django projects' 'flask projects' Login_app
(jango1) harsh@harsh-vivobook:~/Desktop/Projects$ cd Django\ projects/
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects$ ls
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects$ python -m pip
install Django
Collecting Django
Downloading Django-4.1-py3-none-any.whl (8.1 MB)
| 8.1 MB 348 kB/s
Collecting asgiref<4,>=3.5.2
Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting backports.zoneinfo; python_version < "3.9"
Downloading backports.zoneinfo-0.2.1-cp38-cp38-manylinux1_x86_64.whl (74 kB)
| 74 kB 1.2 MB/s
Collecting sqlparse>=0.2.2
Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
| 42 kB 509 kB/s
Installing collected packages: asgiref, backports.zoneinfo, sqlparse, Django

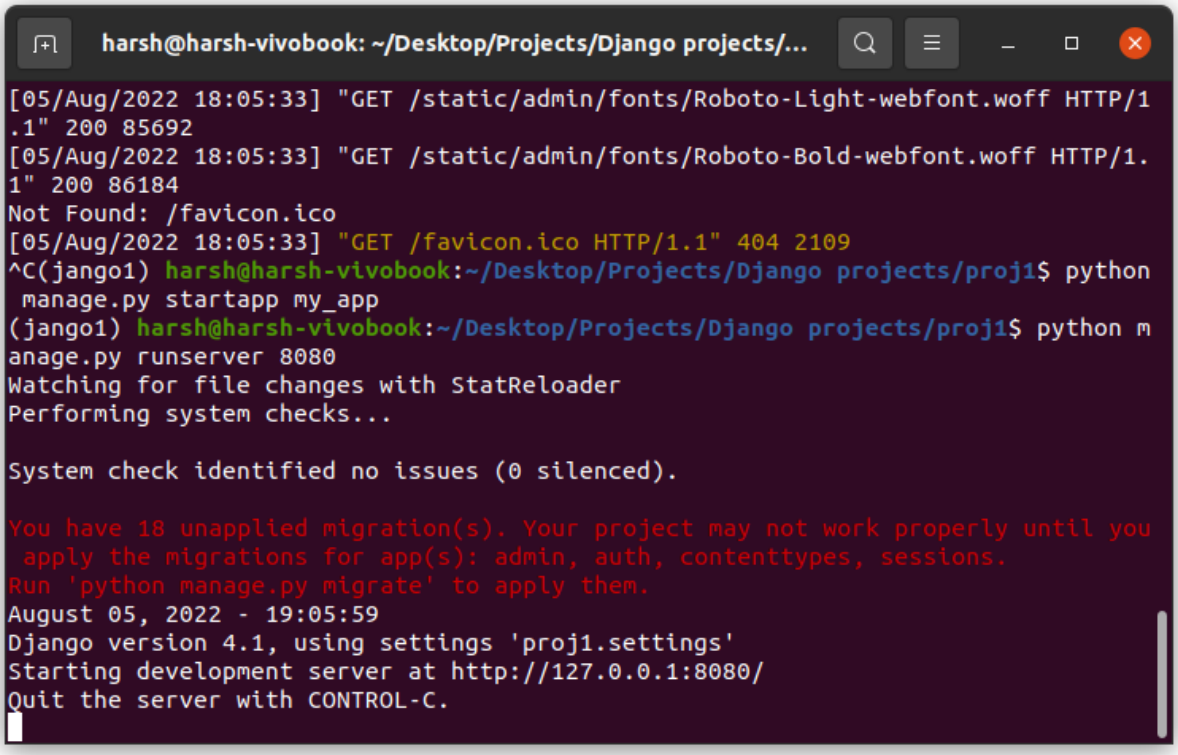


Changing the default port to run server

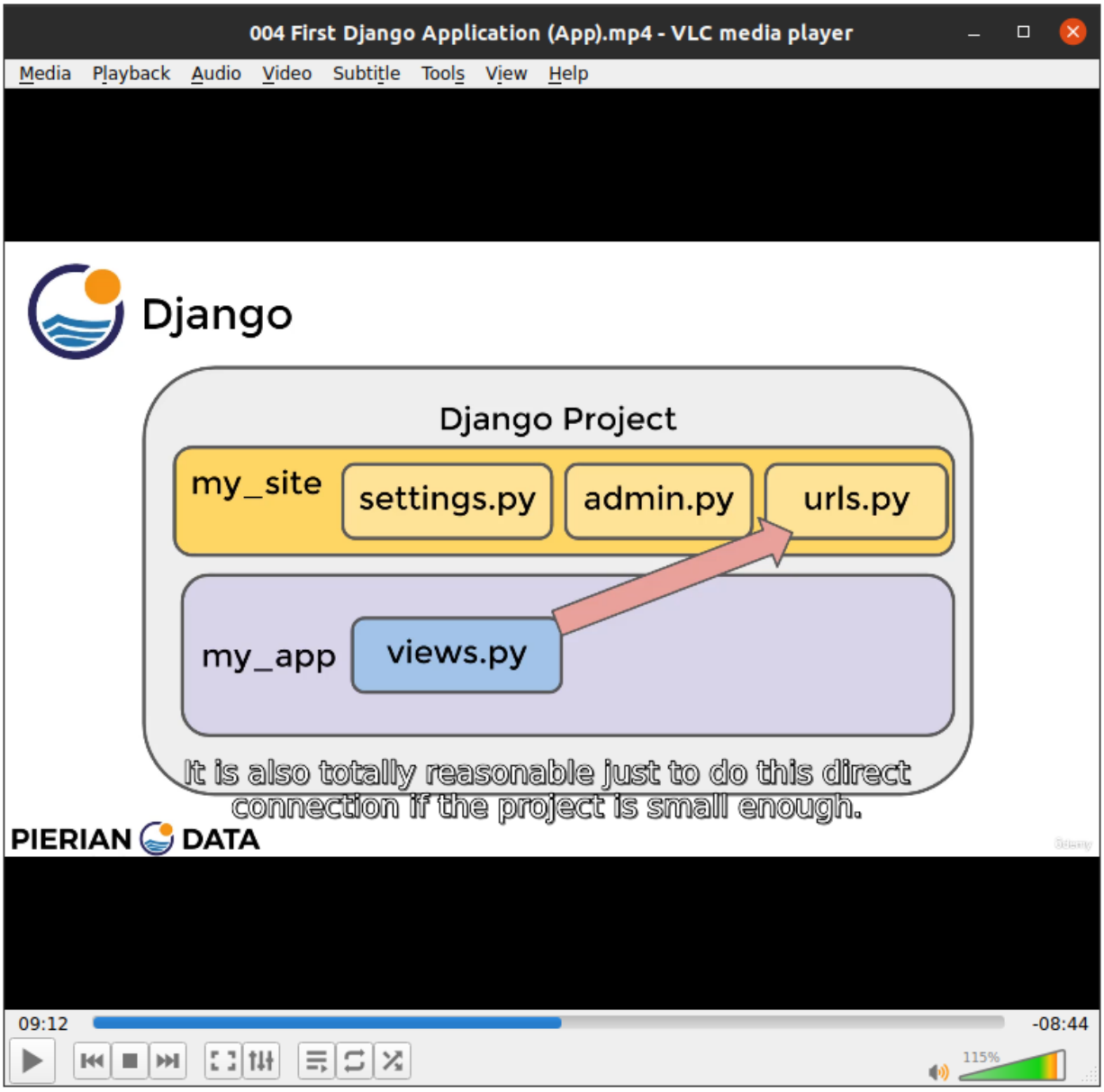

harsh@harsh-vivobook: ~/Desktop/Projects/Django projects/...
[05/Aug/2022 18:01:48] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[05/Aug/2022 18:01:48] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 86184
[05/Aug/2022 18:01:48] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 85876
[05/Aug/2022 18:01:48] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 85692
Not Found: /favicon.ico
[05/Aug/2022 18:01:48] "GET /favicon.ico HTTP/1.1" 404 2109
^C(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/proj1$ python manage.py runserver 8080
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 05, 2022 - 18:05:16
Django version 4.1, using settings 'proj1.settings'
Starting development server at http://127.0.0.1:8080/
Quit the server with CONTROL-C.


localhost:8080
Paused
New folder
Most Reliable...
diy touch
tumbler pront...
gtu
E-University
(9) How To Ins...
MS Office 201...
Other bookmarks
django
View release notes for Django 4.1
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

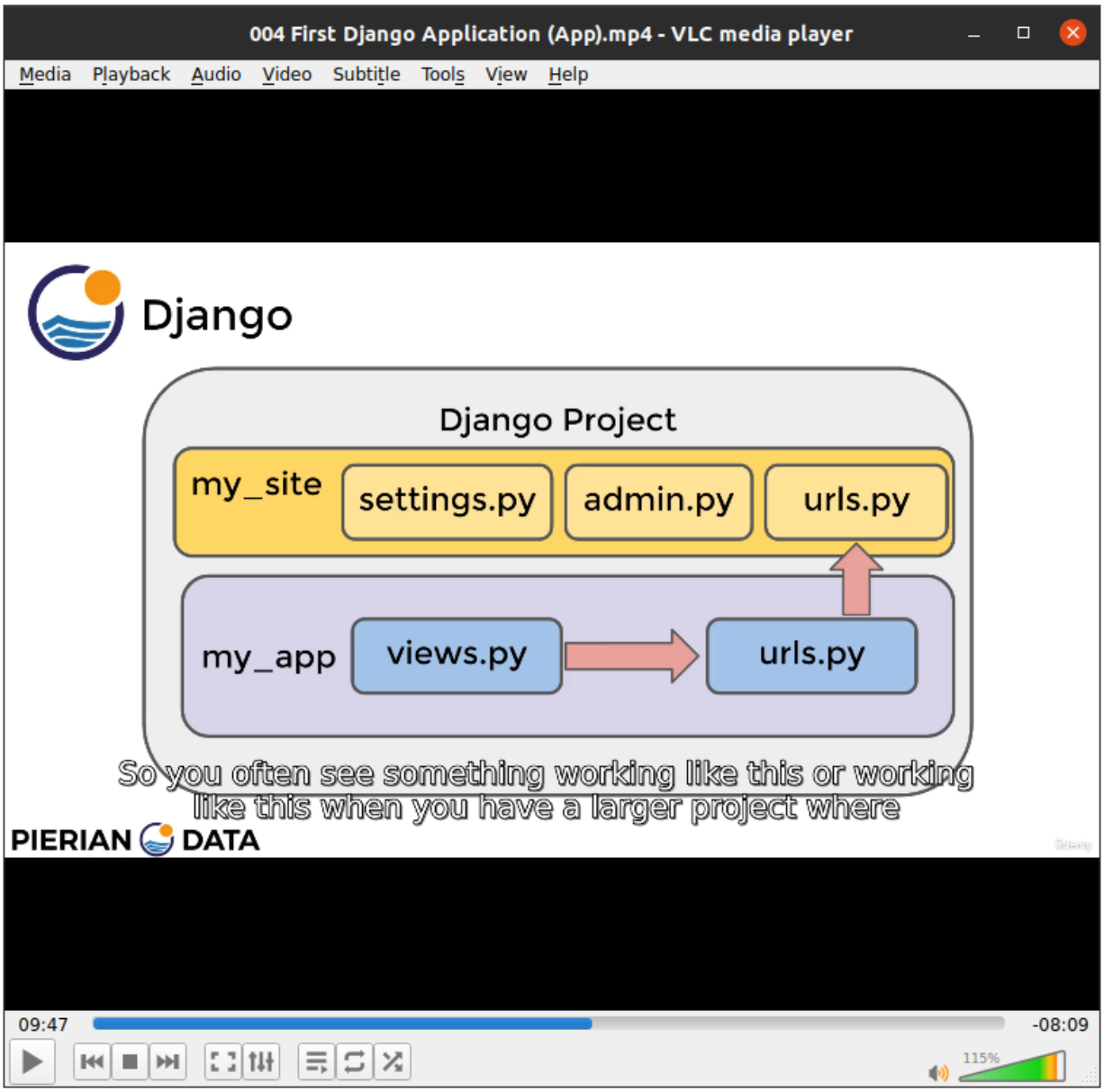
creating a new app

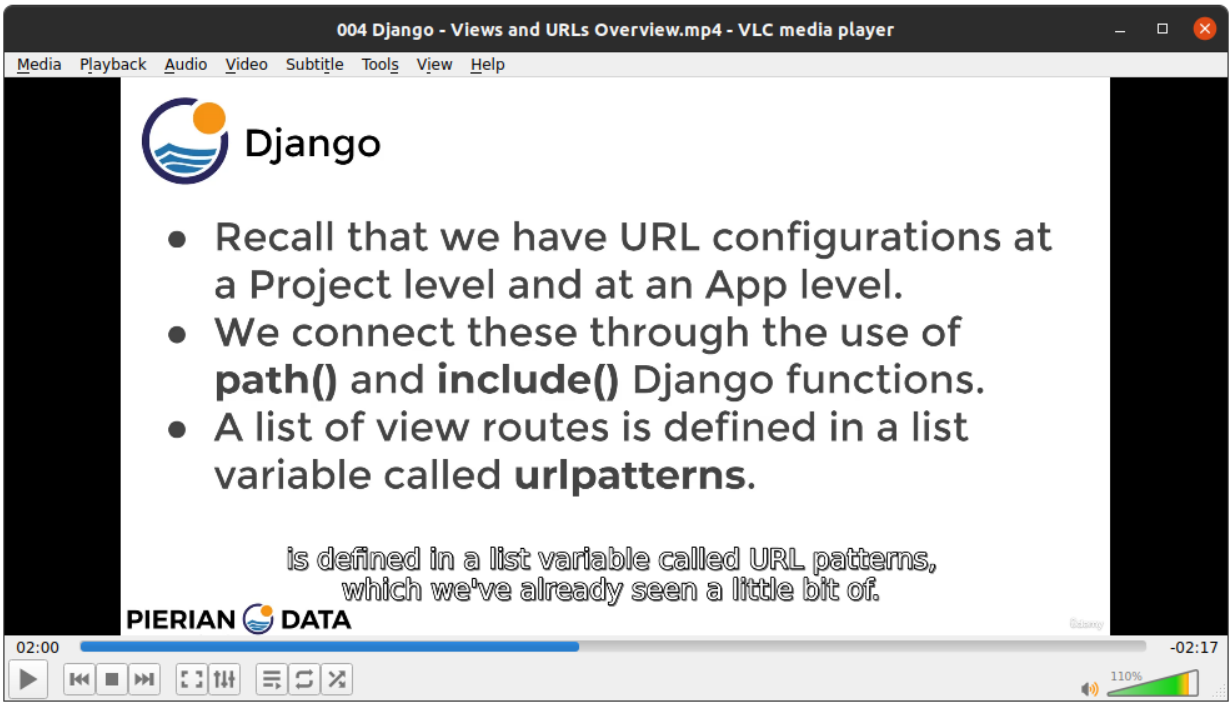


Small project


004 First Django Application (App).mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
Django
Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py
It is also totally reasonable just to do this direct connection if the project is small enough.
PIERIAN DATA
09:12
-08:44
115%

