



Institute of Technology, Nirma University
Programming For Scientific Computing(2CS404)

Year 2023-24

Semester : 4

E-commerce Web-app using Django API

Submitted by

Name	Roll No
Smit Patel	21bce217
Urvish Patel	21bce221
Kevin Prajapati	21bce230

● Procedure to Start Project:

- Once you have downloaded pyshop.zip and extracted all of its contents.
- Next step is to open the extracted folder with IDE(we have used Visual studio code).
- Next install django using command `pip install django`
- Now check you are inside your extracted folder
- Next we have to run the command `python manage.py runserver` to launch our django server.

```
PS G:\Academic sem 4\PSC\Python_Final> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 28, 2023 - 19:43:09
Django version 4.2, using settings 'pyshop.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

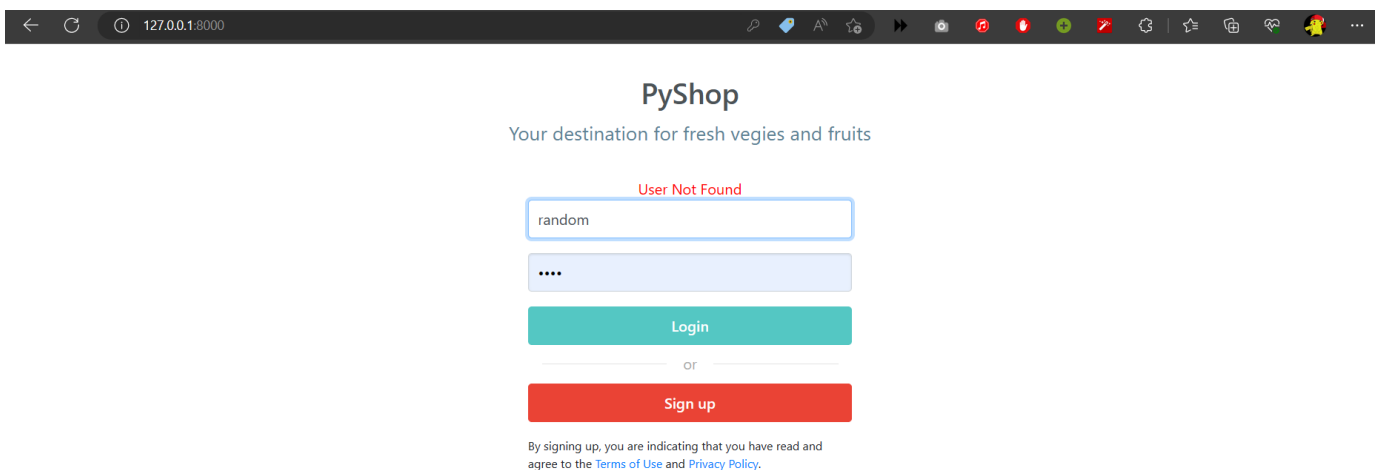
- Now click on the IDE generated server link(`http://127.0.0.1:8000/`) to go to the home page of our project.

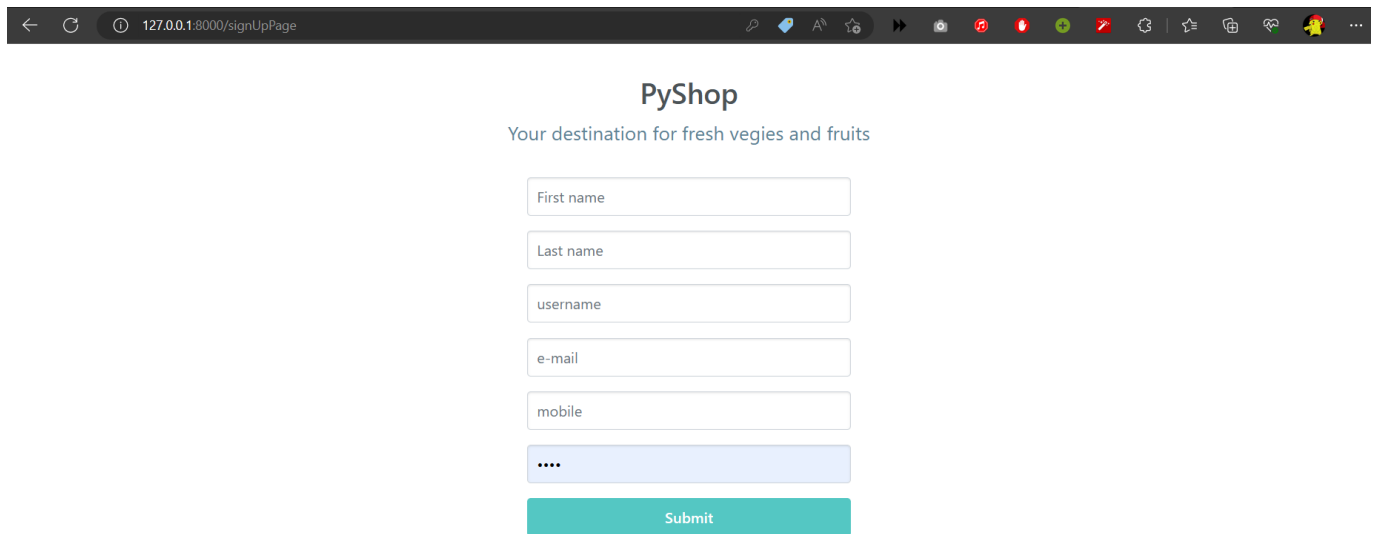
● Features:

We have 2 types of profiles: user and admin.

