

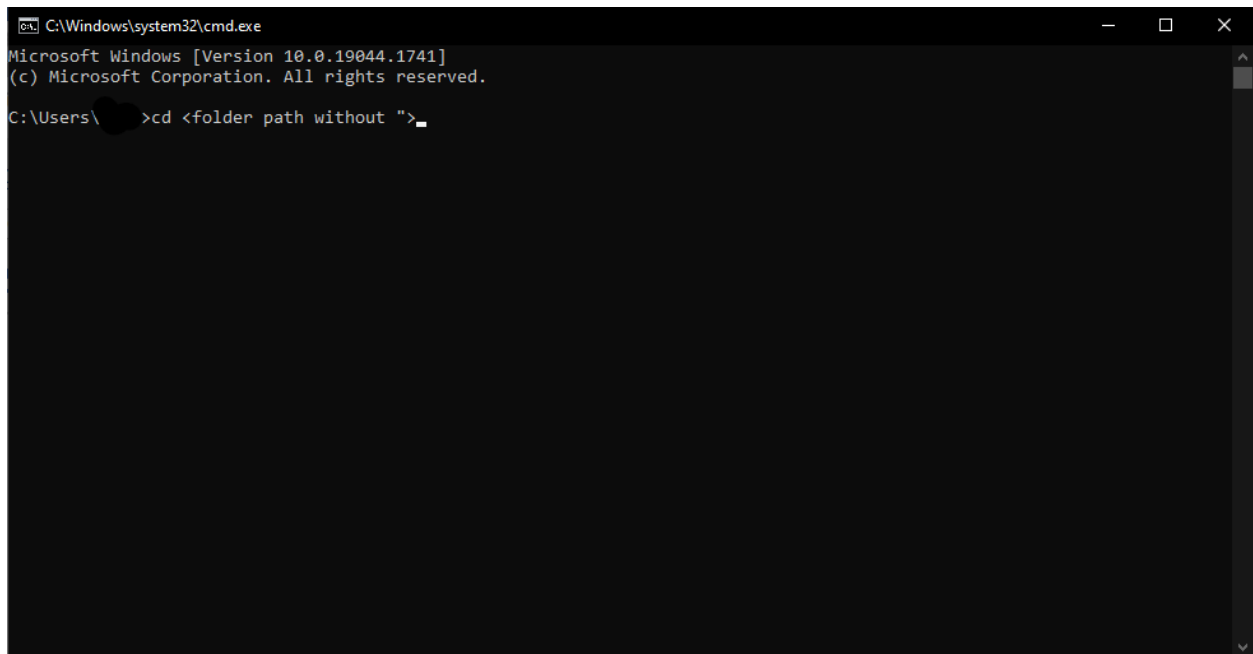
# Django E-Commerce Website Project

Procedural Instructions [\*\(GitHub Reference at the last of the document\)\*](#)

## Step 1 | Installation

### *Creating a virtual environment*

Creating a virtual environment is recommended as it makes it easy to install multiple python plugins, if necessary, without changing the global settings. In short, it makes a virtual environment for all your needs. First, create a new folder for your project. In this case I'll create a folder named **shop**. Then open a new cmd window inside the new folder you've created by hitting **Win + R** and typing **cmd** and hit **Ok**.



Then type the command above and replacing **<folder path without ">** with your folder path. Then typing the following command and replacing **<name>** with your project name.

```
C:\Users\ >py -m venv \<name>_
```

This will create a folder named **venv** in your project folder.

As of updated python versions, there's no need to activate the virtual environment but still we can activate it by running the **activate.bat** script inside the venv folder.

Now you've successfully activated the virtual environment and your cmd should have **(venv) <project path>** as its first line.

### *Installing Django*

Type **py -m pip install Django** inside the cmd window. This will install Django inside your virtual environment (*if you have one*). Now run **django-admin startproject <project name>**. This will create a project folder named with your project name (*here its shop*). Now your directory should look something like this. "." Is root directory.

```
.
├── <project_name>
│   ├── <project>
│   │   ├── __init__.py
│   │   ├── asgi.py
│   │   ├── settings.py
│   │   ├── urls.py
│   │   └── wsgi.py
│   └── manage.py
```

Now you've setup all the files needed to make this project a successful one! Let's go!

### *Adding the app app*

Now open a cmd window in <project\_name> folder and type **py manage.py startapp app**. This will create a new folder and your directory should look something like this.

```
.
├── <project_name>
│   ├── app
│   │   ├── migrations
│   │   ├── __init__.py
│   │   ├── admin.py
│   │   ├── apps.py
│   │   ├── models.py
│   │   ├── tests.py
│   │   └── views.py
│   ├── <project>
│   │   ├── __init__.py
│   │   ├── asgi.py
│   │   ├── settings.py
│   │   ├── urls.py
│   │   └── wsgi.py
│   └── manage.py
```

## Adding remaining assets and assets folder

Before we move any further, I would like you to test your project by opening a cmd at the location where **manage.py** is located and open a cmd window here (PowerShell will work as well) and type in **py manage.py runserver**. This will start a development server at your local machine and will be hosted at [localhost:8000](http://localhost:8000) . It should display something like this.

django

View [release notes](#) for Django 4.0



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

Now that you've created everything to basically run the project, its now time to add the assets to your project. You can get the files I've used at my [GitHub](#) page or add your own files. Copy the directory structure given below.

```
.
├── <project_name>
│   ├── app
│   │   ├── migrations
│   │   ├── __init__.py
│   │   ├── admin.py
│   │   ├── apps.py
│   │   ├── models.py
│   │   ├── tests.py
│   │   └── views.py
│   ├── media
│   │   └── productimg
│   │       └── <your image files> <- This isn't a folder or a file
│   └── <project>
│       ├── __init__.py
│       ├── asgi.py
│       ├── settings.py
│       └── urls.py
```

```
|   wsgi.py
|   manage.py
```

## Step 2 | Coding

### *Editing settings.py*

Now that you've setup the stuff, its time to edit the **settings.py** file. Refer my [GitHub](#) for more details on it.

Add this (40) line to **INSTALLED\_APPS** block

```
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      'app',
41  ]
```

Add this (58) line to **TEMPLATES** block

```
55  TEMPLATES = [
56      {
57          'BACKEND': 'django.template.backends.django.DjangoTemplates',
58          'DIRS': [BASE_DIR / 'templates'],
59          'APP_DIRS': True,
60          'OPTIONS': {
61              'context_processors': [
62                  'django.template.context_processors.debug',
63                  'django.template.context_processors.request',
64                  'django.contrib.auth.context_processors.auth',
65                  'django.contrib.messages.context_processors.messages',
66              ],
67          },
68      },
69  ]
```

Add this (113) line to last bit of code

```

107 LANGUAGE_CODE = 'en-us'
108
109 TIME_ZONE = 'UTC'
110
111 USE_I18N = True
112
113 USE_L10N = True
114
115 USE_TZ = True

```

Add this before **DEFAULT\_AUTO\_FIELD**

```

118 # Static files (CSS, JavaScript, Images)
119 # https://docs.djangoproject.com/en/4.0/howto/static-files/
120
121 STATIC_URL = 'static/'
122 MEDIA_URL = '/media/'
123 MEDIA_ROOT = BASE_DIR / 'media'
124 LOGIN_REDIRECT_URL = '/profile/'
125 EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
126 SESSION_ENGINE = "django.contrib.sessions.backends.db"

```

Now you've completely edited the **settings.py** file to your needs

### *Editing urls.py*

Now we will edit **urls.py** file

Edit the file to resemble this

```

1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('app.urls')),
7 ]
8
9 handler404 = "shopx.views.page_not_found_view"

```

## Step 3 | Working with *apps* folder

### *Creating files needed*

Copy the directory provided below to add the files needed. You could copy the files off my [GitHub](#).

```

.
└─ app