Big project


004 First Django Application (App).mp4 - VLC media player
Media Playback Audio Video Subtitle Tools View Help
Django
Django Project
my_site
settings.py
admin.py
urls.py
my_app
views.py
urls.py
So you often see something working like this or working like this when you have a larger project where
PIERIAN DATA
09:47
-08:09
115%



The Web Server Gateway Interface
(WSGI, pronounced whiskey or WIZ-ghee)

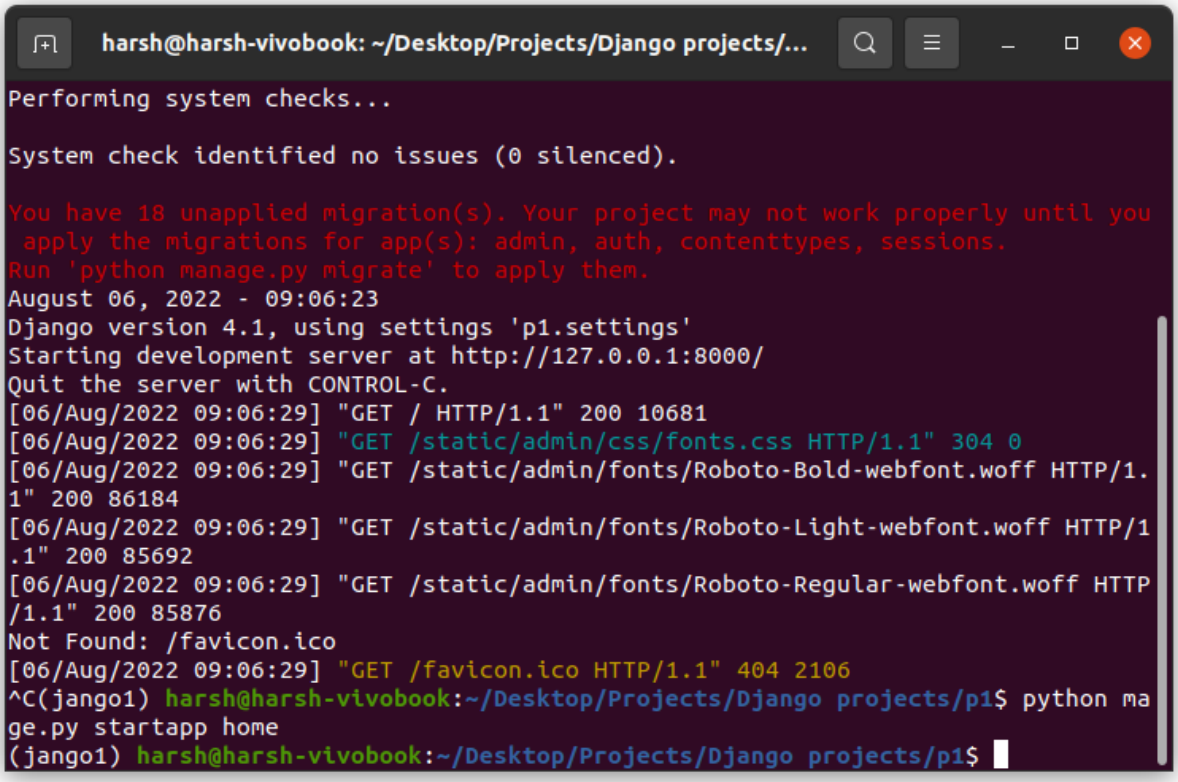
basically, **provides a standard interface between web applications written in Python and Webservers**
. That means, WSGI gives portability to your Python Web Application across many different Web Servers, without any additional configurations on your NGINX, Apache, etc.

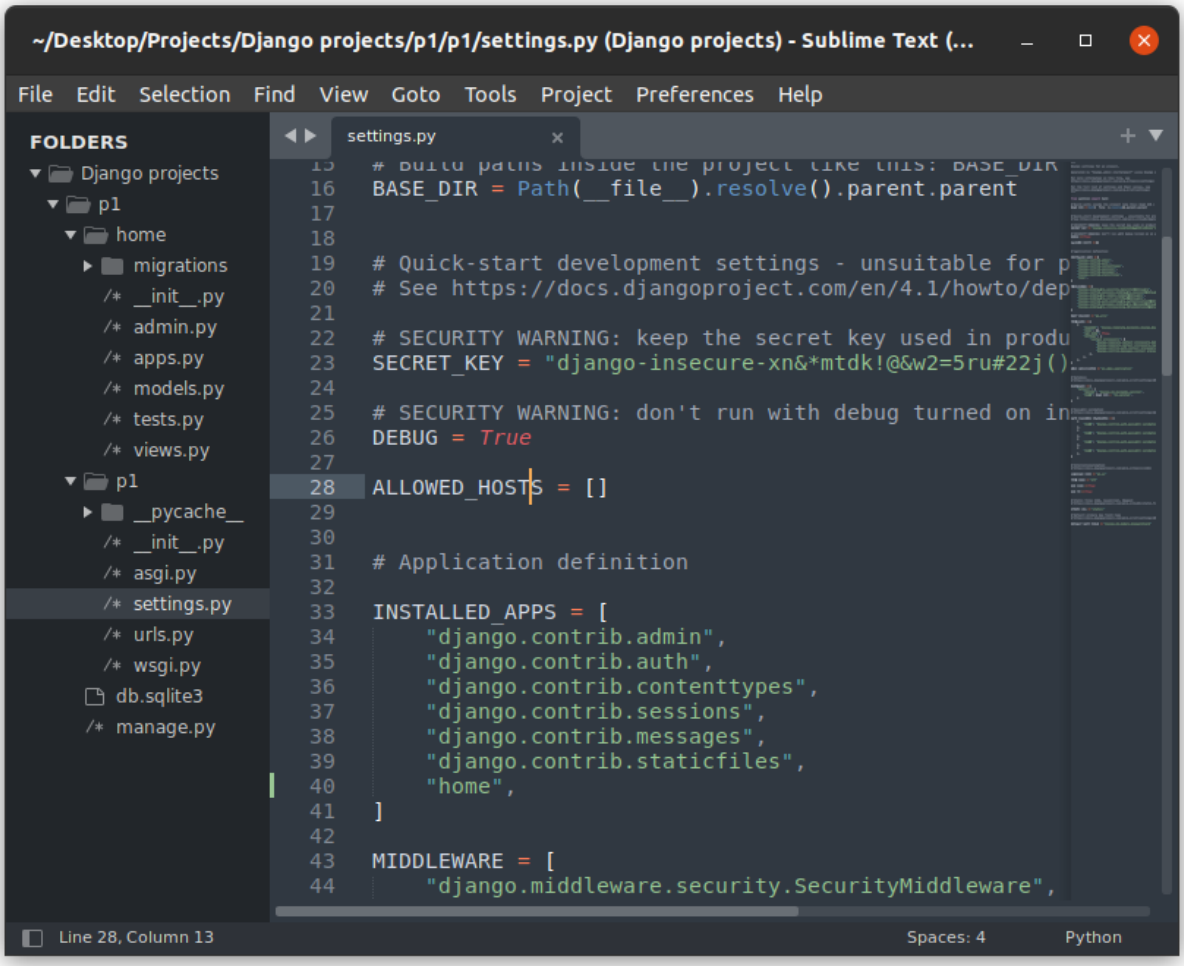
- http request use ab use krna hai

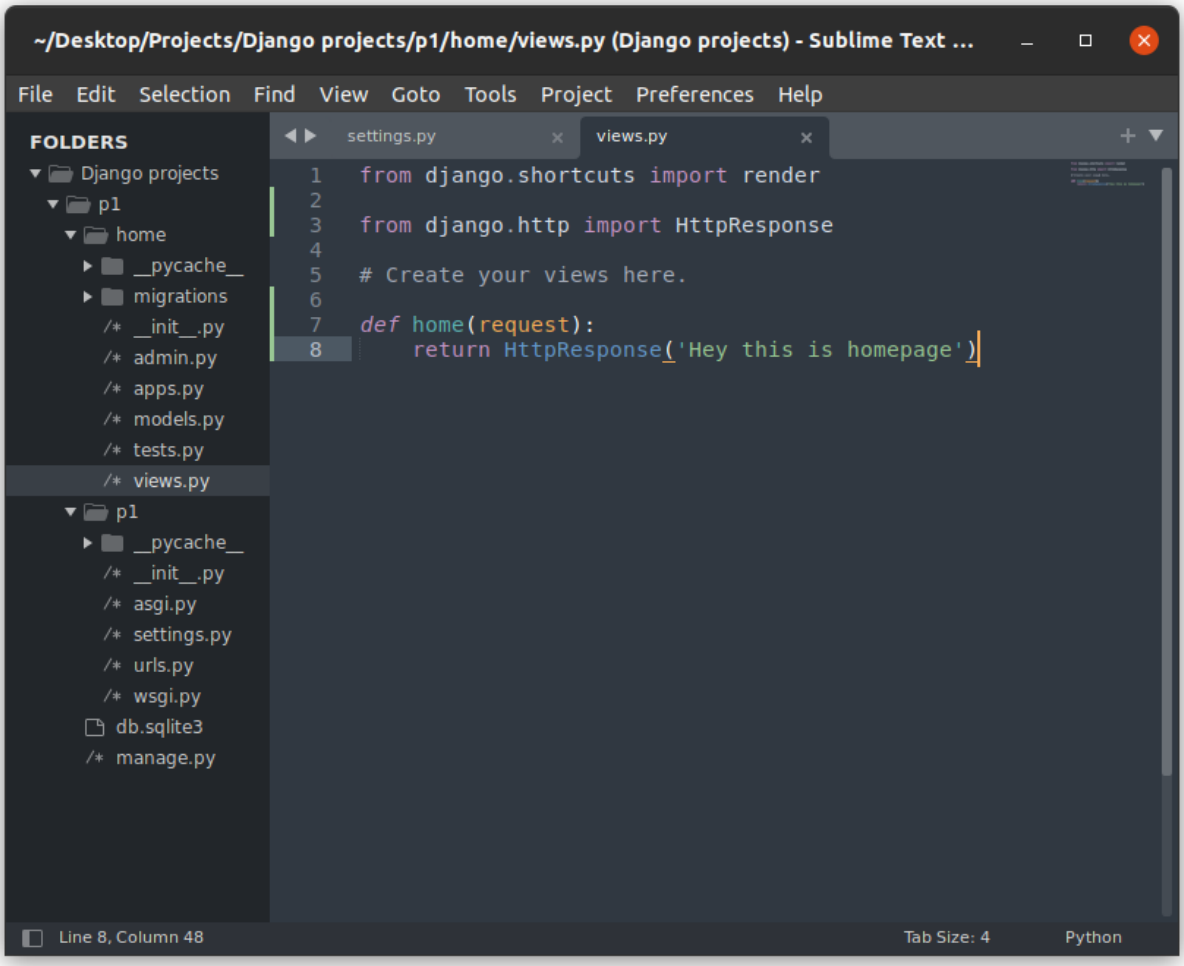
The Asynchronous Server Gateway Interface (ASGI) is **a calling convention for web servers to forward requests to asynchronous-capable Python programming language frameworks, and applications**
. It is built as a successor to the Web Server Gateway Interface (WSGI).