Let's first explore the user side of the project.

- The webpage you travelled on clicking on the link from IDE will take you to the User login/signup page where you can create new users.
- Login/Signup page → user can login into existing accounts or sign up for a new account. If you try to login with the wrong user it will display "User Not Found". All the users are saved in the backend.





The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/signUpPage'. The page has a dark header with navigation icons. The main content area is white and features the 'PyShop' logo at the top, followed by the tagline 'Your destination for fresh vegies and fruits'. Below this is a registration form with six input fields: 'First name', 'Last name', 'username', 'e-mail', 'mobile', and a password field represented by four dots. A teal 'Submit' button is positioned at the bottom of the form.

PyShop

Your destination for fresh vegies and fruits

First name

Last name

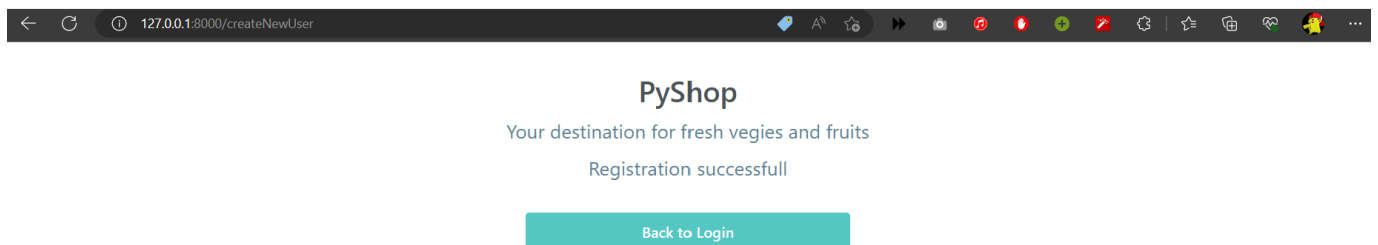
username

e-mail

mobile

....

Submit



The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/createNewUser'. The page has a dark header with navigation icons. The main content area is white and features the 'PyShop' logo at the top, followed by the tagline 'Your destination for fresh vegies and fruits'. Below this, the text 'Registration successfull' is displayed. A teal 'Back to Login' button is positioned at the bottom of the page.