```

```
├── migrations
├── static
│   ├── app
│   │   ├── css
│   │   │   ├── all.min.css
│   │   │   ├── owl.carousel.min.css
│   │   │   └── style.css
│   │   ├── images
│   │   │   ├── banner
│   │   │   │   ├── b1.png
│   │   │   ├── pavavail
│   │   │   │   ├── upi.png
│   │   │   ├── product
│   │   │   │   ├── 1.png
│   │   │   ├── dsd.png
│   │   │   ├── emptycart.png
│   │   │   └── payment.png
│   │   └── js
│   │       ├── all.min.js
│   │       ├── myscript.js
│   │       └── owl.carousel.min.js
├── templates
├── __init__.py
├── admin.py
├── apps.py
├── forms.py
├── models.py
├── tests.py
├── urls.py
└── views.py
```

### *Editing admin.py*

Add these lines to your **admin.py** file

```

1 from django.contrib import admin
2 from django.utils.html import format_html
3 from django.urls import reverse
4 from .models import (
5     Customer,
6     Product,
7     Cart,
8     OrderPlaced
9 )
10
11 @admin.register(Customer)
12 class CustomerModelAdmin(admin.ModelAdmin):
13     list_display = ['id', 'user', 'name', 'locality', 'city', 'zipcode', 'state']
14
15 @admin.register(Product)
16 class ProductModelAdmin(admin.ModelAdmin):
17     list_display = ['id', 'title', 'selling_price', 'discounted_price', 'description', 'brand', 'category', 'product_image']
18
19 @admin.register(Cart)
20 class CartModelAdmin(admin.ModelAdmin):
21     list_display = ['id', 'user', 'product', 'quantity']
22
23
24 @admin.register(OrderPlaced)
25 class OrderPlacedModelAdmin(admin.ModelAdmin):
26     list_display = ['id', 'user', 'customer', 'customer_info', 'product', 'product_info', 'quantity', 'ordered_date', 'status']
27
28     def product_info(self, obj):
29         link = reverse("admin:app_product_change", args=[obj.product.pk])
30         return format_html('<a href="{0}">{1}</a>', link, obj.product.title)
31
32     def customer_info(self, obj):
33         link = reverse("admin:app_customer_change", args=[obj.customer.pk])
34         return format_html('<a href="{0}">{1}</a>', link, obj.customer.name)

```

### Editing forms.py

Now we will add forms to our project. We will start by importing all the necessary modules. Add these lines at the start of the document and *replacing any existing ones (optional)*

```

1 from django import forms
2 from django.contrib.auth.forms import UserCreationForm, AuthenticationForm, PasswordChangeForm, UsernameField, PasswordResetForm, SetPasswordForm
3 from django.contrib.auth.models import User
4 from django.utils.translation import gettext, gettext_lazy as _
5 from .models import Customer
6 from django.contrib.auth import password_validation

```

Now we will start adding our classes. First off let's add **CustomerRegistrationForm**. This will be used in customer's registration process.

```

8 class CustomerRegistrationForm(UserCreationForm):
9     password1 = forms.CharField(label='Password', widget=forms.PasswordInput(attrs={'class': 'form-control'}))
10    password2 = forms.CharField(label='Confirm Password (again)', widget=forms.PasswordInput(attrs={'class': 'form-control'}))
11    email = forms.CharField(required=True, widget=forms.EmailInput(attrs={'class': 'form-control'}))
12    class Meta:
13        model = User
14        fields = ['username', 'email', 'password1', 'password2']
15        labels = {'email': 'Email'}
16        widgets = {'username': forms.TextInput(attrs={'class': 'form-control'})}

```

Second, we will add **LoginForm**. As the name suggests, this will be used in login process.

```

18 class LoginForm(AuthenticationForm):
19     username = UsernameField(widget=forms.TextInput(attrs={'autofocus': True, 'class': 'form-control'}))
20     password = forms.CharField(label=_('Password'), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'current-password', 'class': 'form-control'}))

```

Now we will add classes to manage passwords, such as **changing, resetting and setting a new password**. Add these lines

```

22 class MyPasswordChangeForm(PasswordChangeForm):
23     old_password = forms.CharField(label=_("Old Password"), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'current-password', 'autofocus': True, 'class': 'form-control'}))
24     new_password1 = forms.CharField(label=_("New Password"), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}), help_text=password_validation.password_validators_help_text_html())
25     new_password2 = forms.CharField(label=_("Confirm New Password"), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}))
26
27 class MyPasswordResetForm(PasswordResetForm):
28     email = forms.EmailField(label=_("Email"), max_length=254, widget=forms.EmailInput(attrs={'autocomplete': 'email', 'class': 'form-control'}))
29
30 class MySetPasswordForm(SetPasswordForm):
31     new_password1 = forms.CharField(label=_("New Password"), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}), help_text=password_validation.password_validators_help_text_html())
32     new_password2 = forms.CharField(label=_("Confirm New Password"), strip=False, widget=forms.PasswordInput(attrs={'autocomplete': 'new-password', 'class': 'form-control'}))
33

```

Now we will add **CustomerProfileForm**

```

34 class CustomerProfileForm(forms.ModelForm):
35     class Meta:
36         model = Customer
37         fields = ['name', 'locality', 'city', 'state', 'zipcode']
38         widgets = {'name': forms.TextInput(attrs={'class': 'form-control'}), 'locality': forms.TextInput(attrs={'class': 'form-control'}), 'city': forms.TextInput(attrs={'class': 'form-control'}), 'state': forms.Select(attrs={'class': 'form-control'}), 'zipcode': forms.NumberInput(attrs={'class': 'form-control'})}
39
40

```

### *Editing models.py*

Now we will start editing **models.py** file. Start with importing necessary modules. Add these lines to the start of the file.

```

1 from django.db import models
2 from django.contrib.auth.models import User
3 from django.core.validators import MaxValueValidator, MinValueValidator

```

Now we will add a tuple with all states from India.



```

4  STATE_CHOICES = (
5      ('Andaman & Nicobar Islands','Andaman & Nicobar Islands'),
6      ('Andhra Pradesh','Andhra Pradesh'),
7      ('Arunachal Pradesh','Arunachal Pradesh'),
8      ('Assam','Assam'),
9      ('Bihar','Bihar'),
10     ('Chandigarh','Chandigarh'),
11     ('Chhattisgarh','Chhattisgarh'),
12     ('Dadra & Nagar Haveli','Dadra & Nagar Haveli'),
13     ('Daman and Diu','Daman and Diu'),
14     ('Delhi','Delhi'),
15     ('Goa','Goa'),
16     ('Gujarat','Gujarat'),
17     ('Haryana','Haryana'),
18     ('Himachal Pradesh','Himachal Pradesh'),
19     ('Jammu & Kashmir','Jammu & Kashmir'),
20     ('Jharkhand','Jharkhand'),
21     ('Karnataka','Karnataka'),
22     ('Kerala','Kerala'),
23     ('Lakshadweep','Lakshadweep'),
24     ('Madhya Pradesh','Madhya Pradesh'),
25     ('Maharashtra','Maharashtra'),
26     ('Manipur','Manipur'),
27     ('Meghalaya','Meghalaya'),
28     ('Mizoram','Mizoram'),
29     ('Nagaland','Nagaland'),
30     ('Odisha','Odisha'),
31     ('Puducherry','Puducherry'),
32     ('Punjab','Punjab'),
33     ('Rajasthan','Rajasthan'),
34     ('Sikkim','Sikkim'),
35     ('Tamil Nadu','Tamil Nadu'),
36     ('Telangana','Telangana'),
37     ('Tripura','Tripura'),
38     ('Uttarakhand','Uttarakhand'),
39     ('Uttar Pradesh','Uttar Pradesh'),
40     ('West Bengal','West Bengal'),
41 )