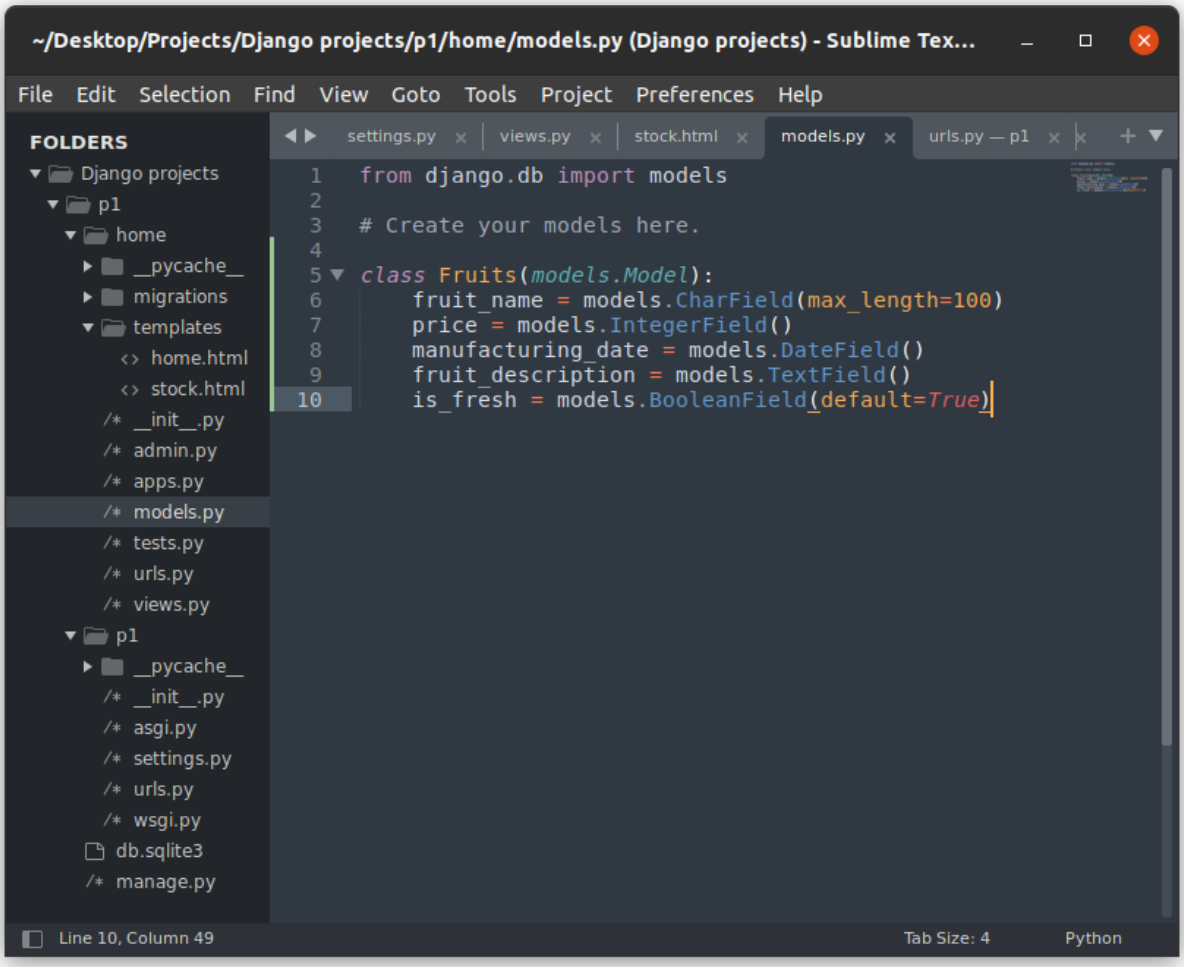
- http request aur web sockets tab ye use krte hai

crating a new app and adding it to setting.py of main proj

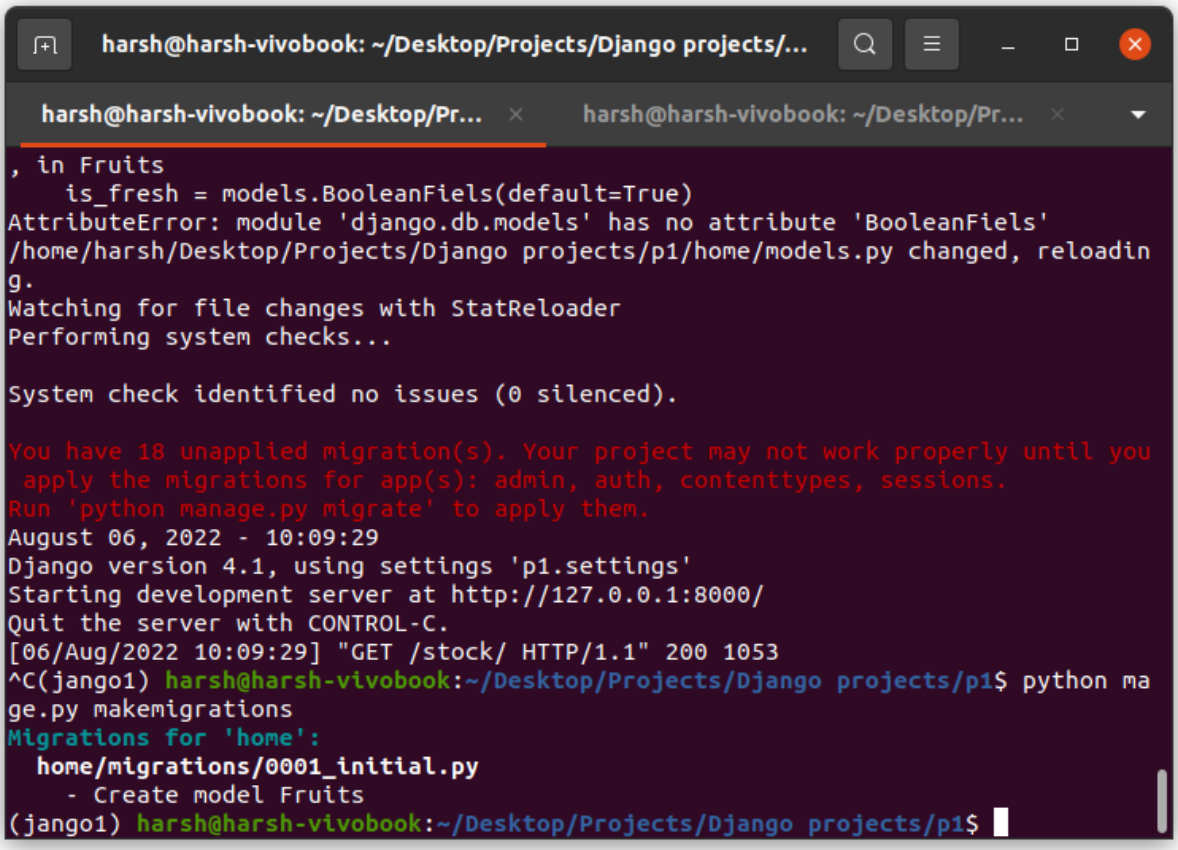



~/Desktop/Projects/Django projects/p1/p1/settings.py (Django projects) - Sublime Text (...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
settings.py
# Build paths inside the project like this: BASE_DIR
BASE_DIR = Path(__file__).resolve().parent.parent
# Quick-start development settings - unsuitable for p
# See https://docs.djangoproject.com/en/4.1/howto/dep
# SECURITY WARNING: keep the secret key used in produ
SECRET_KEY = "django-insecure-xn&*mtdk!@&w2=5ru#22j()
# SECURITY WARNING: don't run with debug turned on in
DEBUG = True
ALLOWED_HOSTS = []
# Application definition
INSTALLED_APPS = [
"django.contrib.admin",
"django.contrib.auth",
"django.contrib.contenttypes",
"django.contrib.sessions",
"django.contrib.messages",
"django.contrib.staticfiles",
"home",
]
MIDDLEWARE = [
"django.middleware.security.SecurityMiddleware",
Line 28, Column 13
Spaces: 4
Python


~/Desktop/Projects/Django projects/p1/home/views.py (Django projects) - Sublime Text ...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
settings.py
views.py
1 from django.shortcuts import render
2
3 from django.http import HttpResponse
4
5 # Create your views here.
6
7 def home(request):
8 return HttpResponse('Hey this is homepage')
Line 8, Column 48
Tab Size: 4
Python

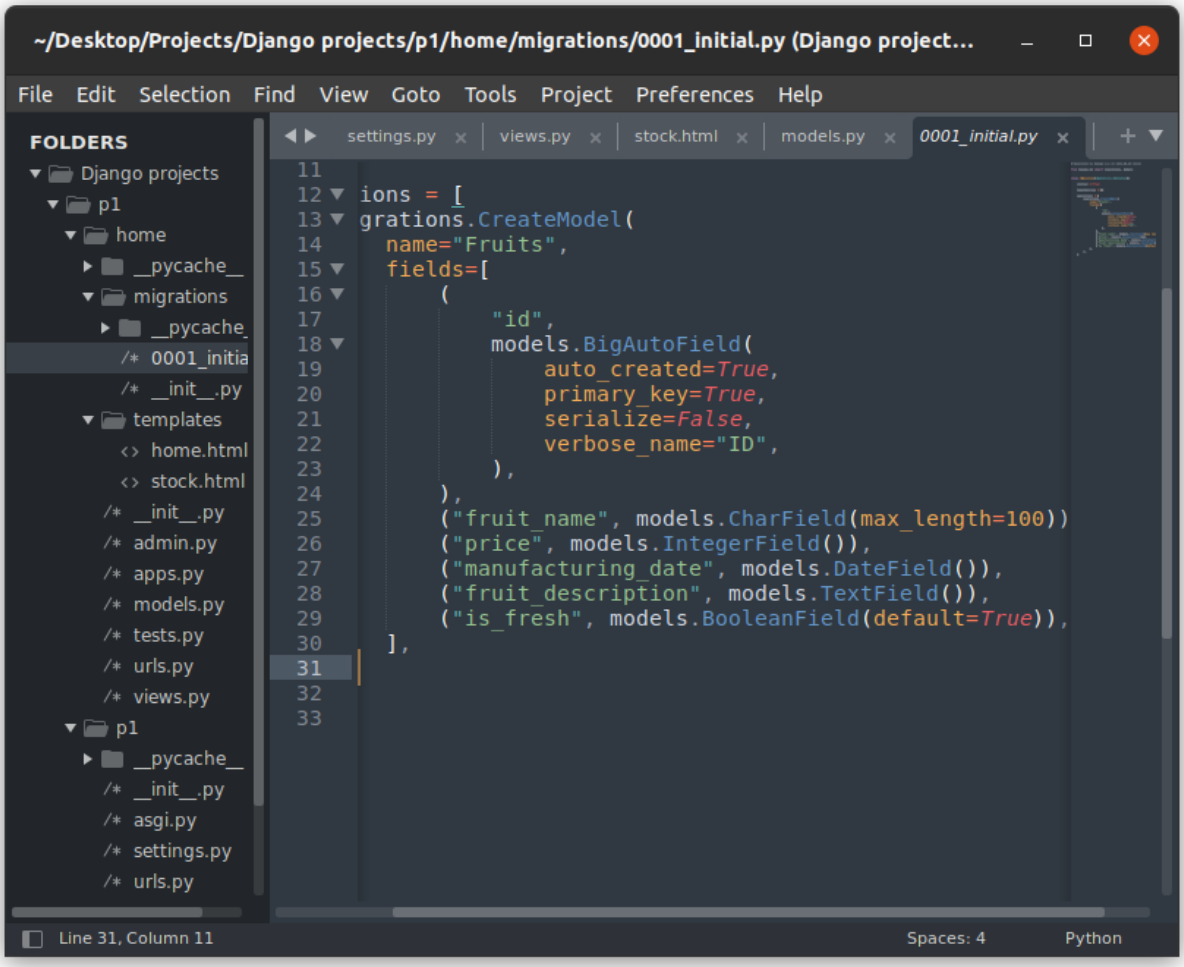


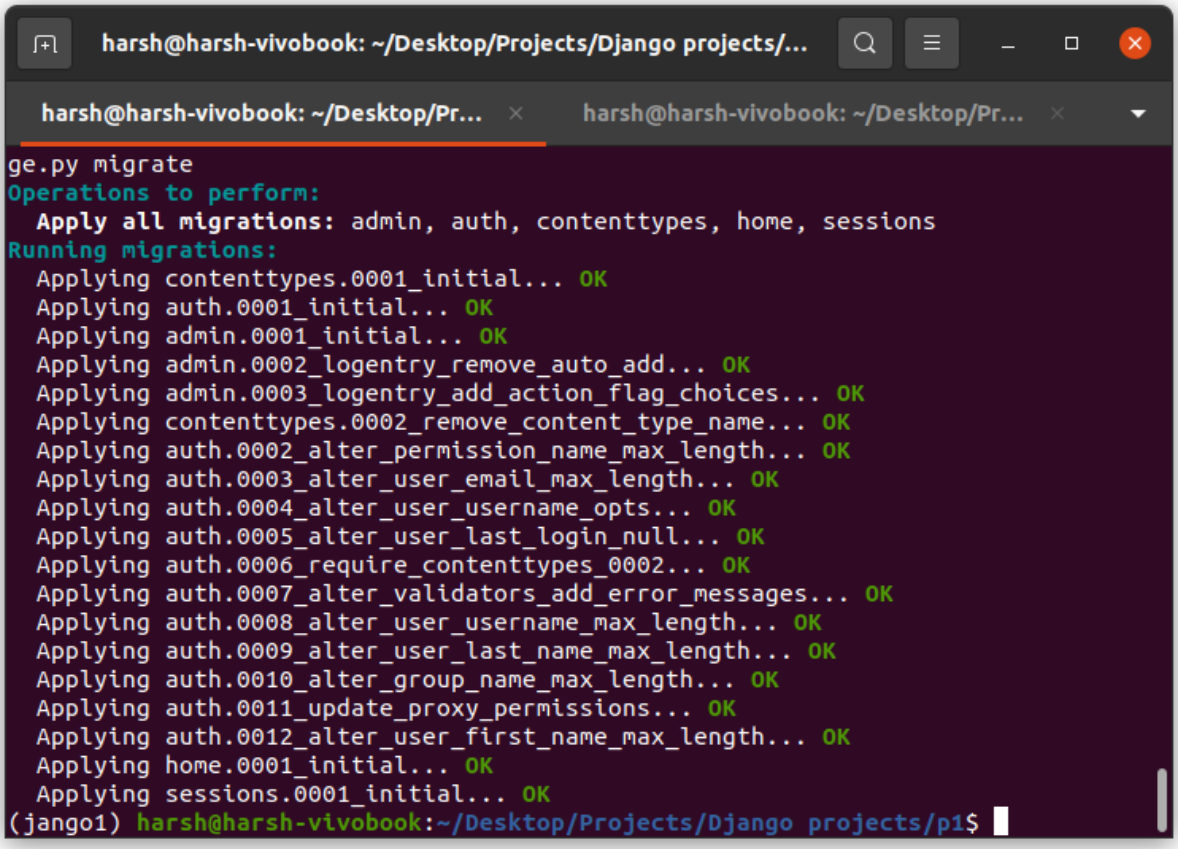
After adding the models we need to make migrations to save the changes