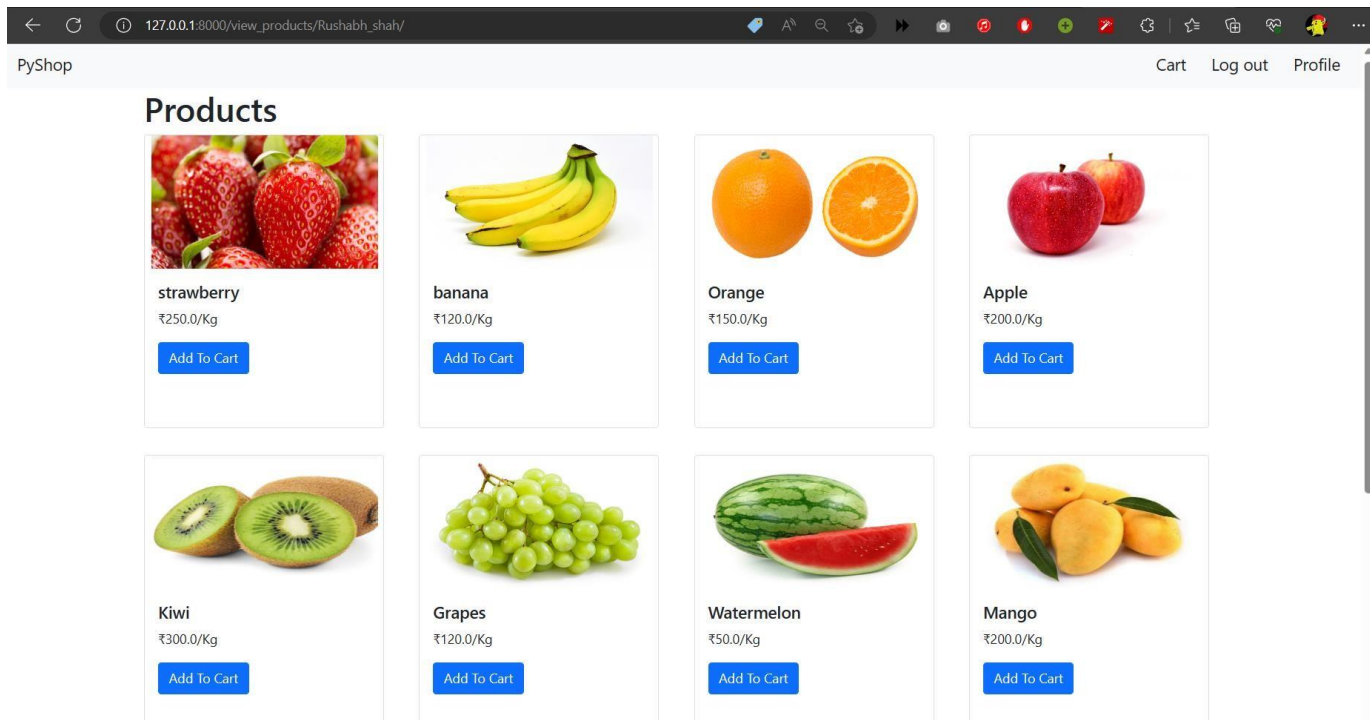
PyShop

Your destination for fresh vegies and fruits

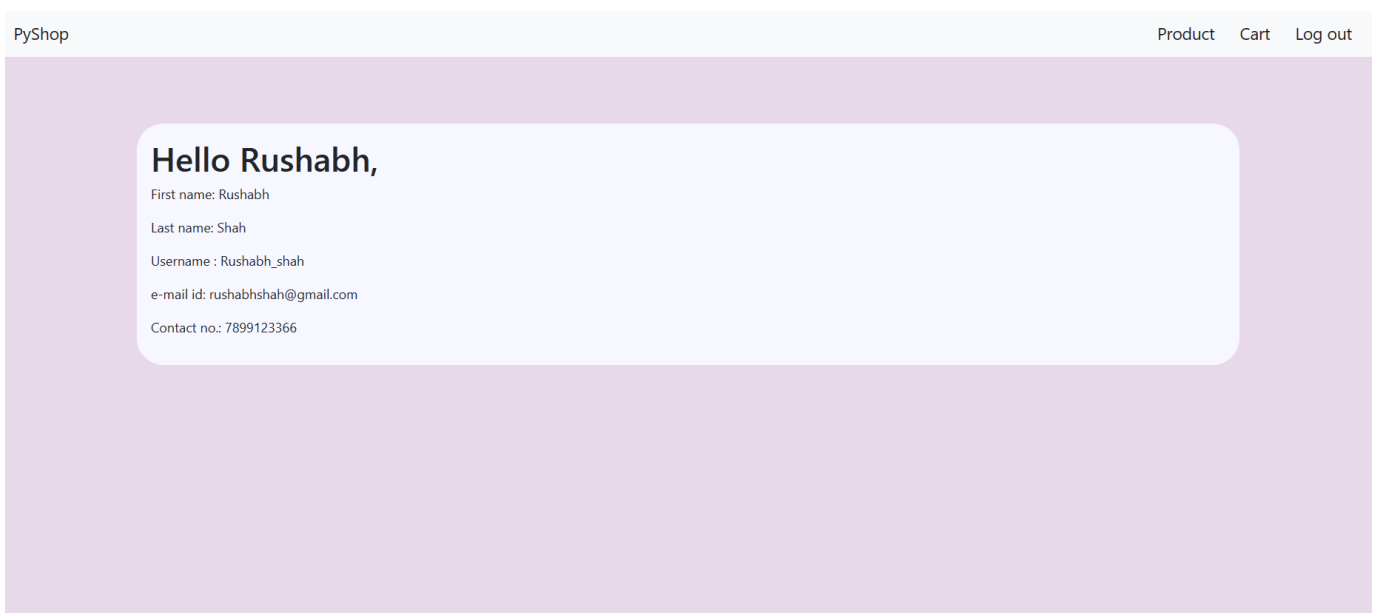
Registration successfull

Back to Login

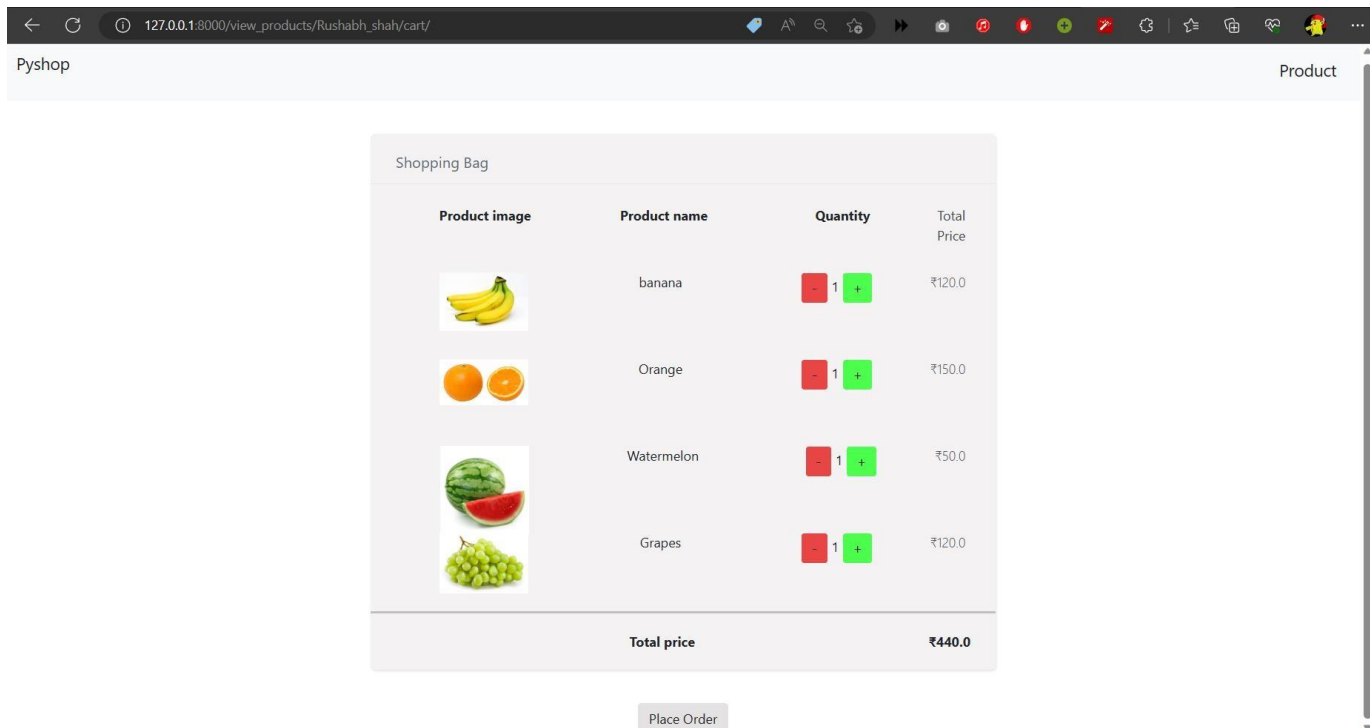
- As you move to any other page, you will see the difference in url also. You will see your login username in the url as well and if you tried to change the username you will be logged out.
- Products(home) page → this is the first page you will see after logging in your account. It displays all the products offered by the website for sale. Users can select multiple products from the home page which will automatically be added to the user's cart. This page allows the user to navigate to 'cart', 'profile' and 'logout button'.



- Profile page → it will display the user's profile details. Users can navigate to the 'Product' page, 'Cart' page or log out from the account.



- Cart page → this page displays details of our ordered(selected) products. Users can add more products or remove some products. Once a user has decided on the order he/she can place it by clicking on the “Place Order” button. This site also lets users navigate back to the Products page by clicking Product button which is at right corner.