```

Now we will add **Customer** class. This will be used to define the database entries.

```

42 class Customer(models.Model):
43     user = models.ForeignKey(User, on_delete=models.CASCADE)
44     name = models.CharField(max_length=200)
45     locality = models.CharField(max_length=200)
46     city = models.CharField(max_length=50)
47     zipcode = models.IntegerField()
48     state = models.CharField(choices=STATE_CHOICES, max_length=50)
49
50     def __str__(self):
51         # return self.user.username
52         return str(self.id)

```

Now we will add **CATEGORY\_CHOICES** tuple and **Product** class.

```
55 CATEGORY_CHOICES = (  
56     ('M', 'Mobile'),  
57     ('L', 'Laptop'),  
58     ('TW', 'Top Wear'),  
59     ('BW', 'Bottom Wear'),  
60 )  
61 class Product(models.Model):  
62     title = models.CharField(max_length=100)  
63     selling_price = models.FloatField()  
64     discounted_price = models.FloatField()  
65     description = models.TextField()  
66     brand = models.CharField(max_length=100)  
67     category = models.CharField( choices=CATEGORY_CHOICES, max_length=2)  
68     product_image = models.ImageField(upload_to='producting')  
69  
70     def __str__(self):  
71         return str(self.id)
```

Now we will add **OrderPlaced** class and the **property**. Note that the image shows a comment stating use of **orders.html**. We will create that file later.

```
95 class OrderPlaced(models.Model):  
96     user = models.ForeignKey(User, on_delete=models.CASCADE)  
97     customer = models.ForeignKey(Customer, on_delete=models.CASCADE)  
98     product = models.ForeignKey(Product, on_delete=models.CASCADE)  
99     quantity = models.PositiveIntegerField(default=1)  
100     ordered_date = models.DateTimeField(auto_now_add=True)  
101     status = models.CharField(max_length=50, choices=STATUS_CHOICES, default='Pending')  
102  
103     # Below Property will be used by orders.html page to show total cost  
104     @property  
105     def total_cost(self):  
106         return self.quantity * self.product.discounted_price
```

### *Editing urls.py*

Now that we have created almost all of the backend, now we need to link the ... well, links to show the website's interface (*created later on*) on specified web addresses. First off, we will import necessary stuff needed to write the `urls.py` file in first place :D. Add these lines at the start.

```
1 from django.urls import path  
2 from app import views  
3 from django.conf import settings  
4 from django.conf.urls.static import static  
5 from django.contrib.auth import views as auth_views  
6 from .forms import LoginForm, MyPasswordChangeForm, MyPasswordResetForm, MySetPasswordForm
```

Then we need to specify all the links with redirects to a special list named **urlpatterns**. Add them like below mentioned

```
7 urlpatterns = [  
8     # path('', views.home),  
9     path('', views.ProductView.as_view(), name="home"),  
10    # path('product-detail', views.product_detail, name='product-detail'),  
11    path('product-detail/<int:pk>', views.ProductDetailView.as_view(), name='product-detail'),  
12    path('add-to-cart/', views.add_to_cart, name='add-to-cart'),  
13    path('cart/', views.show_cart, name='showcart'),  
14    path('pluscart/', views.plus_cart),  
15    path('minuscart/', views.minus_cart),  
16    path('removecart/', views.remove_cart),  
17    path('checkout/', views.checkout, name='checkout'),  
18    path('address/', views.address, name='address'),  
19    path('orders/', views.orders, name='orders'),  
20    path('paymentdone/', views.payment_done, name='paymentdone'),  
21  
22    path('mobile/', views.mobile, name='mobile'),  
23    path('mobile/<slug:data>', views.mobile, name='mobiledata'),  
24  
25    path('laptop/', views.laptop, name='laptop'),  
26    path('laptop/<slug:data>', views.laptop, name='laptopdata'),  
27  
28    path('accounts/login/', auth_views.LoginView.as_view(template_name='app/login.html',  
29    authentication_form=LoginForm), name='login'),  
30    # path('profile/', views.profile, name='profile'),  
31    path('profile/', views.ProfileView.as_view(), name='profile'),  
32    path('logout/', auth_views.LogoutView.as_view(next_page='login'), name='logout'),  
33    path('passwordchange/', auth_views.PasswordChangeView.as_view(template_name='app/  
passwordchange.html', form_class=MyPasswordChangeForm, success_url='/passwordchangedone/'),  
34    name='passwordchange'),  
35    path('passwordchangedone/', auth_views.PasswordChangeDoneView.as_view(template_name='app/  
passwordchangedone.html'), name='passwordchangedone'),  
36  
37    path("password-reset/", auth_views.PasswordResetView.as_view(template_name='app/  
password_reset.html', form_class=MyPasswordResetForm), name="password_reset"),  
38    path("password-reset/done/", auth_views.PasswordResetDoneView.as_view(template_name='app/  
password_reset_done.html'), name="password_reset_done"),  
39    path("password-reset-confirm/<uidb64>/<token>/", auth_views.PasswordResetConfirmView.as_view(  
40    template_name='app/password_reset_confirm.html', form_class=MySetPasswordForm), name="  
password_reset_confirm"),  
41    path("password-reset-complete/", auth_views.PasswordResetCompleteView.as_view(template_name='app/  
password_reset_complete.html'), name="password_reset_complete"),  
42  
43    path('registration/', views.CustomerRegistrationView.as_view(), name='customerregistration')  
44 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Here we have now setup all the links with their respective redirects.

### Editing views.py

Now we can work on views of our app, the most essential thing of our project so far. As always, first we will import everything. Start by coding this into start portion of **views.py** file.

```
1 from django.shortcuts import render, redirect, HttpResponseRedirect
2 from .models import Customer, Product, Cart, OrderPlaced
3 from .forms import CustomerRegistrationForm, CustomerProfileForm
4 from django.views import View
5 from django.contrib import messages
6 from django.http import JsonResponse
7 from django.db.models import Q
8 from django.contrib.auth.decorators import login_required
9 from django.utils.decorators import method_decorator
```

Then we can code our **ProductView** view. Here we call **home.html** (*yet to be made*) in **return render** tag.

```
11 class ProductView(View):
12     def get(self, request):
13         totalitem = 0
14         topwears = Product.objects.filter(category='TW')
15         bottomwears = Product.objects.filter(category='BW')
16         mobiles = Product.objects.filter(category='M')
17         if request.user.is_authenticated:
18             totalitem = len(Cart.objects.filter(user=request.user))
19         return render(request, 'app/home.html', {'topwears':topwears, 'bottomwears':bottomwears, '
            mobiles':mobiles, 'totalitem':totalitem})
```

Then we will code **ProductDetailView** view.