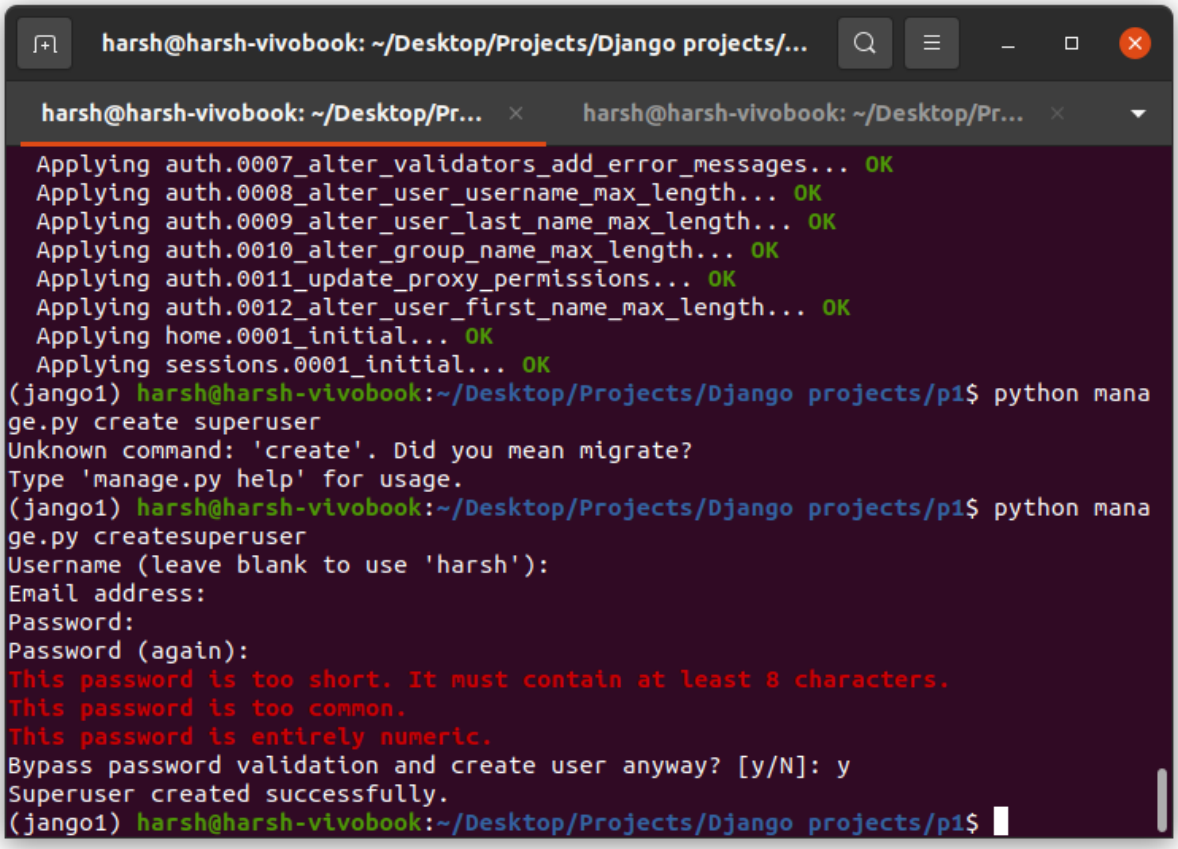
After making the migration a mirgrate folder is generated

now we need to run a command to save this migrations

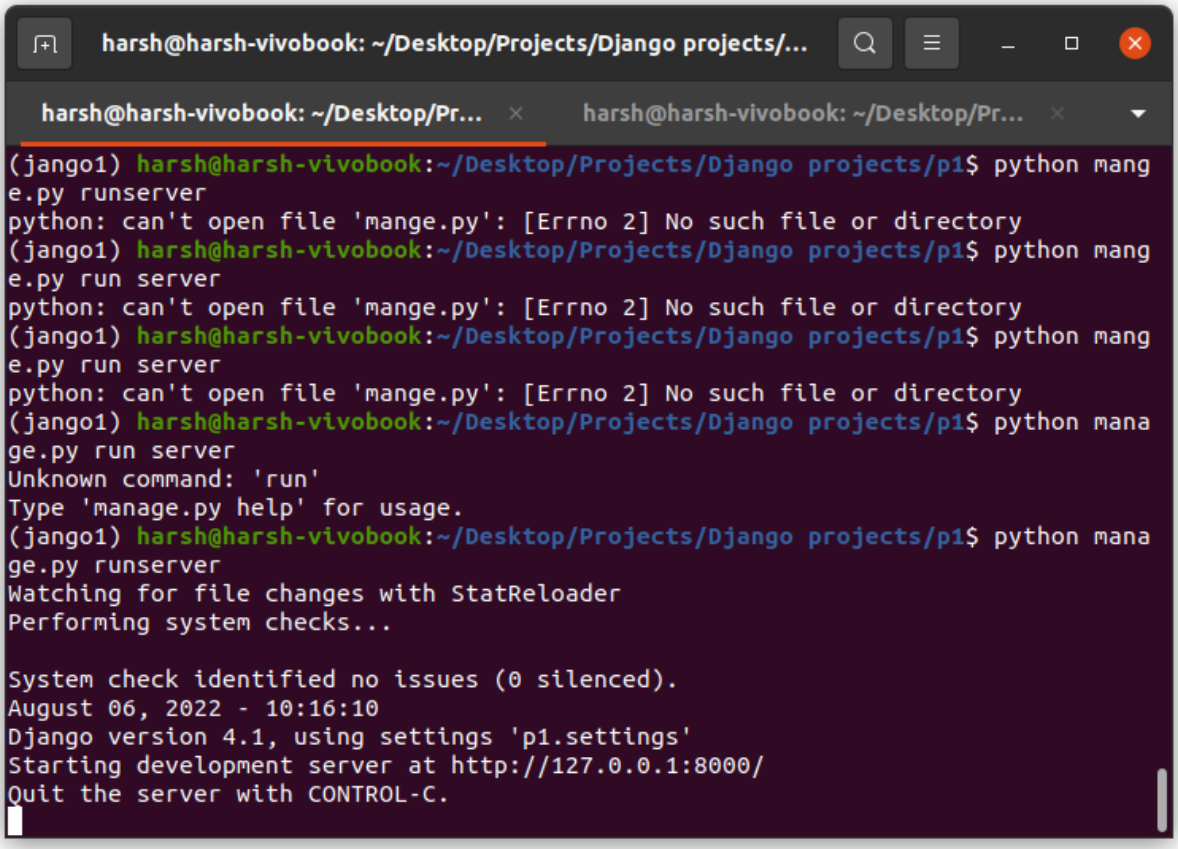

~/Desktop/Projects/Django projects/p1/home/migrations/0001_initial.py (Django project...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__pycache__
0001_initia
__init__.py
templates
home.html
stock.html
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
settings.py
views.py
stock.html
models.py
0001_initial.py
ions = [
grations.CreateModel(
name="Fruits",
fields=[
("id",
models.BigAutoField(
auto_created=True,
primary_key=True,
serialize=False,
verbose_name="ID",
),
),
("fruit_name", models.CharField(max_length=100))
("price", models.IntegerField()),
("manufacturing_date", models.DateField()),
("fruit_description", models.TextField()),
("is_fresh", models.BooleanField(default=True)),
],
Line 31, Column 11
Spaces: 4
Python

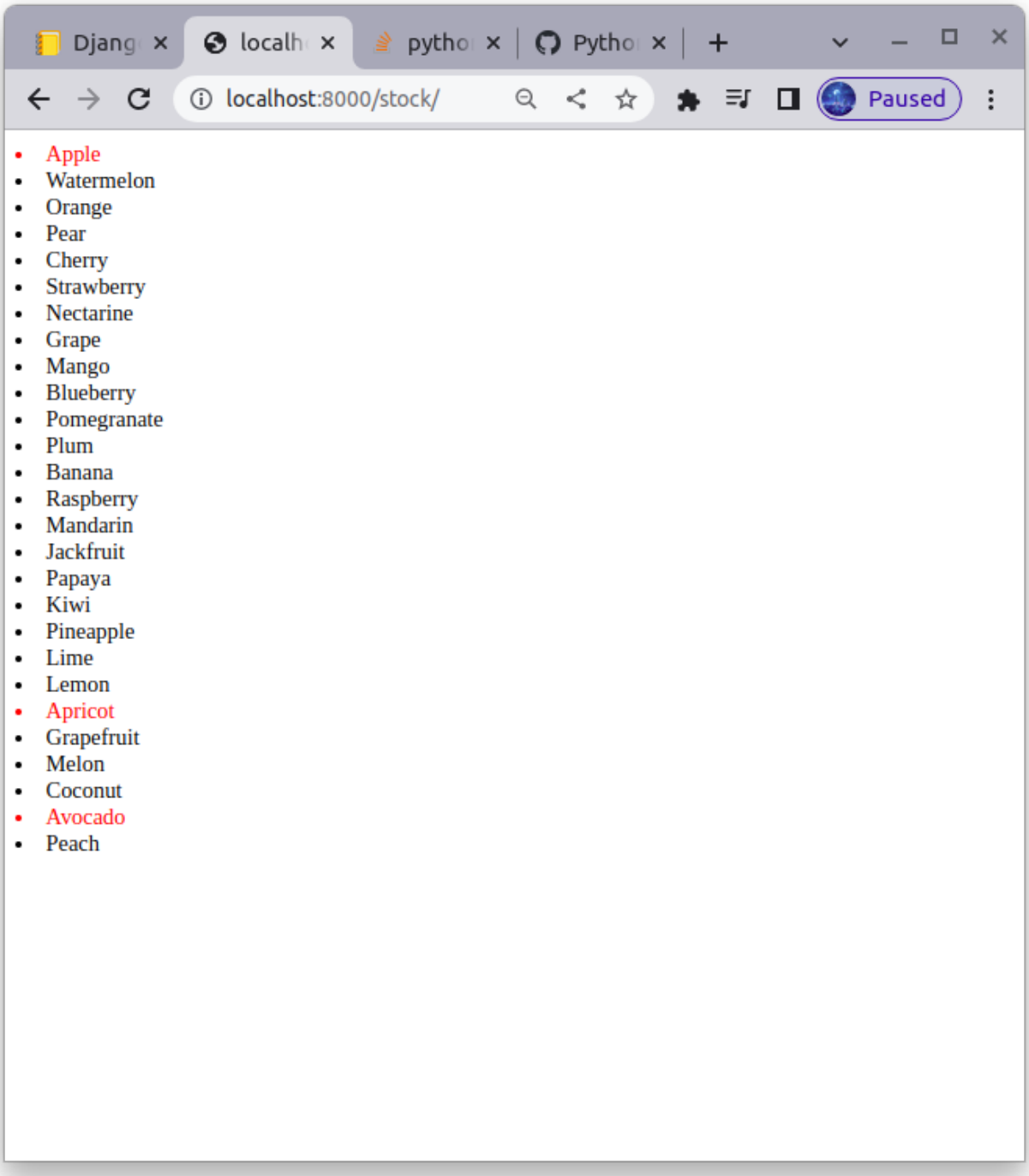


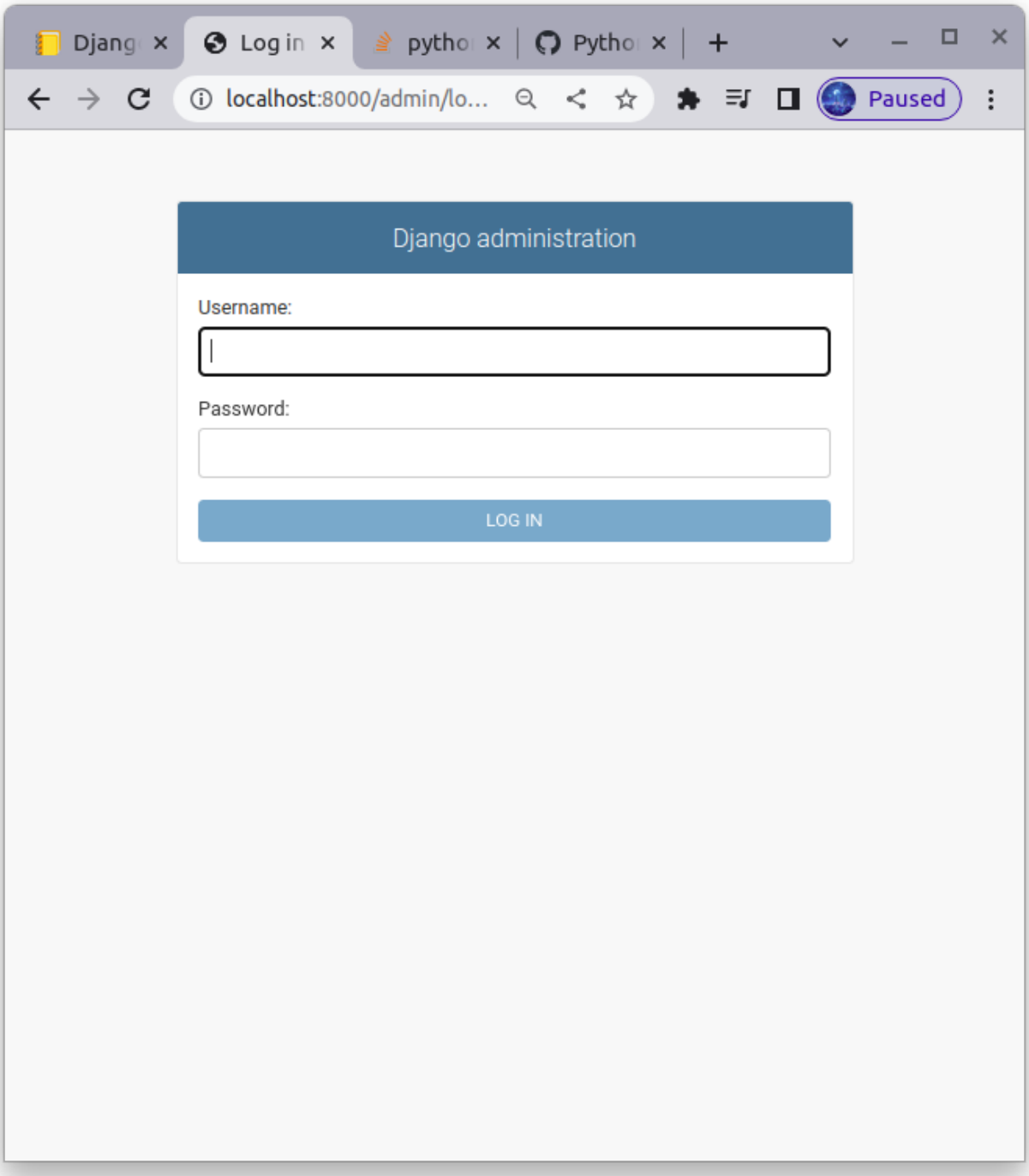
Now for creating a admin for the admin login