- Checkout window → on clicking the place order button of the cart user is sent to the checkout window. This window asks details required for delivering the products. All the fields are mandatory to be filled. Once the user has filled in all the details he/she can place the order by clicking on Submit button.

Pyshop Back to Cart

Enter Address for This Order

Address

Deliver to*
Rushabh Shah

Apartment, unit, suite, or floor #
48 shivanandan residency

City*
Ahmedabad

State*
Gujarat

Postal code*
382330

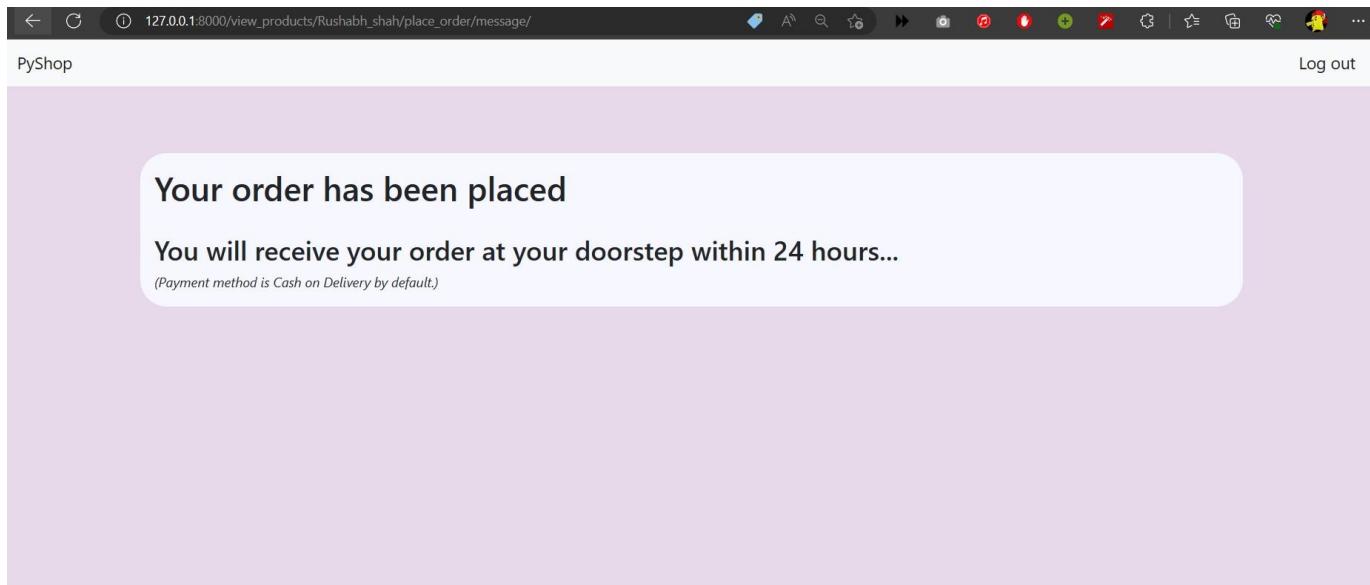
Country*
India

Submit

Place Order

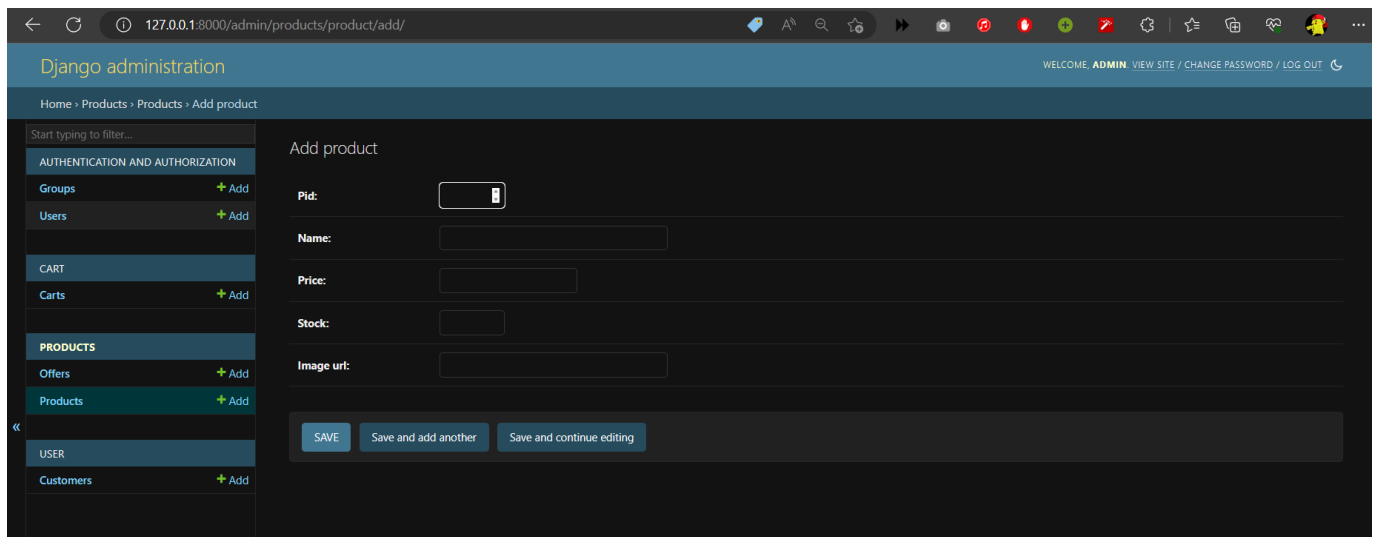
* = required field

- Message window → after placing an order website takes you to a webpage that contains an auto generated message that confirms your order. You can logout of your account by clicking on the logout button.



Next let's see the admin side.

- Admin side facilities are provided by django and for accessing it you need to copy the server link provided by IDE on a new tab and concatenate it with "/admin". This will take you to django's default functionality for admins.
- Now you can enter with username:admin and password:1234.
- You can also create your own superuser(from terminal using) but for simplicity use the account given above.