```
21 class ProductDetailView(View):
22     def get(self, request, pk):
23         totalitem = 0
24         product = Product.objects.get(pk=pk)
25         print(product.id)
26         item_already_in_cart=False
27         if request.user.is_authenticated:
28             totalitem = len(Cart.objects.filter(user=request.user))
29             item_already_in_cart = Cart.objects.filter(Q(product=product.id) & Q(user=request.user)).
                exists()
30         return render(request, 'app/productdetail.html', {'product':product, 'item_already_in_cart':
            item_already_in_cart, 'totalitem':totalitem})
```

Now we code **add\_to\_cart** view. Note that we have also included a **@login\_required()** tag, which tells the program to run this code only when asked by the user and the user is logged in otherwise it will

throw an error

```
32 @login_required()
33 def add_to_cart(request):
34     user = request.user
35     item_already_in_cart1 = False
36     product = request.GET.get('prod_id')
37     item_already_in_cart1 = Cart.objects.filter(Q(product=product) & Q(user=request.user)).exists()
38     if item_already_in_cart1 == False:
39         product_title = Product.objects.get(id=product)
40         Cart(user=user, product=product_title).save()
41         messages.success(request, 'Product Added to Cart Successfully !!' )
42         return redirect('/cart')
43     else:
44         return redirect('/cart')
45     # Below Code is used to return to same page
46     # return redirect(request.META['HTTP_REFERER'])
```

Then we create the following views step by step

```
48 @login_required
49 def show_cart(request):
50     totalitem = 0
51     if request.user.is_authenticated:
52         totalitem = len(Cart.objects.filter(user=request.user))
53         user = request.user
54         cart = Cart.objects.filter(user=user)
55         amount = 0.0
56         shipping_amount = 70.0
57         totalamount=0.0
58         cart_product = [p for p in Cart.objects.all() if p.user == request.user]
59         print(cart_product)
60         if cart_product:
61             for p in cart_product:
62                 tempamount = (p.quantity * p.product.discounted_price)
63                 amount += tempamount
64                 totalamount = amount+shipping_amount
65             return render(request, 'app/addtocart.html', {'carts':cart, 'amount':amount, 'totalamount':totalamount, 'totalitem':totalitem})
66         else:
67             return render(request, 'app/emptycart.html', {'totalitem':totalitem})
68     else:
69         return render(request, 'app/emptycart.html', {'totalitem':totalitem})
71 def plus_cart(request):
72     if request.method == 'GET':
73         prod_id = request.GET['prod_id']
74         c = Cart.objects.get(Q(product=prod_id) & Q(user=request.user))
75         c.quantity+=1
76         c.save()
77         amount = 0.0
78         shipping_amount= 70.0
79         cart_product = [p for p in Cart.objects.all() if p.user == request.user]
80         for p in cart_product:
81             tempamount = (p.quantity * p.product.discounted_price)
82             # print("Quantity", p.quantity)
83             # print("Selling Price", p.product.discounted_price)
84             # print("Before", amount)
85             amount += tempamount
86             # print("After", amount)
87         # print("Total", amount)
88         data = {
89             'quantity':c.quantity,
90             'amount':amount,
91             'totalamount':amount+shipping_amount
92         }
93         return JsonResponse(data)
94     else:
95         return HttpResponse("")
```

```

97 def minus_cart(request):
98     if request.method == 'GET':
99         prod_id = request.GET['prod_id']
100         c = Cart.objects.get(Q(product=prod_id) & Q(user=request.user))
101         c.quantity-=1
102         c.save()
103         amount = 0.0
104         shipping_amount= 70.0
105         cart_product = [p for p in Cart.objects.all() if p.user == request.user]
106         for p in cart_product:
107             tempamount = (p.quantity * p.product.discounted_price)
108             # print("Quantity", p.quantity)
109             # print("Selling Price", p.product.discounted_price)
110             # print("Before", amount)
111             amount += tempamount
112             # print("After", amount)
113         # print("Total", amount)
114         data = {
115             'quantity':c.quantity,
116             'amount':amount,
117             'totalamount':amount+shipping_amount
118         }
119         return JsonResponse(data)
120     else:
121         return HttpResponse("")
122
123 @login_required
124 def checkout(request):
125     user = request.user
126     add = Customer.objects.filter(user=user)
127     cart_items = Cart.objects.filter(user=request.user)
128     return render(request, 'app/checkout.html', {'add':add, 'cart_items':cart_items})
129
130 @login_required
131 def payment_done(request):
132     custid = request.GET.get('custid')
133     print("Customer ID", custid)
134     user = request.user
135     cartid = Cart.objects.filter(user = user)
136     customer = Customer.objects.get(id=custid)
137     print(customer)
138     for cid in cartid:
139         OrderPlaced(user=user, customer=customer, product=cid.product, quantity=cid.quantity).save()
140         print("Order Saved")
141         cid.delete()
142         print("Cart Item Deleted")
143     return redirect("orders")
144

```

```

145 def remove_cart(request):
146     if request.method == 'GET':
147         prod_id = request.GET['prod_id']
148         c = Cart.objects.get(Q(product=prod_id) & Q(user=request.user))
149         c.delete()
150         amount = 0.0
151         shipping_amount= 70.0
152         cart_product = [p for p in Cart.objects.all() if p.user == request.user]
153         for p in cart_product:
154             tempamount = (p.quantity * p.product.discounted_price)
155             # print("Quantity", p.quantity)
156             # print("Selling Price", p.product.discounted_price)
157             # print("Before", amount)
158             amount += tempamount
159             # print("After", amount)
160         # print("Total", amount)
161         data = {
162             'amount':amount,
163             'totalamount':amount+shipping_amount
164         }
165         return JsonResponse(data)
166     else:
167         return HttpResponse("")
168
169 @login_required
170 def address(request):
171     totalitem = 0
172     if request.user.is_authenticated:
173         totalitem = len(Cart.objects.filter(user=request.user))
174         add = Customer.objects.filter(user=request.user)
175         return render(request, 'app/address.html', {'add':add, 'active':'btn-primary', 'totalitem':
            totalitem})
176
177 @login_required
178 def orders(request):
179     op = OrderPlaced.objects.filter(user=request.user)
180     return render(request, 'app/orders.html', {'order_placed':op})
181
182 def mobile(request, data=None):
183     totalitem = 0
184     if request.user.is_authenticated:
185         totalitem = len(Cart.objects.filter(user=request.user))
186     if data==None :
187         mobiles = Product.objects.filter(category='M')
188     elif data == 'Redmi' or data == 'Samsung':
189         mobiles = Product.objects.filter(category='M').filter(brand=data)
190     elif data == 'below':
191         mobiles = Product.objects.filter(category='M').filter(discounted_price__lt=10000)
192     elif data == 'above':
193         mobiles = Product.objects.filter(category='M').filter(discounted_price__gt=10000)
194     return render(request, 'app/mobile.html', {'mobiles':mobiles, 'totalitem':totalitem})
195
196 def laptop(request, data=None):
197     totalitem = 0
198     if request.user.is_authenticated:
199         totalitem = len(Cart.objects.filter(user=request.user))
200     if data==None :
201         laptops = Product.objects.filter(category='L')
202     elif data == 'HP' or data == 'Dell':
203         laptops = Product.objects.filter(category='L').filter(brand=data)
204     elif data == 'below':
205         laptops = Product.objects.filter(category='L').filter(discounted_price__lt=10000)
206     elif data == 'above':
207         laptops = Product.objects.filter(category='L').filter(discounted_price__gt=10000)
208     return render(request, 'app/laptop.html', {'laptops':laptop, 'totalitem':totalitem})
209

```



```

211 class CustomerRegistrationView(View):
212     def get(self, request):
213         form = CustomerRegistrationForm()
214         return render(request, 'app/customerregistration.html', {'form':form})
215
216     def post(self, request):
217         form = CustomerRegistrationForm(request.POST)
218         if form.is_valid():
219             messages.success(request, 'Congratulations!! Registered Successfully.')
220             form.save()
221         return render(request, 'app/customerregistration.html', {'form':form})
222

```

Note that here we add a new tag called **@method\_decorator**.

To decorate every instance of a class-based view, you need to decorate the class definition itself. To do this you apply the decorator to the dispatch() method of the class.

A method on a class isn't quite the same as a standalone function, so you can't just apply a function decorator to the method – you need to transform it into a method decorator first.

The **method\_decorator** decorator transforms a function decorator into a method decorator so that it can be used on an instance method.

More info at [docs@django](https://docs.djangoproject.com/en/1.11/topics/decorators/#method-decorators)