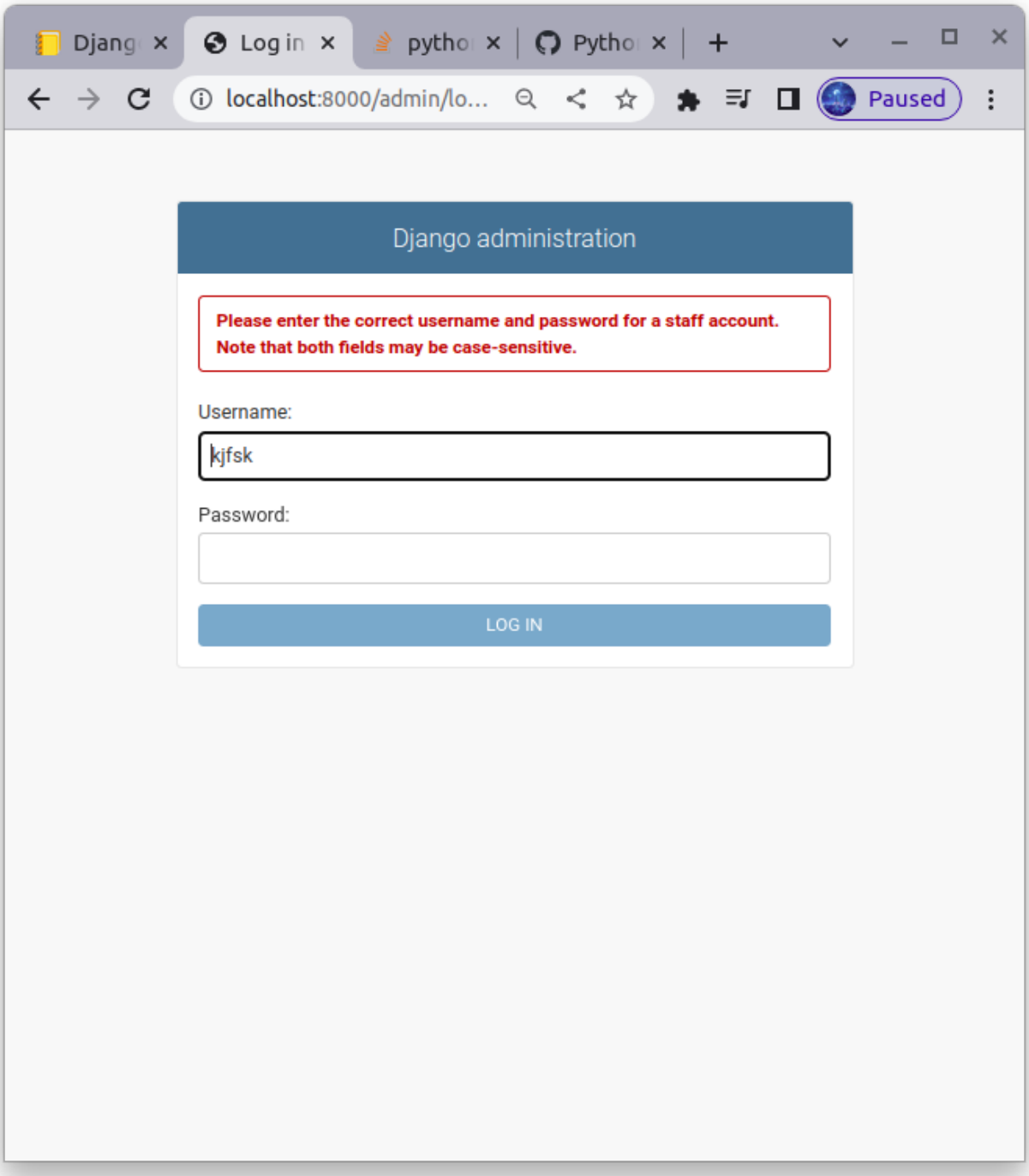


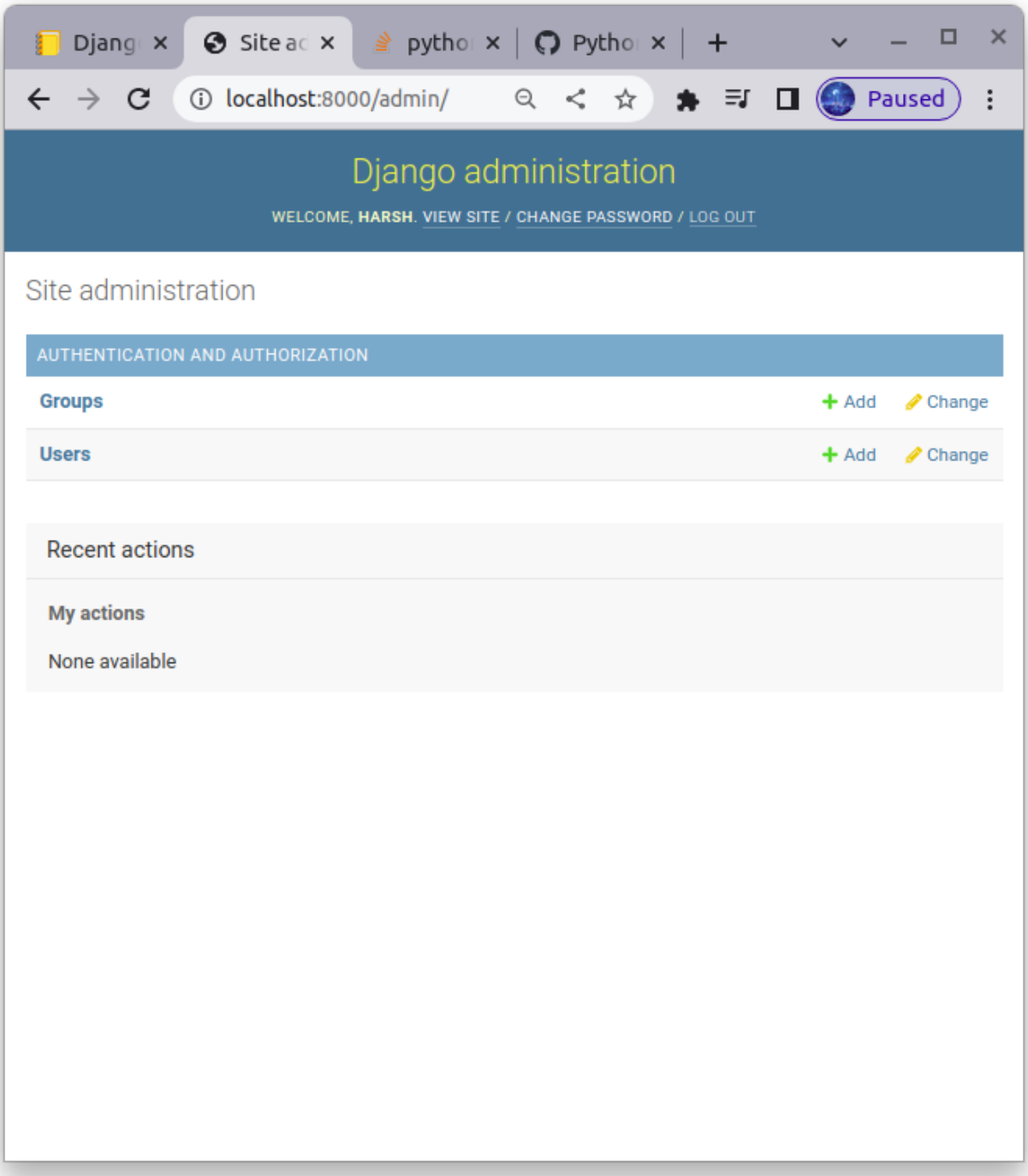
Now testing if our admin page is working properly or not


harsh@harsh-vivobook: ~/Desktop/Projects/Django projects/...
harsh@harsh-vivobook: ~/Desktop/Pr...
harsh@harsh-vivobook: ~/Desktop/Pr...
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$ python mang
e.py runserver
python: can't open file 'mange.py': [Errno 2] No such file or directory
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$ python mang
e.py run server
python: can't open file 'mange.py': [Errno 2] No such file or directory
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$ python mang
e.py run server
python: can't open file 'mange.py': [Errno 2] No such file or directory
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$ python mana
ge.py run server
Unknown command: 'run'
Type 'manage.py help' for usage.
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$ python mana
ge.py runserver
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
August 06, 2022 - 10:16:10
Django version 4.1, using settings 'p1.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

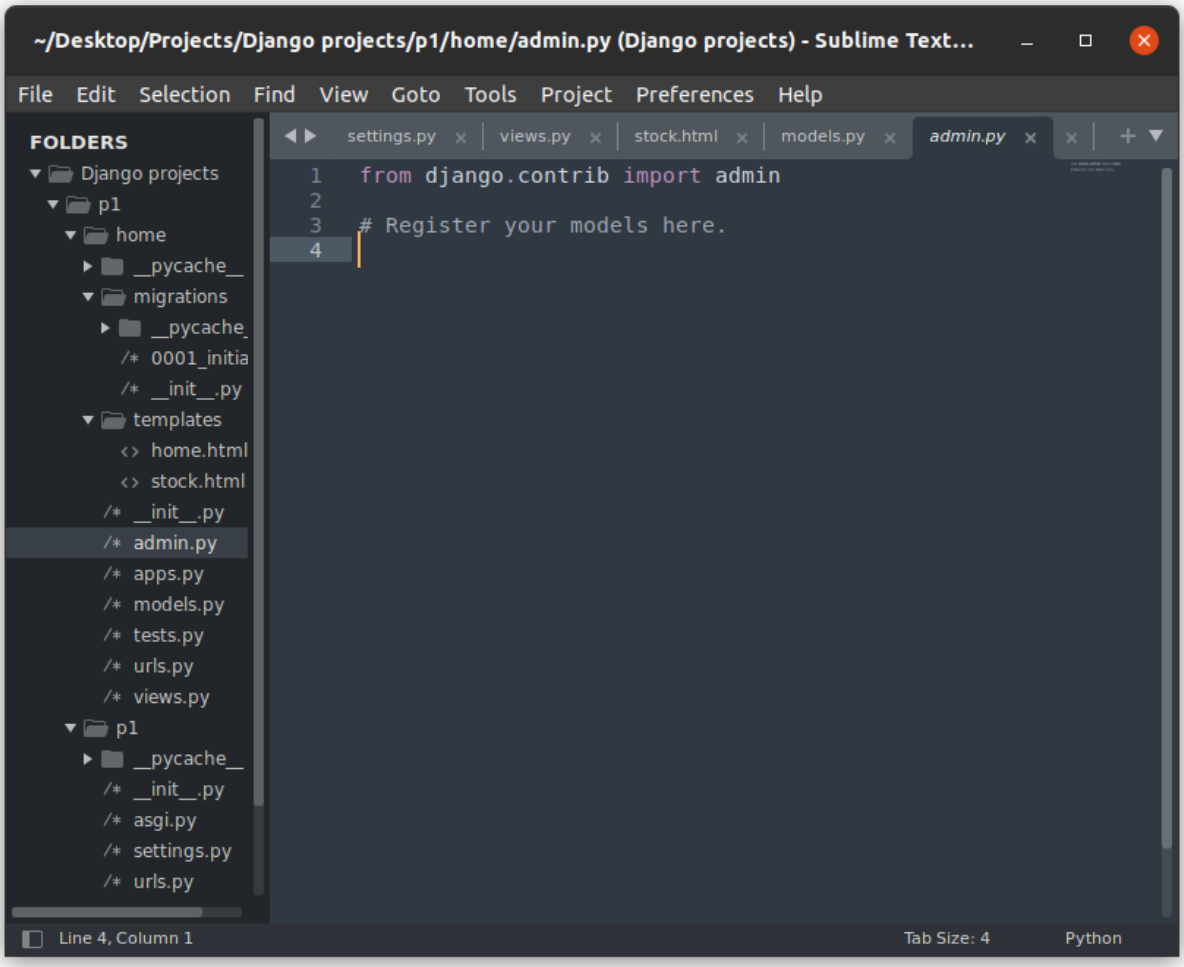

localhost:8000/stock/
Paused
Apple
Watermelon
Orange
Pear
Cherry
Strawberry
Nectarine
Grape
Mango
Blueberry
Pomegranate
Plum
Banana
Raspberry
Mandarin
Jackfruit
Papaya
Kiwi
Pineapple
Lime
Lemon
Apricot
Grapefruit
Melon
Coconut
Avocado
Peach