- We have created 4 Entities: Carts, Products, Offers and Customers.
- Admin account can be used to modify and authorise all details in the database.
- Admin can add, update or delete products from the website.



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

CART

Carts + Add

PRODUCTS

Offers + Add

Products + Add

USER

Customers + Add

Select product to change

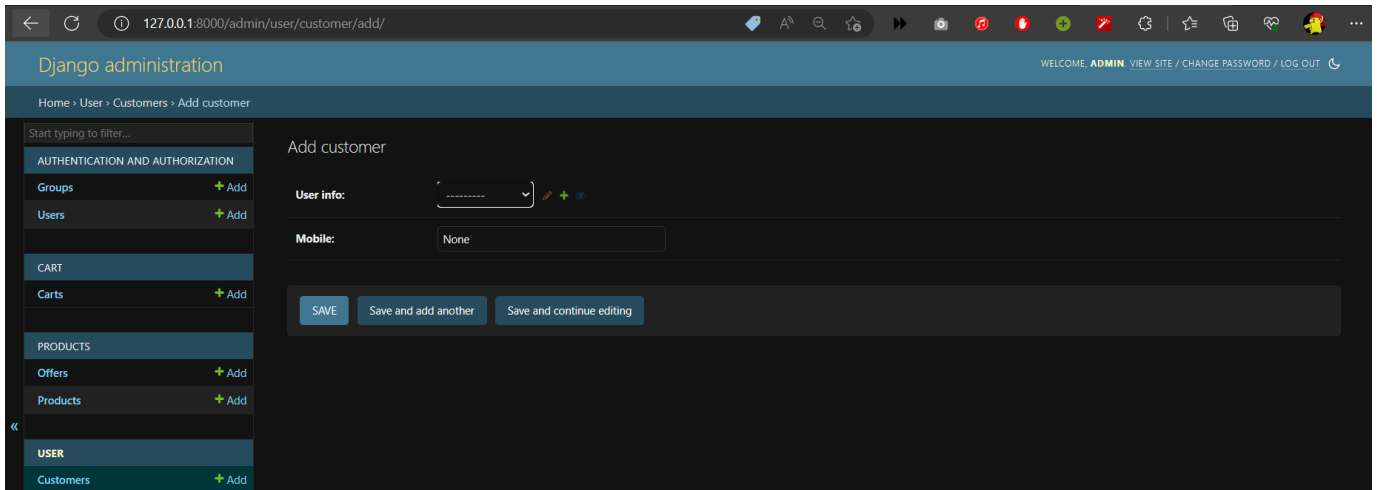
ADD PRODUCT +

Action: ----- Go 0 of 9 selected

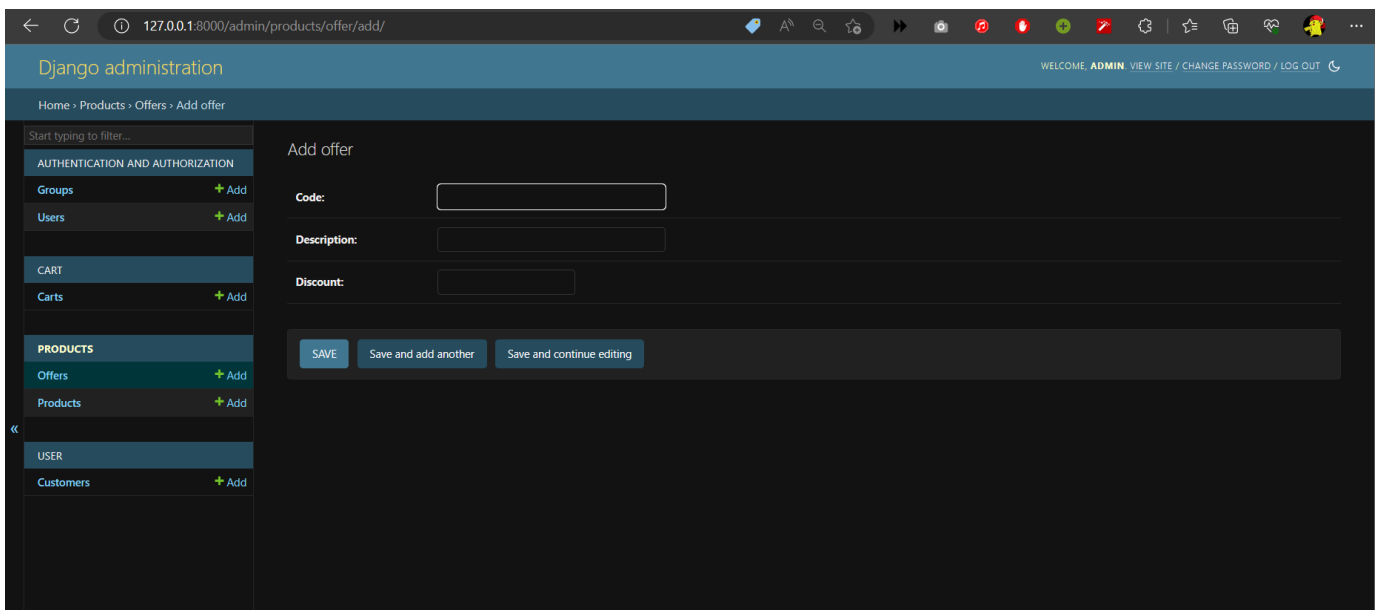
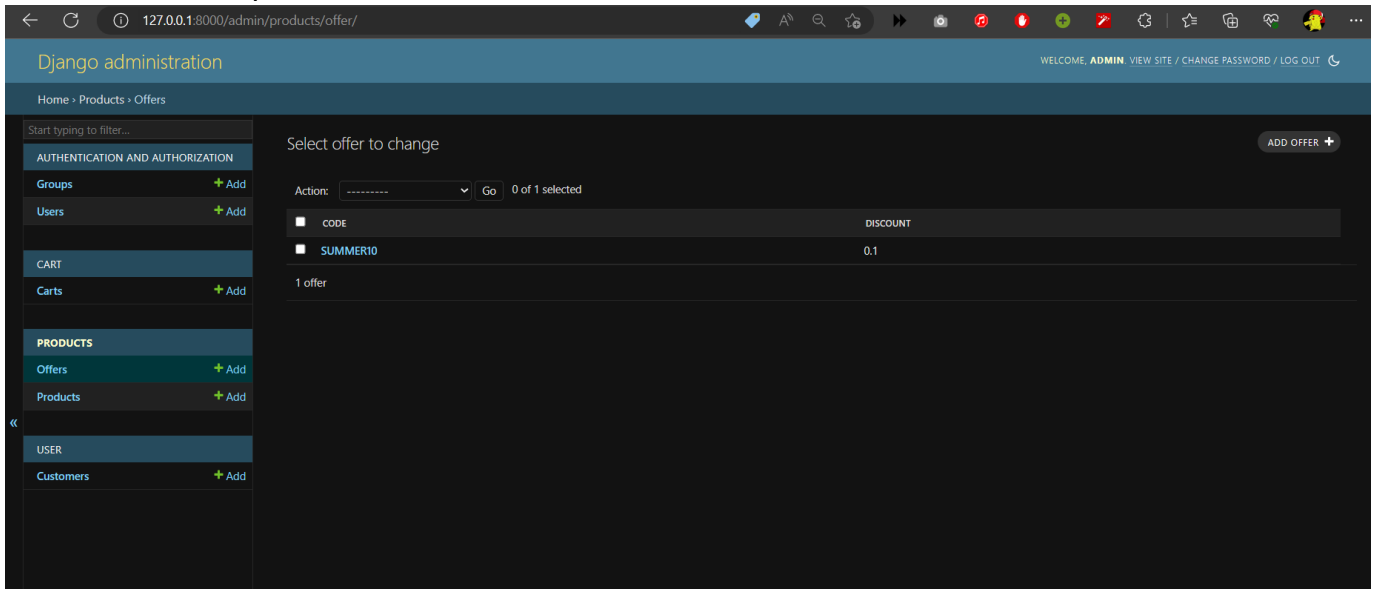
NAME	PID	PRICE	STOCK
Custard apple	9	180.0	40
Mango	8	200.0	500
Watermelon	7	50.0	60
Grapes	6	120.0	200
Kiwi	5	300.0	30
Apple	4	200.0	100
Orange	3	150.0	40
banana	2	120.0	20
strawberry	1	250.0	30

9 products

- Admin can add, update or delete customers from the website.



- Admin can update offers from the website.



- Admin can add cart option(eg..any fruits) for to the home page.

127.0.0.1:8000/admin/cart/cart/add/

Django administration

Home > Cart > Carts > Add cart

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CART

Carts + Add

PRODUCTS

Offers + Add

Products + Add

USER

Customers + Add

Add cart

Username:

Productkey:

Productname:

Image url:

Quantity:

Total price:

SAVE Save and add another Save and continue editing

➤ Admin can add, update or delete user from the website.

127.0.0.1:8000/admin/auth/user/add/

Django administration

Home > Authentication and Authorization > Users > Add user

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CART

Carts + Add

PRODUCTS

Offers + Add

Products + Add

USER

Customers + Add

Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username:

Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

SAVE Save and add another Save and continue editing

127.0.0.1:8000/admin/auth/user/

Django administration

Home > Authentication and Authorization > Users

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CART

Carts + Add

PRODUCTS

Offers + Add

Products + Add

USER

Customers + Add

Select user to change

Search

Action: Go 0 of 5 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Rushabh_shah	rushabhshah@gmail.com	Rushabh	Shah	<input type="radio"/>
<input type="checkbox"/>	Urvish_1609	urvishpatel1609@gmail.com	Urvish	Patel	<input type="radio"/>
<input type="checkbox"/>	admin	admin_account@pyshop.com	Admin		<input checked="" type="radio"/>
<input type="checkbox"/>	kevin_47	kevinprajapati@gmail.com	Kevin	Prajapati	<input type="radio"/>
<input type="checkbox"/>	smit_patel	smitpatel@gmail.com	Smit	Patel	<input checked="" type="radio"/>

5 users

ADD USER +

FILTER

By staff status

All

Yes

No

By superuser status

All

Yes

No

By active

All

Yes

No

By groups

All

Customer

-

✧ Urls:

```
from django.contrib import admin
from django.urls import path, include
from user import views as uv
from products import views as pv
from cart import views as cv

urlpatterns = [
    path('', uv.log_in_view),
    path('admin/', admin.site.urls),
    path('view_products/<str:username>/', uv.showProduct, name='products'),
    path('view_products/<str:username>/cart/', cv.cart, name='cart'),
    path('createNewUser', uv.createNewUser),
    path('signUpPage', uv.signUpPage),
    path('view_products/<str:username>/place_order', cv.placeOrder),
    path('logout/', uv.user_logout, name='logout'),
    path('profile/', uv.user_profile, name='profile'),
    path('view_products/<str:username>/place_order/', cv.placeOrder),
    path('view_products/<str:username>/place_order/message/', cv.message),
    path('view_products/<str:username>/cart/', cv.cart, name='cart'),
]
```

✧ Our Business logic .py files

- user/view.py
- cart/view.py
- pyshop/view.py

And rest of the code in Zip file.