```

223 @method_decorator(login_required, name='dispatch')
224 class ProfileView(View):
225     def get(self, request):
226         totalitem = 0
227         if request.user.is_authenticated:
228             totalitem = len(Cart.objects.filter(user=request.user))
229         form = CustomerProfileForm()
230         return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary', 'totalitem':
                totalitem})
231
232     def post(self, request):
233         totalitem = 0
234         if request.user.is_authenticated:
235             totalitem = len(Cart.objects.filter(user=request.user))
236         form = CustomerProfileForm(request.POST)
237         if form.is_valid():
238             usr = request.user
239             name = form.cleaned_data['name']
240             locality = form.cleaned_data['locality']
241             city = form.cleaned_data['city']
242             state = form.cleaned_data['state']
243             zipcode = form.cleaned_data['zipcode']
244             reg = Customer(user=usr, name=name, locality=locality, city=city, state=state, zipcode=
                zipcode)
245             reg.save()
246             messages.success(request, 'Congratulations!! Profile Updated Successfully.')
247         return render(request, 'app/profile.html', {'form':form, 'active':'btn-primary', 'totalitem':
                totalitem})
248

```

Now before moving any further, I'd recommend you to apply your migrations. To do so, open a cmd window at your **manage.py** location and run **py manage.py migrate**. This will make sure that there are no issues about any migrations while running the project. Now running the [runserver](#) command will display an error as we haven't yet created the html files. So, let's just do that exact thing!



## Step 4 | Creating HTML files

### *Coding base.html*

Now that we have done all of our backend (*I hope so*), let's work on our front-end!

First off, we need to create **base.html** at **templates** folder inside **app** folder. Including other html files mentioned in this documentation, the file structure should look something like this. Feel free to create these files before hand as that will save some of your precious seconds.

```
.
├── templates
│   └── app
│       ├── address.html
│       ├── addtocart.html
│       ├── base.html <- The file we are going to create first
│       ├── checkout.html
│       ├── customerregistration.html
│       ├── emptycart.html
│       ├── home.html
│       ├── laptop.html
│       ├── mobile.html
│       ├── orders.html
│       ├── password_reset.html
│       ├── password_reset_complete.html
│       ├── password_reset_confirm.html
│       ├── password_reset_done.html
│       ├── passwordchange.html
│       ├── passwordchangedone.html
│       ├── productdetail.html
│       └── profile.html
```

Now we will begin with **base.html**

Here we load the **static** URL defined in **urls.py** file to utilize the static files. Here we also complete our **<head>** part of our **base.html**.

```

1 <!doctype html>
2 {% load static %}
3 <html lang="en">
4   <head>
5     <!-- Required meta tags -->
6     <meta charset="utf-8">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8
9     <!-- Bootstrap CSS -->
10    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="
stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmp1"
crossorigin="anonymous">
11
12    <!-- Owl Carousel CSS -->
13    <link rel="stylesheet" href="{% static 'app/css/owl.carousel.min.css' %}">
14
15    <!-- Font Awesome CSS -->
16    <link rel="stylesheet" href="{% static 'app/css/all.min.css' %}">
17
18    <!-- Custom CSS -->
19    <link rel="stylesheet" href="{% static 'app/css/style.css' %}">
20
21
22    <title>ShopX | {% block title %} {% endblock title %} </title>
23  </head>

```

Then we code the remaining part of our file as follows

```

22     <title>ShopX | {% block title %} {% endblock title %} </title>
23 </head>
24 <body>
25     <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
26         <div class="container">
27             <a class="navbar-brand" href="/">ShopX</a>
28             <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="
#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
29                 <span class="navbar-toggler-icon"></span>
30             </button>
31             <div class="collapse navbar-collapse" id="navbarSupportedContent">
32                 <ul class="navbar-nav me-auto mb-2 mb-lg-0">
33                     <li class="nav-item">
34                         <a class="nav-link active" aria-current="page" href="/">Home</a>
35                     </li>
36                     <li class="nav-item dropdown">
37                         <a class="nav-link dropdown-toggle text-white" href="#" id="electronicsDropdown" role="
button" data-bs-toggle="dropdown" aria-expanded="false">
38                             Electronics
39                         </a>
40                         <ul class="dropdown-menu" aria-labelledby="electronicsDropdown">
41                             <li><a class="dropdown-item" href="{% url 'mobile' %}">Mobile</a></li>
42                             <li><a class="dropdown-item" href="{% url 'laptop' %}">Laptop</a></li>
43                         </ul>
44                     </li>
45                     <li class="nav-item dropdown">
46                         <a class="nav-link dropdown-toggle text-white" href="#" id="fashionDropdown" role="
button" data-bs-toggle="dropdown" aria-expanded="false">
47                             Fashion
48                         </a>
49                         <ul class="dropdown-menu" aria-labelledby="fashionDropdown">
50                             <li><a class="dropdown-item" href="#">Top Wear</a></li>
51                             <li><a class="dropdown-item" href="#">Bottom Wear</a></li>
52                         </ul>
53                     </li>
54                 </ul>
55                 <form class="d-flex">
56                     <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
57                     <button class="btn btn-warning" type="submit">S</button>
58                 </form>
59                 <div>
60                     <ul class="navbar-nav me-auto mb-2 mb-lg-0">
61                         {% if request.user.is_authenticated %}
62                         <li class="nav-item dropdown mx-2">
63                             <a class="nav-link dropdown-toggle text-white" href="#" id="profileDropdown" role="
button" data-bs-toggle="dropdown" aria-expanded="false">
64                                 {{request.user.username|capfirst}}
65                             </a>
66                             <ul class="dropdown-menu" aria-labelledby="profileDropdown">

```

```

67         <li><a class="dropdown-item" href="{% url 'profile' %}">Profile</a></li>
68         <li><a class="dropdown-item" href="{% url 'orders' %}">Orders</a></li>
69         <li><a class="dropdown-item" href="{% url 'passwordchange' %}">Change Password</a></li>
70         <li><a class="dropdown-item" href="{% url 'logout' %}">Logout</a></li>
71     </ul>
72 </li>
73 <li class="nav-item mx-2">
74     <a href="{% url 'showcart' %}" class="nav-link text-white"><span class="badge bg-danger">{{totalitem}}</span> Cart </a>
75 </li>
76 {% else %}
77 <li class="nav-item mx-2">
78     <a href="{% url 'login' %}" class="nav-link text-white">Login</a>
79 </li>
80 <li class="nav-item mx-2">
81     <a href="{% url 'customerregistration' %}" class="nav-link text-white">Registration</a>
82 </li>
83 {% endif %}
84 </ul>
85 </div>
86 </div>
87 </div>
88 </nav>

```

Now we add our footer to our **base.html**

```

89 {% block banner_slider %} {% endblock banner_slider %}
90 {% block livesale %} {% endblock livesale %}
91 {% block main-content %} {% endblock main-content %}
92
93 <!-- Start Footer -->
94 <footer class="container-fluid bg-dark text-center p-2 mt-5">
95     <small class="text-white">Copyright &copy; 2022 || Modified by Aman || </small>
96     
97 </footer> <!-- End Footer -->
98
99 <!-- JQuery -->
100 <script src="https://code.jquery.com/jquery-3.5.1.min.js" integrity="sha256-9/aliU8dGd2tb60SsuzixeV4y/faTqgFtohetphbbj0=" crossorigin="anonymous"></script>
101 <!-- Bootstrap Bundle with Popper -->
102 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf800JFdroafW" crossorigin="anonymous"></script>
103
104 <script src="{% static 'app/js/owl.carousel.min.js' %}"></script>
105 <script src="{% static 'app/js/all.min.js' %}"></script>
106 <script src="{% static 'app/js/myscript.js' %}"></script>
107 </body>
108 </html>

```

## Coding other .html files

### address.html

Here you can see, we have included a special tag

**{% extends 'app/base.html' %}**

```

1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Address{% endblock title %}
4 {% block main-content %}

```

What this tag does is it loads our base.html as ... well our base code and adds the following code after <head> block.

Now we add the remaining code to the file