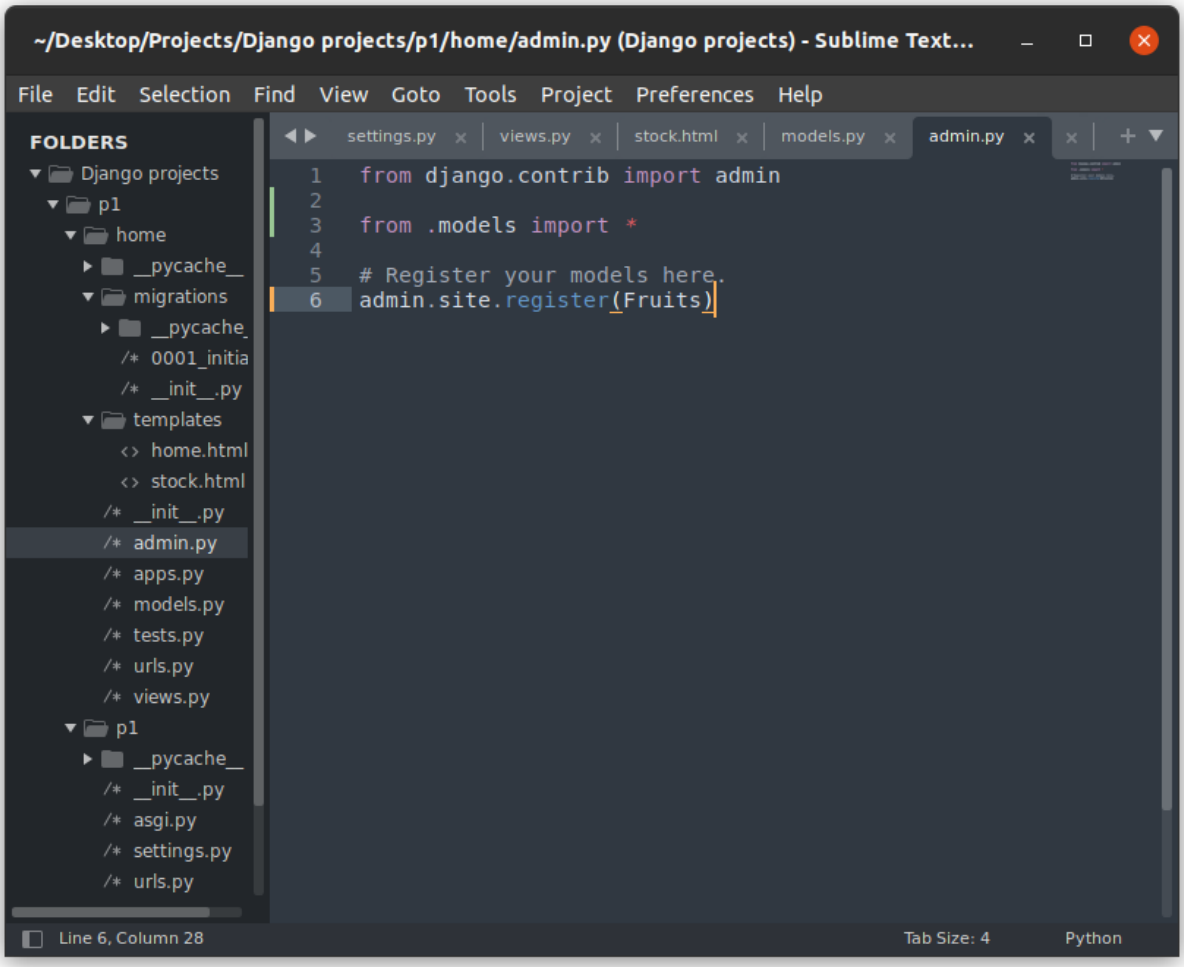

Log in
localhost:8000/admin/lo...
Paused
Django administration
Username:
Password:
LOG IN

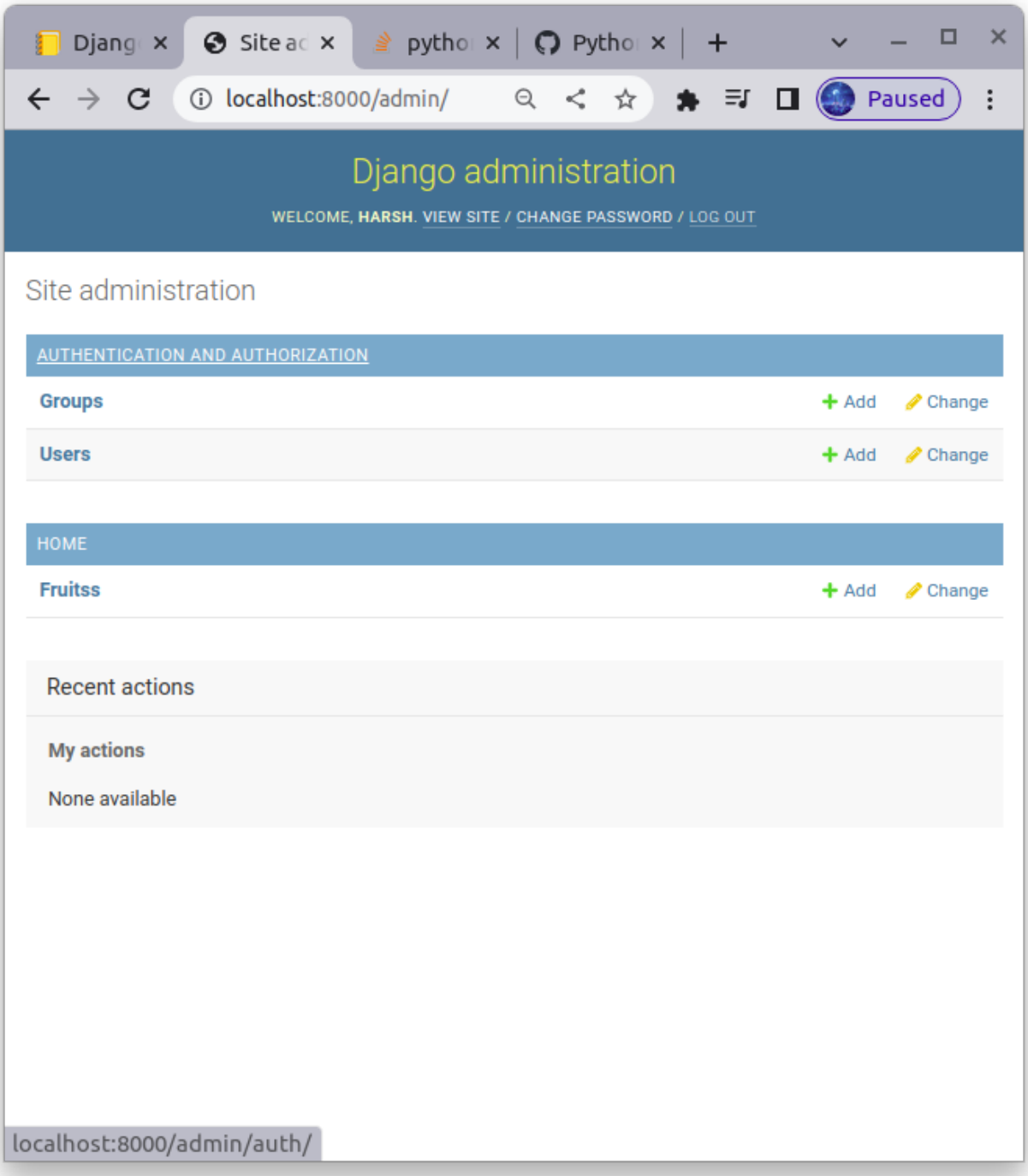

Log in
localhost:8000/admin/lo...
Paused
Django administration
Please enter the correct username and password for a staff account.
Note that both fields may be case-sensitive.
Username:
kjfsk
Password:
LOG IN



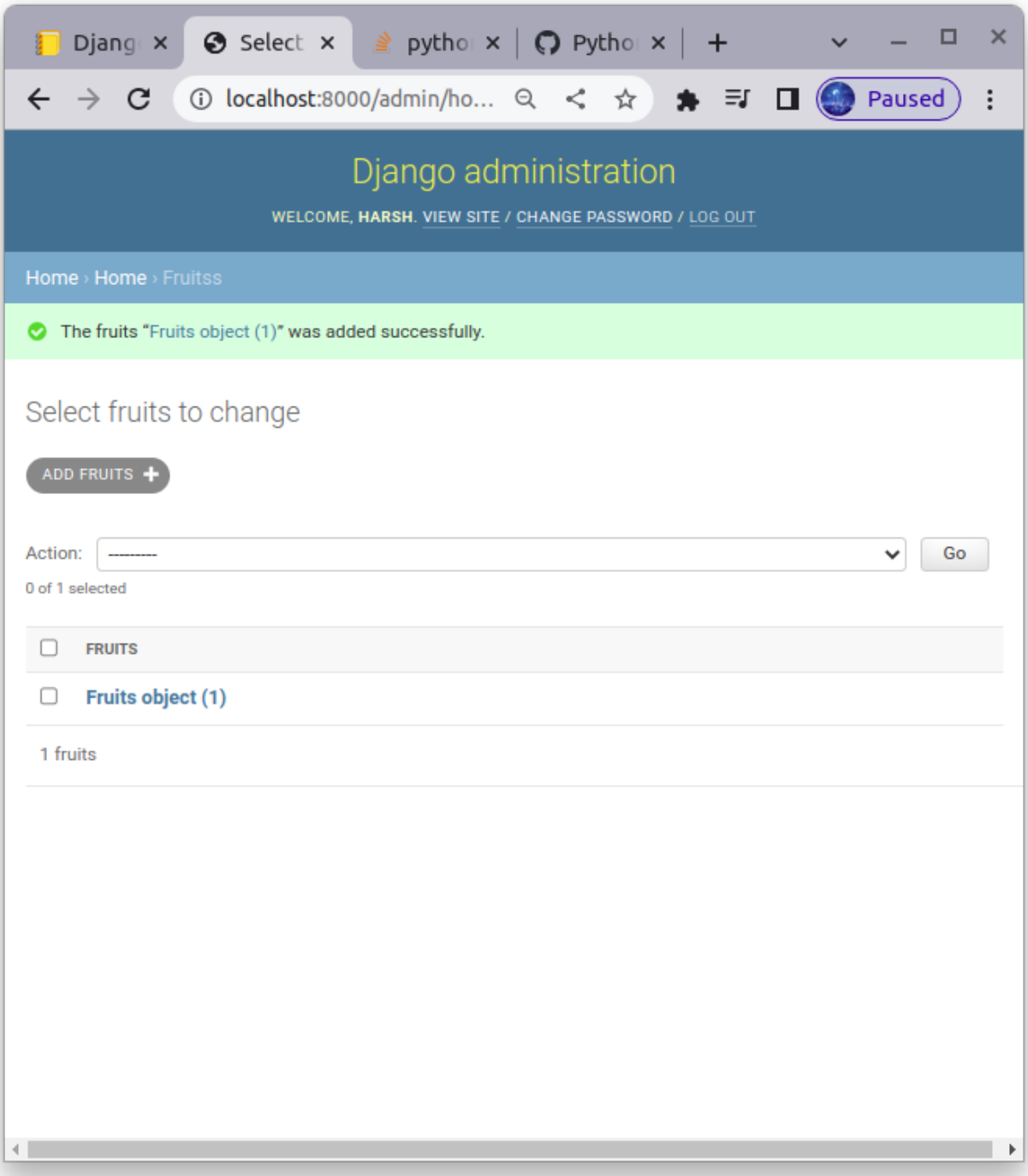
as we can see we can’t see our fruits table here because we need to register that model here first, in admin.py


~/Desktop/Projects/Django projects/p1/home/admin.py (Django projects) - Sublime Text...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__pycache_
0001_initia
__init__.py
templates
home.html
stock.html
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
settings.py
views.py
stock.html
models.py
admin.py
1 from django.contrib import admin
2
3 # Register your models here.
4
Line 4, Column 1
Tab Size: 4
Python


~/Desktop/Projects/Django projects/p1/home/admin.py (Django projects) - Sublime Text...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__pycache_
0001_initia
__init__.py
templates
home.html
stock.html
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
settings.py
views.py
stock.html
models.py
admin.py
1 from django.contrib import admin
2
3 from .models import *
4
5 # Register your models here.
6 admin.site.register(Fruits)
Line 6, Column 28
Tab Size: 4
Python

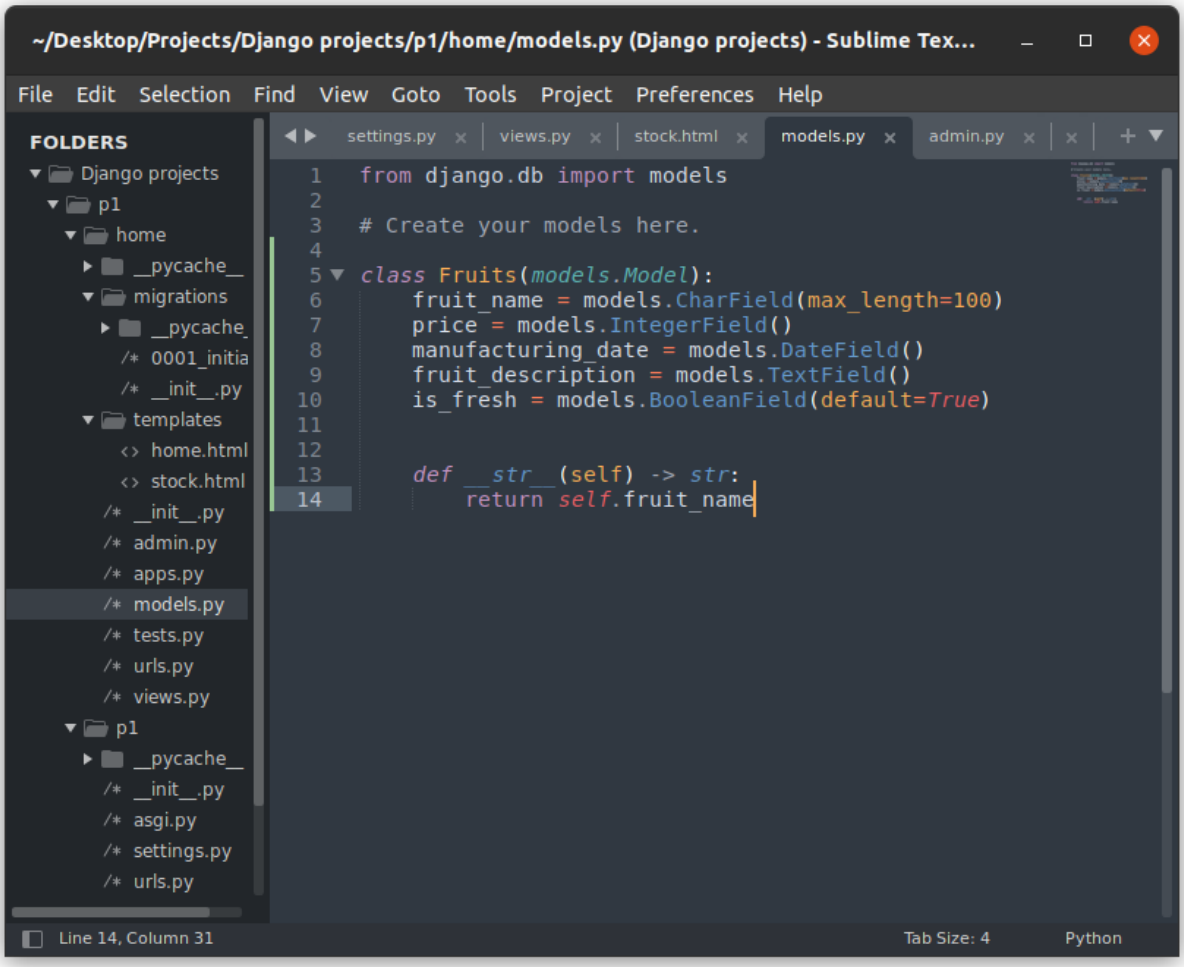


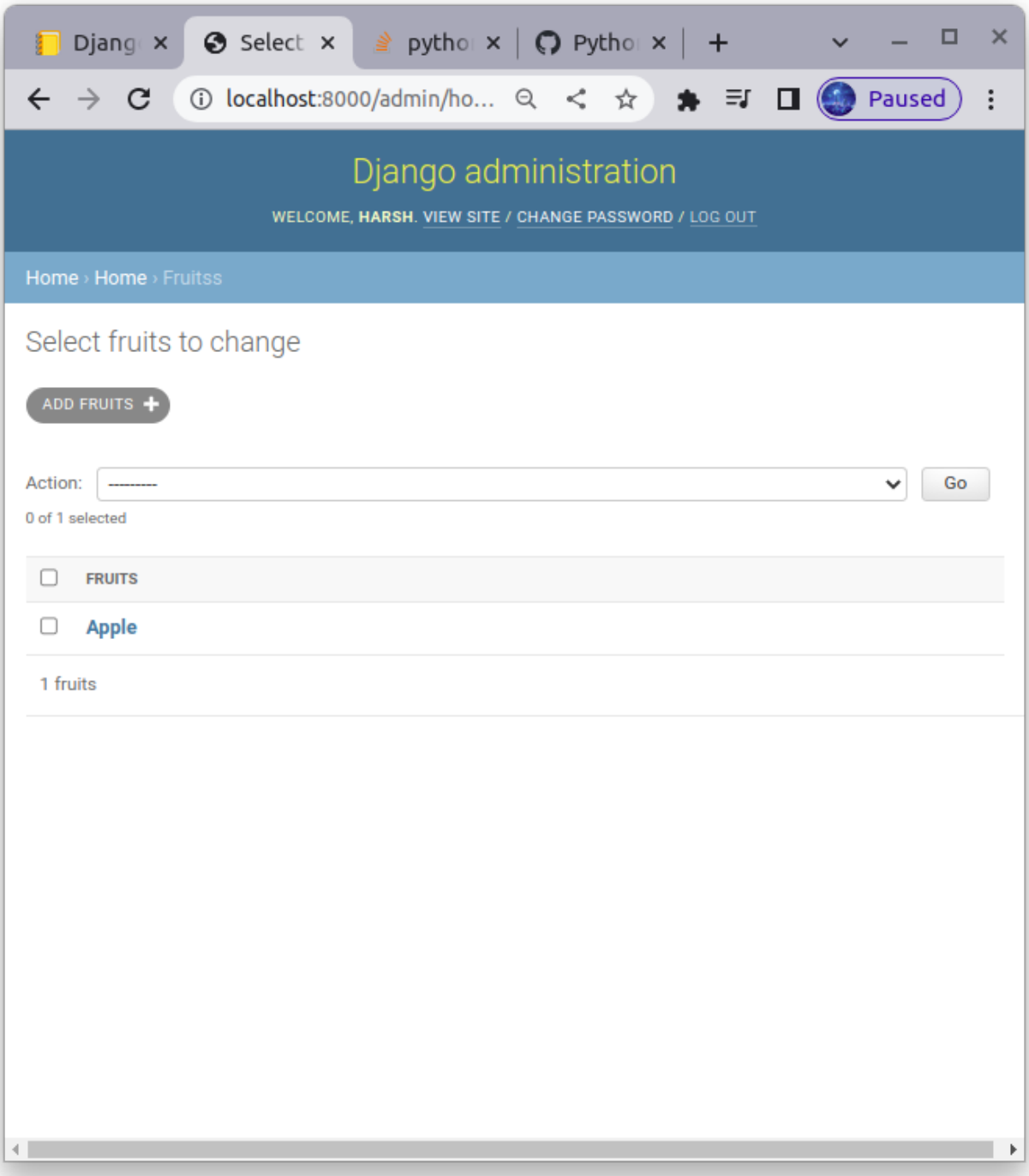
we can perform all the crud operations here



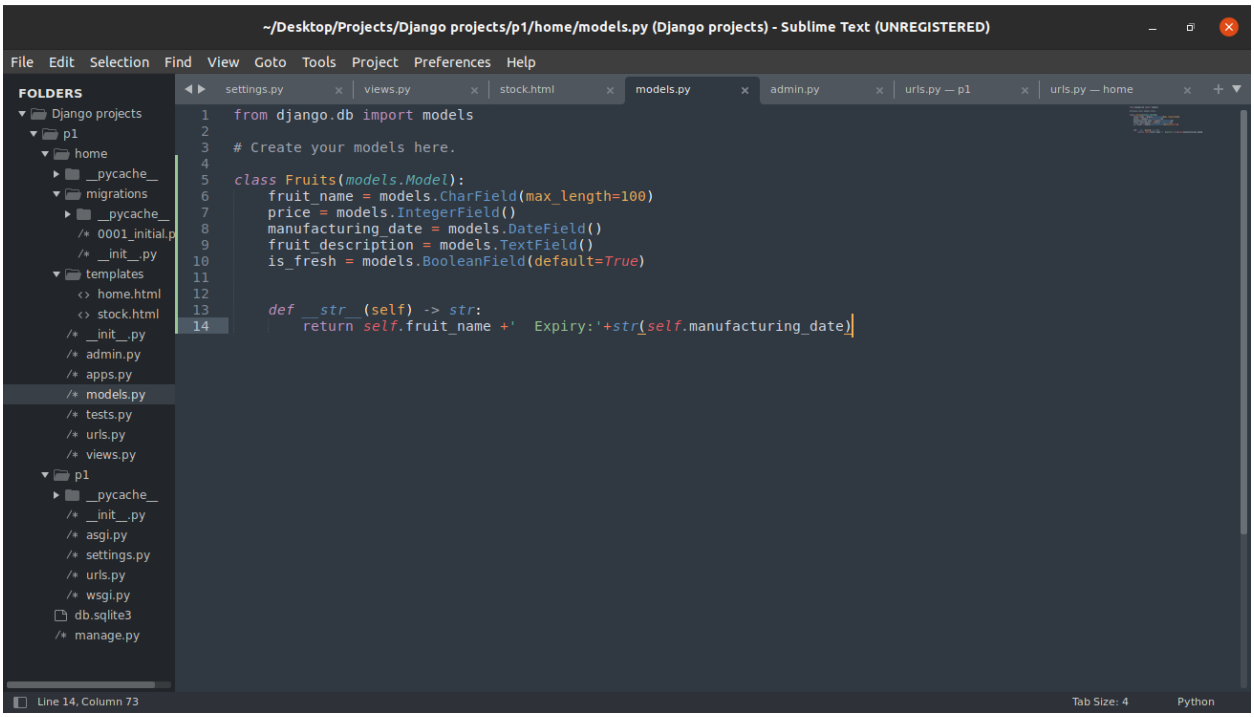
To make change that we should show fruit name instead of Fruits object in admin panel

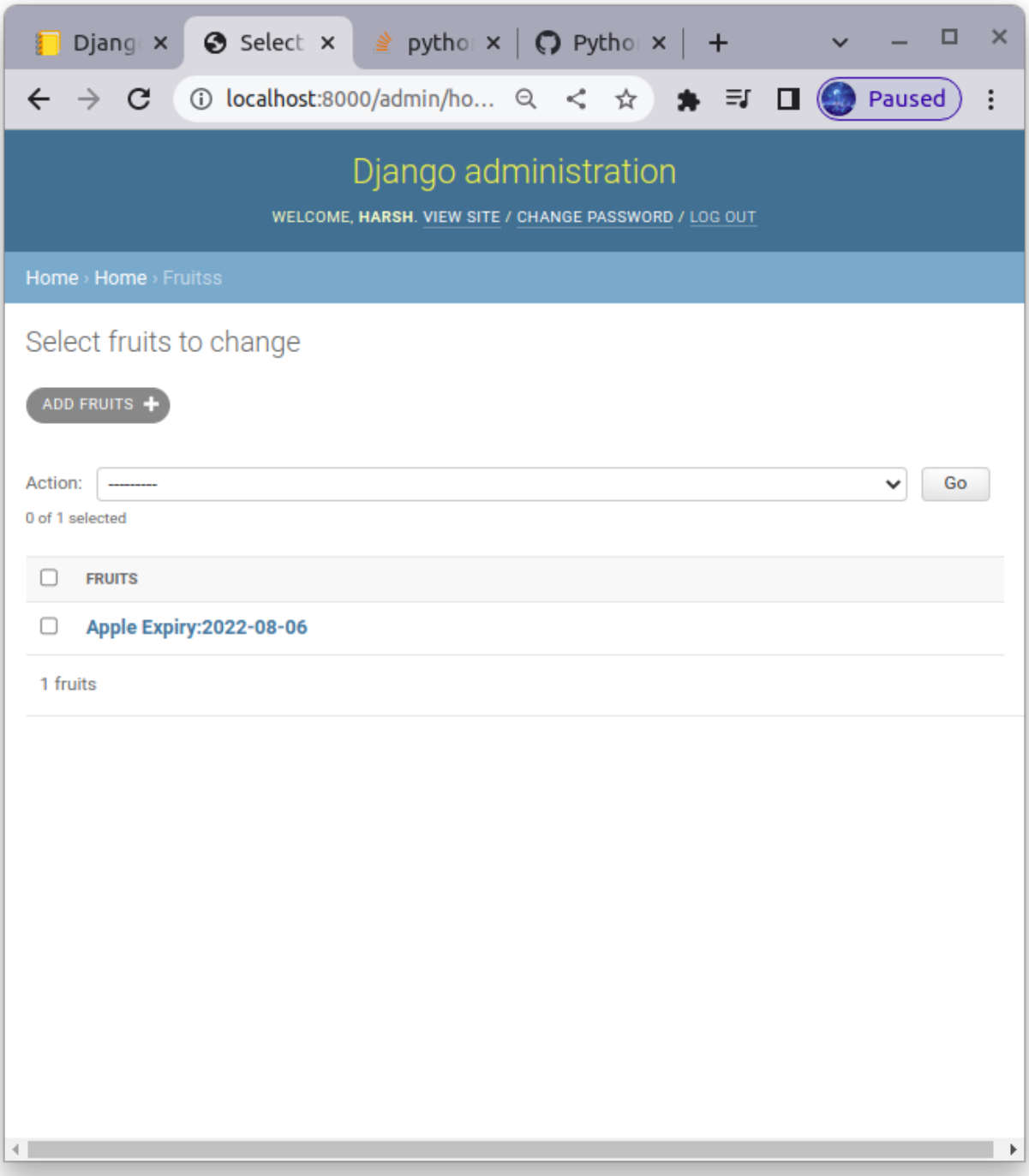
- we need to make change in the models


~/Desktop/Projects/Django projects/p1/home/models.py (Django projects) - Sublime Tex...
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__pycache_
0001_initia
__init__.py
templates
home.html
stock.html
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
settings.py
views.py
stock.html
models.py
admin.py
from django.db import models
# Create your models here.
class Fruits(models.Model):
fruit_name = models.CharField(max_length=100)
price = models.IntegerField()
manufacturing_date = models.DateField()
fruit_description = models.TextField()
is_fresh = models.BooleanField(default=True)
def __str__(self) -> str:
return self.fruit_name
Line 14, Column 31
Tab Size: 4
Python