```
5 <div class="container my-5">
6   <div class="row">
7     <h3>Welcome <span class="text-capitalize">{{request.user}}</span></h3>
8     <div class="col-sm-2 border-end">
9       <ul class="list-unstyled">
10        <li class="d-grid"><a href="{% url 'profile' %}" class="btn">Profile</a></li>
11        <li class="d-grid"><a href="{% url 'address' %}" class="btn {{active}}">Address</a></li>
12      </ul>
13    </div>
14    <div class="col-sm-9 offset-sm-1">
15      <div class="row">
16        {% for ad in add %}
17          <div class="col-sm-6">
18            <div class="card">
19              <div class="card-body">
20                <h3>Address: {{ forloop.counter }}</h3>
21                <p>Name: {{ad.name}}</p>
22                <p>Locality: {{ad.locality}}</p>
23                <p>City: {{ad.city}}</p>
24                <p>State: {{ad.state}}</p>
25                <p>Pin Code: {{ad.zipcode}}</p>
26              </div>
27            </div>
28          </div>
29          {% endfor %}
30        </div>
31      </div>
32    </div>
33  </div>
34  {% endblock main-content %}
```

### *addtocart.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Cart{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   {% comment %} Below Code will be used by add_to_cart view {% endcomment %}
7   {% if messages %}
8   {% for message in messages %}
9     <p {% if message.tags %} class="alert alert-{{message.tags}} mb-5" {% endif %}>{{message}}</p>
10   {% endfor %}
11   {% endif %}
12   <div class="row">
13     <h1 class="text-center mb-5">Shopping Cart</h1>
14     <div class="col-sm-8">
15       <div class="card">
16         <div class="card-body">
17           <h3>Cart Items</h3>
18           {% for cart in carts %}
19             <hr>
20             <div class="row">
21               <div class="col-sm-3 text-center align-self-center"> </div>
23               <div class="col-sm-9">
24                 <div>
25                   <h5>{{cart.product.title}}</h5>
26                   <p class="mb-2 text-muted small">{{cart.product.description}}</p>
27                   <div class="my-3">
28                     <label for="quantity">Quantity:</label>
29                     <a class="minus-cart btn" pid="{{cart.product.id}}"><i class="fas fa-minus-square fa-lg"></i>
30                     <span id="quantity">{{cart.quantity}}</span>
31                     <a class="plus-cart btn" pid="{{cart.product.id}}"><i class="fas fa-plus-square fa-lg"></i></a>
32                   </div>
33                   <div class="d-flex justify-content-between">
34                     <a class="btn btn-sm btn-secondary mr-3 remove-cart" pid="{{cart.product.id}}">Remove item </a>
35                     <p class="mb-0"><span><strong>Rs. {{cart.product.discounted_price}}</strong></span></p>
36                   </div>
37                 </div>
38               </div>
39             </div>
40           {% endfor %}
41         </div>
42       </div>
43     </div>
44   </div>
45   <div class="col-sm-4">
46     <div class="card">
```

```

48     <div class="card-body">
49         <h3>The Total Amount of</h3>
50         <ul class="list-group">
51             <li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0 pb-0">
52                 Amount<span>Rs. <span id="amount">{{amount|floatformat:2}}</span> </span></li>
53             <li class="list-group-item d-flex justify-content-between align-items-center border-0 px-0">
54                 Shipping<span>Rs. 70.00</span></li>
55             <div>
56                 <strong>Total</strong> <small>(including VAT)</small>
57             </div>
58             <span><strong>Rs. </strong><strong id="totalamount"> {{totalamount|floatformat:2}}</strong></span>
59         </li>
60     </ul>
61     <div class="d-grid"><a href="{% url 'checkout' %}" class="btn btn-danger fw-bold">Place Order</a>
62 </div>
63 </div>
64 </div>
65
66 </div>
67 </div>
68 <div class="container">
69     <div class="row">
70         <div class="col-sm-8">
71             <div class="card">
72                 <div class="card-body">
73                     <h5 class="mb-4">We accept</h5>
74                     
75                 </div>
76             </div>
77         </div>
78     </div>
79 </div>
80 {% endblock main-content %}
81

```

*checkout.html*

```
{% extends 'app/base.html' %}
{% load static %}
{% block title %}Checkout{% endblock title %}
{% block main-content %}
<div class="container">
  <div class="row mt-5">
    <div class="col-sm-6">
      <h4>Order Summary</h4>
      <hr>
      {% for item in cart_items %}
        <div class="card mb-2">
          <div class="card-body">
            <h5>Product: {{item.product.title}}</h5>
            <p>Quantity: {{item.quantity}}</p>
            <p class="fw-bold">Price: {{item.total_cost}}</p>
          </div>
        </div>
      {% endfor %}
      <small>Terms and Conditions: You are hereby accepting with our extended
      terms and conditions while making transactions with our service.</small>
    </div>
    <div class="col-sm-4 offset-sm-1">
      <h4>Select Shipping Address</h4>
      <hr>
      <form action="/paymentdone">
        {% for ad in add %}
          <div class="card">
            <div class="card-body">
              <h5>{{ad.name}}</h5>
              <p>{{ad.locality}}, {{ad.city}}, {{ad.state}}- {{ad.zipcode}}</p>
            </div>
          </div>
          <div class="form-check mt-2 mb-5">
            <input class="form-check-input" type="radio" name="custid" id="
            custadd{{forloop.counter}}" value="{{ad.id}}">
            <label class="form-check-label fw-bold" for="
            custadd{{forloop.counter}}">
              Address: {{ forloop.counter }} </label>
          </div>
        {% endfor %}
        <div class="text-end">
          <button type="submit" class="btn btn-warning mt-3 px-5 fw-bold">
            Continue</button>
        </div>
      </form>
    </div>
  </div>
</div>
{% endblock main-content %}
```



### *customerregistration.html*

```
{% extends 'app/base.html' %}
{% load static %}
{% block title %}Customer Registration{% endblock title %}
{% block main-content %}
<div class="container">
  <div class="row my-3">
    <div class="col-sm-6 offset-sm-3">
      <h3>Customer Registration</h3>
      <hr>

      <form action="" method="post" novalidate class="shadow p-5">
        {% if messages %}
        {% for message in messages %}
          <p {% if message.tags %} class="alert alert-{{message.tags}} mb-5" {% endif %}>{{message}}</p>
        {% endfor %}
        {% endif %}
        {% csrf_token %}
        {% for fm in form %}
          <div class="form-group mb-3">
            {{fm.label_tag}} {{fm}} <small class="text-danger">{{fm.errors|striptags}}</small>
          </div>
        {% endfor %}
        <input type="submit" value="Submit" class="btn btn-primary">
        <br>
        <div class="text-center text-primary fw-bold"><small>Existing User ? <a href="{% url 'login' %}" class="text-danger">Login Now</a> </small></div>
        {% if form.non_field_errors %}
        {% for error in form.non_field_errors %}
          <p class="alert alert-danger my-3">{{error}}</p>
        {% endfor %}
        {% endif %}
      </form>
    </div>
  </div>
</div>
{% endblock main-content %}
```

### *emptycart.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Empty Cart{% endblock title %}
4 {% block main-content %}
5   <div class="container my-5">
6     <div class="row">
7       <h3 class="text-center">Ayup! Seems like your cart is empty. Try filling it maybe! :D</h3>
8       <div class="text-center">
9         
10      </div>
11    </div>
12  </div>
13 {% endblock main-content %}
```

## home.html

```
{% extends 'app/base.html' %}
{% load static %}
{% block title %}Home{% endblock title %}

{% block banner_slider %}
<!-- Banner Slider -->
<div id="carouselExampleControls" class="carousel slide" data-bs-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
  </a>
</div>
<!-- End Banner Slider -->
{% endblock banner_slider %}