If we want we can also add more things to it to show with the same label


~/Desktop/Projects/Django projects/p1/home/models.py (Django projects) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
Django projects
p1
home
__pycache__
migrations
__pycache__
0001_initial.p
__init__.py
templates
home.html
stock.html
__init__.py
admin.py
apps.py
models.py
tests.py
urls.py
views.py
p1
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
db.sqlite3
manage.py
settings.py
views.py
stock.html
models.py
admin.py
urls.py — p1
urls.py — home
from django.db import models
# Create your models here.
class Fruits(models.Model):
fruit_name = models.CharField(max_length=100)
price = models.IntegerField()
manufacturing_date = models.DateField()
fruit_description = models.TextField()
is_fresh = models.BooleanField(default=True)
def __str__(self) -> str:
return self.fruit_name + ' Expiry:'+str(self.manufacturing_date)
Line 14, Column 73
Tab Size: 4
Python


Djang
Select
pytho
Pytho
localhost:8000/admin/ho...
Paused
Django administration
WELCOME, HARSH. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Home › Fruitss
Select fruits to change
ADD FRUITS +
Action:

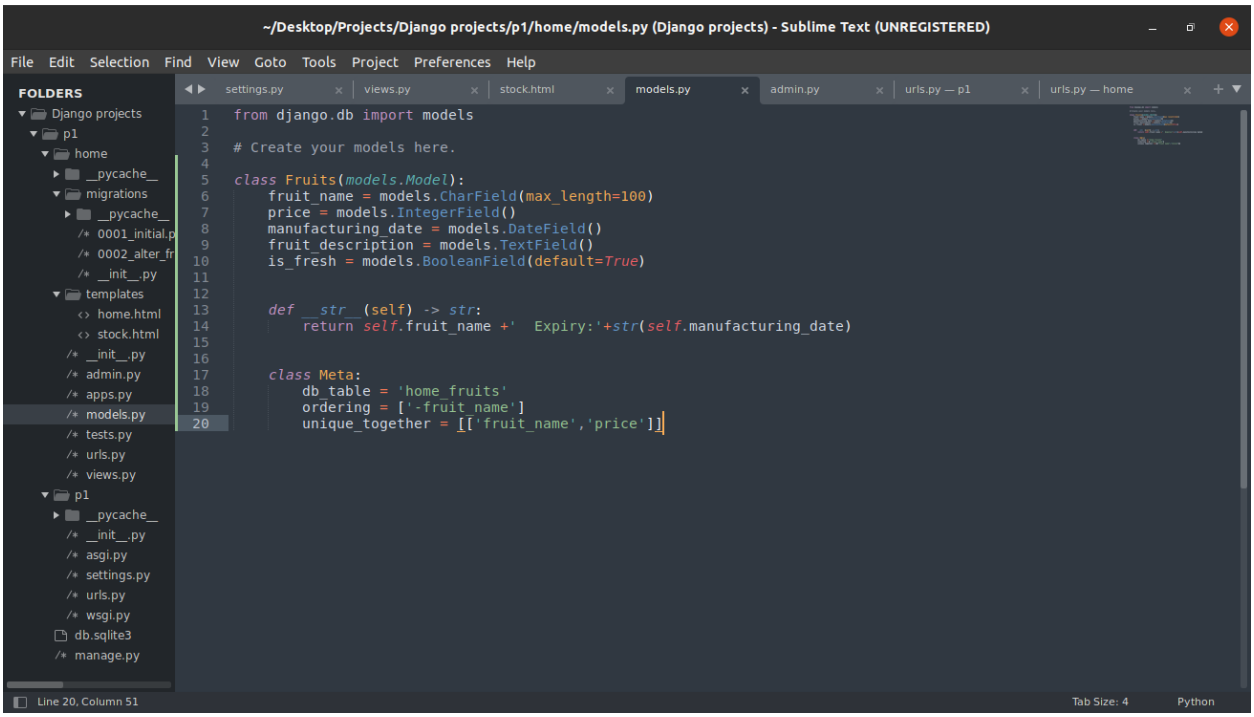
Go
0 of 1 selected
FRUITS
Apple Expiry:2022-08-06
1 fruits

Model Meta options | Django documentation | Django

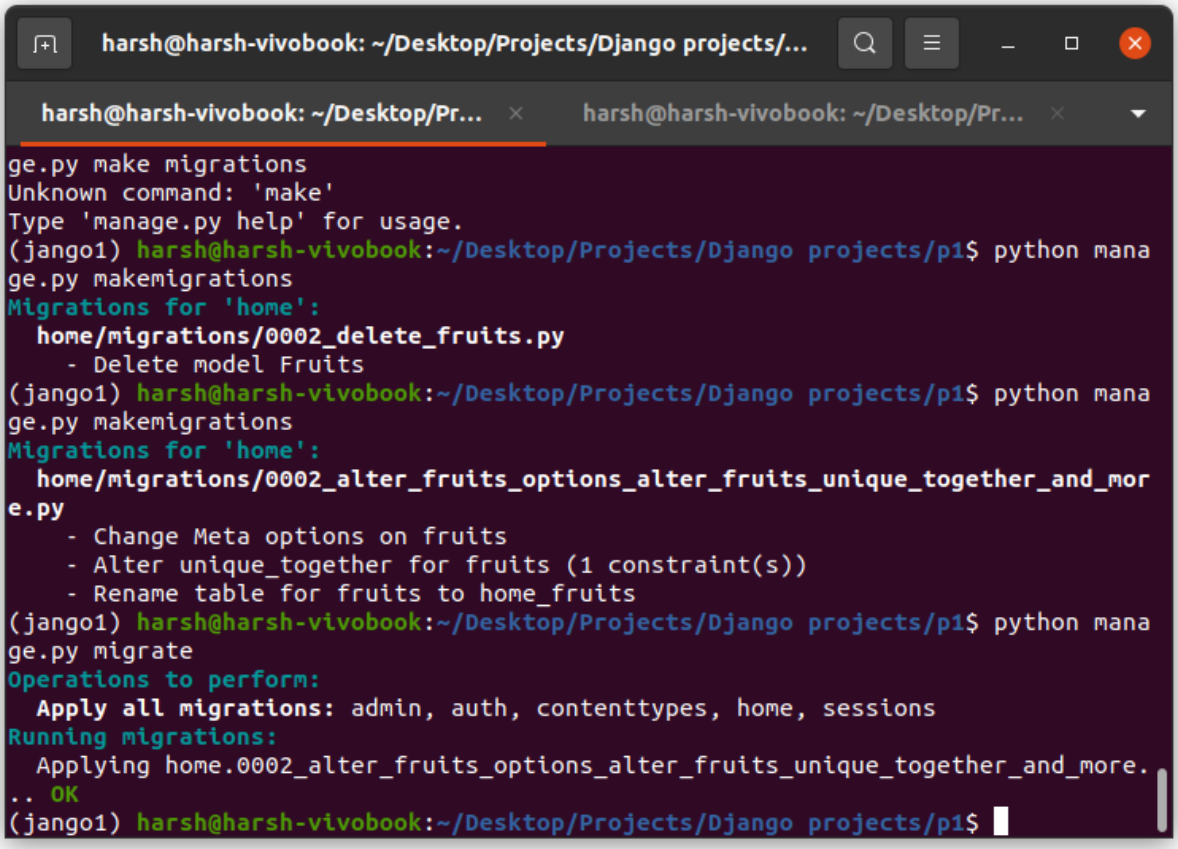
To save you time, Django automatically derives the name of the database table from the name of your model class and the app that contains it. A model's database table name is constructed by joining the model's "app label" - the name you used in - to the model's class name, with an underscore between them.

https://docs.djangoproject.com/en/4.0/ref/models/options/

using some meta class (inner class) functionalities in our code



So to make this changes in our db we need to make migrations again



If we want to keep some table default as none

we need to pass null=True and blank=True

Creating and fetching the data from the tables using ORM

- fetching the data from the db
- creating a new entry inside the table

```
python manage.py shell

>>> from home.models import *

>>> Fruits.objects.create(fruit_name='stawberry',price=344,manufacturing_date='2022-01-0
5', fruit_description='lol')
<Fruits: stawberry  Expiry:2022-01-05>

>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: peach  Expiry:2022-08-06>, <Fr
```

```
uits: papaya  Expiry:2022-08-06>, <Fruits: carrot  Expiry:2022-08-06>, <Fruits: Apple  Exp
iry:2022-08-06>]>
```

Another way of creating a new item inside the db

```
>>> obj = Fruits(fruit_name='stawberry',price=344,manufacturing_date='2022-01-05', fruit_d
escription='lol')
>>> obj = Fruits(fruit_name='pineapple',price=344,manufacturing_date='2022-01-05')
>>> obj.save()

>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Apple  Expiry:2022-08-06>]>
>>>
```

Filtering the data from the db

```
>>> Fruits.objects.filter(fruit_name = "peach")
<QuerySet [<Fruits: peach  Expiry:2022-08-06>]>

>>> objs = Fruits.objects.filter(fruit_name='peach')
>>> objs
<QuerySet [<Fruits: peach  Expiry:2022-08-06>]>
>>> objs[0]
<Fruits: peach  Expiry:2022-08-06>
```

```
(jango1) harsh@harsh-vivobook:~/Desktop/Projects/Django projects/p1$     python manage.py
 shell

>>> from home.models import *
>>> obj = Fruits.objects.all()
>>> obj
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Apple  Expiry:2022-08-06>]>
```

```
>>> obj[0]


>>> obj = Fruits(fruit_name = 'Salak', price=100, manufacturing_date='2022-03-04',fruit_de
scription='rare fruit')
>>> obj.save()
>>> getallobj = Fruits.objects.all()
>>> getallobj
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak  Expiry:2022-03-04>, <Fruits: Apple  Expiry:2022-08-06
>]>


>>> Fruits.objects.filter(fruit_name='Salak')
<QuerySet [<Fruits: Salak  Expiry:2022-03-04>]>
>>> f = Fruits.objects.filter(fruit_name='Salak')
>>> f[0]
<Fruits: Salak  Expiry:2022-03-04>
>>> f[0].fruit_description
'rare fruit'

>>> f[0].id
7


>>> f = Fruits.objects.get(id =7)
>>> f.fruit_description
'rare fruit'
>>> f.fruit_description = 'Indonesian rare fruit'
>>> f.save()

>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak  Expiry:2022-03-04>, <Fruits: Apple  Expiry:2022-08-06
>]>
>>>
>>>

>>> fn = Fruits.objects.filter(fruit_name='Salak')
>>> fn
<QuerySet [<Fruits: Salak  Expiry:2022-03-04>]>
>>> fn[0].fruit_description
'Indonesian rare fruit'
```

updating the new changes in the list of all fruits

```
>>> from home.models import *
>>> obj = Fruits.objects.all()
>>> obj
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak  Expiry:2022-03-04>, <Fruits: Apple  Expiry:2022-08-06
>]>
>>>
>>>
>>> getid = Fruits.objects.filter(fruit_name='Salak')
>>> getid[0]
<Fruits: Salak  Expiry:2022-03-04>
>>> getid[0].id
7
>>>

>>> f = Fruits.objects.get(id=7)
>>> f.fruit_name = 'Salak Indo'
>>> f.fruit_description = 'name updated'
>>> f.save()

>>> f
<Fruits: Salak Indo  Expiry:2022-03-04>
>>> f.fruit_name
'Salak Indo'
>>> f.fruit_description
'name updated'
>>>
```

```
>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>, <Fruits: Apple  Expiry:2022-
08-06>]>

>>> Fruits.objects.all().order_by('id')
<QuerySet [<Fruits: Apple  Expiry:2022-08-06>, <Fruits: peach  Expiry:2022-08-06>, <Fruit
s: carrot  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: stawberry  Ex
piry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>, <Fruits: Salak Indo  Expiry:2022
-03-04>]>
>>>

>>> Fruits.objects.get(id=1)
<Fruits: Apple  Expiry:2022-08-06>
```

Deleting a fruit from the database after fetching the fruit id

```
>>> Fruits.objects.all().order_by('id')
<QuerySet [<Fruits: Apple  Expiry:2022-08-06>, <Fruits: peach  Expiry:2022-08-06>, <Fruit
s: carrot  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: stawberry  Ex
piry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>, <Fruits: Salak Indo  Expiry:2022
-03-04>]>
>>>

>>> Fruits.objects.get(id=1)
<Fruits: Apple  Expiry:2022-08-06>
>>> appleobj = Fruits.objects.get(id=1)
>>> appleobj.delete()
(1, {'home.Fruits': 1})
>>>

>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>]>
```

Filtering the price of the fruit, greater than or less than

```
>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>]>

>>> Fruits.objects.filter(price=100)
<QuerySet [<Fruits: Salak Indo  Expiry:2022-03-04>]>

>>> Fruits.objects.filter(price__lte = 100)
<QuerySet [<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruit
s: carrot  Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>]>

>>> Fruits.objects.filter(price__gte = 100)
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: Salak Indo  Expiry:2022-03-04>]>
```

Filtering fruits having a char ‘a’ in their name

```
>>> from home.models import *
>>> Fruits.objects.all()
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>]>

# list of fruits name that consist of char 'a'
>>> Fruits.objects.filter(fruit_name__icontains='a')
<QuerySet [<Fruits: stawberry  Expiry:2022-01-05>, <Fruits: pineapple  Expiry:2022-01-05>,
<Fruits: peach  Expiry:2022-08-06>, <Fruits: papaya  Expiry:2022-08-06>, <Fruits: carrot
 Expiry:2022-08-06>, <Fruits: Salak Indo  Expiry:2022-03-04>]>
```