{% block livesale %}
<!-- Live Sale Section -->
<div class="container">
  <div class="row bg-danger text-center p-5 text-white border-bottom shadow">
    <div>SALE IS LIVE NOW</div>
    <span>5% Instant Discount for no one</span>
    <small class="fw-lighter">Terms and Conditions Applied (For details visit Bank's fake Website)</small>
  </div>
</div>
<!-- End Live Sale Section -->
{% endblock livesale %}

{% block main-content %}
<!-- 1st Product Slider -->
<div class="m-3">
  <h2>Shop Bottom Wear</h2>
  <!-- Slider 1 -->
  <div class="owl-carousel" id="slider1">
    {%- for b in bottomwears %}
    <a href="{% url 'product-detail' b.id %}" class="btn"><div class="item"><span class="fw-bold">{{b.title}}</span><br><span class="fs-5">Rs. {{b.discounted_price}}</span></div></a>
    {%- endfor %}
  </div>
</div>
<!-- End 1st Product Slider -->

<!-- Payment Info Section -->
<div class="container my-5">
  <div class="row">
    <div class="col-sm-3">
      <div class="card mb-3">
        <div class="card-body">
          
        </div>
      </div>
    </div>
    <div class="col-sm-3">
      <div class="card mb-3">
        <div class="card-body">
          
        </div>
      </div>
    </div>
    <div class="col-sm-3">
      <div class="card mb-3">
        <div class="card-body">
          
        </div>
      </div>
    </div>
    <div class="col-sm-3">
      <div class="card mb-3">
        <div class="card-body">
          
        </div>
      </div>
    </div>
  </div>
</div>
<!-- End Payment Info Section -->

<!-- 2nd Product Slider -->
<div class="m-3">
  <h2>Shop Top Wear</h2>
  <!-- Slider 2 -->
  <div class="owl-carousel" id="slider2">
    {%- for tp in topwears %}
    <a href="{% url 'product-detail' tp.id %}" class="btn"><div class="item"><span class="fw-bold">{{tp.title}}</span><br><span class="fs-5">Rs. {{tp.discounted_price}}</span></div></a>
    {%- endfor %}
  </div>
</div>
<!-- End 2nd Product Slider -->

<!-- 3rd Product Slider -->
<div class="m-3">
  <h2>Shop Mobiles</h2>
  <!-- Slider 3 -->
  <div class="owl-carousel" id="slider3">
    {%- for m in mobiles %}
    <a href="{% url 'product-detail' m.id %}" class="btn"><div class="item"><span class="fw-bold">{{m.title}}</span><br><span class="fs-5">Rs. {{m.discounted_price}}</span></div></a>
    {%- endfor %}
  </div>
</div>
<!-- End 3rd Product Slider -->
{% endblock main-content %}
```

### *laptop.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Laptop{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   <div class="row">
7     <div class="col-sm-3">
8       <div class="list-group">
9         <a href="{% url 'laptop' %}" class="list-group-item list-group-item-action" aria-current="true">All laptops</a>
10       </div>
11       <h3> Coming Soon! </h3>
12     </div>
13     <div class="col-sm-8">
14       <div class="row">
15         {% for product in laptops %}
16         <div class="col-sm-4 text-center mb-4">
17           <a href="{% url 'product-detail' product.id %}" class="btn">
18             <div class="item">
19               
20               <div class="fw-bold">{{product.title}}</div>
21               <div class="fw-bold">Rs. {{product.discounted_price}} <small class="fw-light text-decoration-line-through">
                {{product.selling_price}}</small></div>
22             </div>
23           </a>
24         </div>
25         {% endfor %}
26       </div>
27     </div>
28   </div>
29 </div>
30 {% endblock main-content %}
```

### *mobile.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Mobile{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   <div class="row">
7     <div class="col-sm-3">
8       <div class="list-group">
9         <a href="{% url 'mobile' %}" class="list-group-item list-group-item-action" aria-current="true">All Mobiles</a>
10         <a href="{% url 'mobiledata' 'Redmi' %}" class="list-group-item list-group-item-action" aria-current="true">Redmi
11         </a>
12         <a href="{% url 'mobiledata' 'Samsung' %}" class="list-group-item list-group-item-action">Samsung</a>
13         <a href="{% url 'mobiledata' 'below' %}" class="list-group-item list-group-item-action">Below 10000</a>
14         <a href="{% url 'mobiledata' 'above' %}" class="list-group-item list-group-item-action">Above 10000</a>
15       </div>
16     </div>
17     <div class="col-sm-8">
18       <div class="row">
19         {% for product in mobiles %}
20         <div class="col-sm-4 text-center mb-4">
21           <a href="{% url 'product-detail' product.id %}" class="btn">
22             <div class="item">
23               
24               <div class="fw-bold">{{product.title}}</div>
25               <div class="fw-bold">Rs. {{product.discounted_price}} <small class="fw-light text-decoration-line-through">
                {{product.selling_price}}</small></div>
26             </div>
27           </a>
28         </div>
29         {% endfor %}
30       </div>
31     </div>
32   </div>
33 </div>
34 {% endblock main-content %}
```

*orders.html*

```

{% extends 'app/base.html' %}
{% load static %}
{% block title %}Orders{% endblock title %}
{% block main-content %}
<div class="container my-5">
  <div class="row">
    <h3>Welcome</h3>
    <div class="col-sm-2 border-end">
      <ul class="list-unstyled">
        <li class="d-grid"><a href="{% url 'orders' %}" class="btn btn-primary">
          Orders</a></li>
      </ul>
    </div>
    <div class="col-sm-9 offset-sm-1">
      <div class="row">
        {% for op in order_placed %}
          <div class="col-sm-12">
            <div class="row shadow-sm mb-3">
              <div class="col-sm-2"></div>
              <div class="col-sm-7">
                <p>Product: {{op.product.title}}</p>
                <p>Quantity: {{op.quantity}}</p>
                <p>Price: {{op.total_cost}}</p>
              </div>
              <div class="col-sm-3 fw-bold">
                <p>Order Status: {{op.status}}</p>
                {% if op.status == 'Accepted' %}
                  <div class="progress">
                    <div class="progress-bar" role="progressbar" style="width: 20%
                      aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
                  </div>
                {% endif %}

                {% if op.status == 'Packed' %}
                  <div class="progress">
                    <div class="progress-bar bg-info" role="progressbar" style="width: 40
                      " aria-valuenow="40" aria-valuemin="0" aria-valuemax="100"></div>
                  </div>
                {% endif %}

                {% if op.status == 'On The Way' %}
                  <div class="progress">
                    <div class="progress-bar bg-warning" role="progressbar" style="width:
                      70% aria-valuenow="70" aria-valuemin="0" aria-valuemax="100"></
                      div>
                  </div>
                {% endif %}

                {% if op.status == 'Delivered' %}
                  <div class="progress">
                    <div class="progress-bar bg-success" role="progressbar" style="width:
                      100% aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"><
                      div>
                  </div>
                {% endif %}
              </div>
            </div>
          </div>
        {% endfor %}
      </div>
    </div>
  </div>
</div>
{% endblock main-content %}

```

### *password\_reset.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Reset Password{% endblock title %}
4 {% block main-content %}
5 <div class="container">
6   <div class="row my-3">
7     <div class="col-sm-6 offset-sm-3">
8       <h3>Reset Password</h3>
9       <hr>
10      <form action="" method="post" novalidate class="shadow p-5">
11        {% csrf_token %}
12        {% for fm in form %}
13          <div class="form-group">
14            {{fm.label_tag}} {{fm}} <small class="text-danger">{{fm.errors|striptags}}</small> <br>
15          </div>
16        {% endfor %}
17        <input type="submit" class="btn btn-primary mt-3" value="Submit">
18        {% if form.non_field_errors %}
19          {% for error in form.non_field_errors %}
20            <p class="alert alert-danger my-3">{{error}}</p>
21          {% endfor %}
22        {% endif %}
23      </form>
24    </div>
25  </div>
26 </div>
27 {% endblock main-content %}
```

### *password\_reset\_complete.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Done!{% endblock title %}
4 {% block main-content %}
5 <div class="container">
6   <div class="row my-3">
7     <div class="col-sm-6 offset-sm-3">
8       <h4>Password Reset Successful!</h4>
9       <div><iframe src="https://giphy.com/embed/ibolLe3mOqHE3PQTtk" width="200" height="200" frameBorder
10         ="0" class="giphy-embed" allowFullScreen></iframe></div>
11     </div>
12   </div>
13 </div>
14 {% endblock main-content %}
```

### *password\_reset\_confirm.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Password Reset Confirm{% endblock title %}
4 {% block main-content %}
5 <div class="container">
6   <div class="row my-3">
7     <div class="col-sm-6 offset-sm-3">
8       <h3>Password Reset Confirm</h3>
9       <hr>
10      <form action="" method="post" novalidate class="shadow p-5">
11        {% csrf token %}
12        {% for fm in form %}
13          <div class="form-group">
14            {{fm.label_tag}} {{fm}} <small class="text-danger">{{fm.errors|striptags}}</small> <br>
15          </div>
16        {% endfor %}
17        <input type="submit" class="btn btn-primary mt-3" value="Submit">
18        {% if form.non_field_errors %}
19          {% for error in form.non_field_errors %}
20            <p class="alert alert-danger my-3">{{error}}</p>
21          {% endfor %}
22        {% endif %}
23      </form>
24    </div>
25  </div>
26 </div>
27 {% endblock main-content %}
```

### *password\_reset\_done.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Password Reset Done{% endblock title %}
4 {% block main-content %}
5 <div class="container">
6   <div class="row my-3">
7     <div class="col-sm-6 offset-sm-3">
8       <h4 class="alert-heading">Well done!</h4>
9       <p>An Email has been sent with instructions to reset your password. <b>Check Your Email!</b></p>
10     </div>
11   </div>
12 </div>
13 </div>
14 {% endblock main-content %}
```

### *passwordchange.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Change Password{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   <div class="row">
7     <h3>Welcome</h3>
8     <div class="col-sm-2 border-end">
9       <ul class="list-unstyled">
10        <li class="d-grid"><a href="{% url 'profile' %}" class="btn btn-primary">Change Password</a></li>
11      </ul>
12    </div>
13    <div class="col-sm-6 offset-sm-1">
14      <h3>Change Password</h3>
15      <hr>
16      <form action="" method="post" novalidate class="shadow-sm p-5">
17        {% csrf_token %}
18        {% for fm in form %}
19          <div class="form-group">
20            {{fm.label_tag}} {{fm}} <small class="text-danger">{{fm.errors|striptags}}</small> <br>
21          </div>
22        {% endfor %}
23        <input type="submit" class="btn btn-primary mt-4" value="Save">
24        {% if form.non_field_errors %}
25          {% for error in form.non_field_errors %}
26            <p class="alert alert-danger my-3">{{error}}</p>
27          {% endfor %}
28        {% endif %}
29      </form>
30    </div>
31  </div>
32 </div>
33 {% endblock main-content %}
```

### *passwordchangedone.html*

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Done!{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   <div class="row">
7     <h3>Welcome</h3>
8     <div class="col-sm-2 border-end">
9       <ul class="list-unstyled">
10        <li class="d-grid"><a href="{% url 'profile' %}" class="btn btn-primary">Changed the
          password successfully!</a></li>
11      </ul>
12    </div>
13    <div class="col-sm-9 offset-sm-1 text-center">
14      <h3>Password Changed Successfully !!</h3>
15      <div><iframe src="https://giphy.com/embed/ibolLe3mOqHE3PQTtk" width="200" height="200"
16        " frameborder="0" class="giphy-embed" allowFullScreen></iframe></div>
17    </div>
18  </div>
19 {% endblock main-content %}
```



*productdetail.html*

```

{% extends 'app/base.html' %}
{% load static %}
{% block title %}Product Detail{% endblock title %}
{% block main-content %}
<div class="container my-5">
  <div class="row">
    <div class="col-sm-6 text-center align-self-center">
      
    </div>
    <div class="col-sm-5 offset-sm-1">
      <h2>{{product.title}}</h2>
      <hr>
      <p>{{product.description}}</p> <br>
      <h4>Rs. {{product.discounted_price}} <small class="fw-light
      text-decoration-line-through">{{product.selling_price}}</small></h4> <br>

      {% if user.is_authenticated %}
      {% if item_already_in_cart %}
        <a href="{% url 'showcart' %}" class="btn btn-warning shadow px-5 py-2 ms-4
        ">Go to Cart</a>
      {% else %}
        <form action="/add-to-cart" class="d-inline">
          <input type="hidden" name="prod_id" value="{{product.id}}" id="prod_id">
          <button type="submit" class="btn btn-primary shadow px-5 py-2">Add to Cart
          </button>
        </form>
      {% endif %}
      {% endif %}

      {% if not request.user.is_authenticated %}
        <form action="/add-to-cart" class="d-inline">
          <input type="hidden" name="prod_id" value="{{product.id}}" id="prod_id">
          <button type="submit" class="btn btn-primary shadow px-5 py-2">Add to Cart
          </button>
        </form>
      {% endif %}

      <a href="{% url 'checkout' %}" class="btn btn-danger shadow px-5 py-2 ms-4">
      Buy Now</a>
      <h5 class="mt-5">Available Offers</h5>
      <ul>
        <li>Bank Offer 5% Unlimited Cashback on Flipkart Axis Bank Credit</li>
        <li>Special Price Get extra ₹3000 off (price inclusive of discount)</li>
        <li>No cost EMI ₹1,667/month. Standard EMI also available</li>
        <li>Partner Offer ₹2000 Flipkart Gift Card on Every 1000th Transaction with
        a new Visa Debit/Credit Card</li>
      </ul>
    </div>
  </div>
</div>
{% endblock main-content %}

```

## profile.html

```
1 {% extends 'app/base.html' %}
2 {% load static %}
3 {% block title %}Profile{% endblock title %}
4 {% block main-content %}
5 <div class="container my-5">
6   <div class="row">
7     <h3>Welcome <span class="text-capitalize">{{request.user}}</span> </h3>
8     <div class="col-sm-2 border-end">
9       <ul class="list-unstyled">
10        <li class="d-grid"><a href="{% url 'profile' %}" class="btn {{active}}">Profile</a></li>
11        <li class="d-grid"><a href="{% url 'address' %}" class="btn">Address</a></li>
12      </ul>
13    </div>
14    <div class="col-sm-8 offset-sm-1">
15      <form action="" method="post" novalidate>
16        {{ csrf_token }}
17        {% for fm in form %}
18          <div class="form-group col-lg-6 mb-3">
19            {{fm.label_tag}} {{fm}} <small class="text-danger">{{fm.errors|striptags}}</small>
20          </div>
21          {% endfor %}
22          <div class="col-12 mt-3">
23            <button type="submit" class="btn btn-primary mb-3">Submit</button>
24          </div>
25        </form>
26        {% if messages %}
27          {% for message in messages %}
28            <p {{ if message.tags }} class="alert alert-{{message.tags}} mb-5" {{ endif }}>{{message}}</p>
29          {% endfor %}
30        {% endif %}
31      </div>
32    </div>
33  </div>
34 {% endblock main-content %}
```

## Step 5 | Adding css, js and image files

Now that you've created the necessary front-end websites, its time to import css, js and image files

I highly recommend you to get all of them from my [GitHub](#).

## Step 6 | Running the project

\*sigh\* Finally! The time has come, to execute our project. I hope you've followed the steps I've mentioned here and now you are ready to roll!

So, first of, open a cmd window at your manage.py location and type **py manage.py migrate**

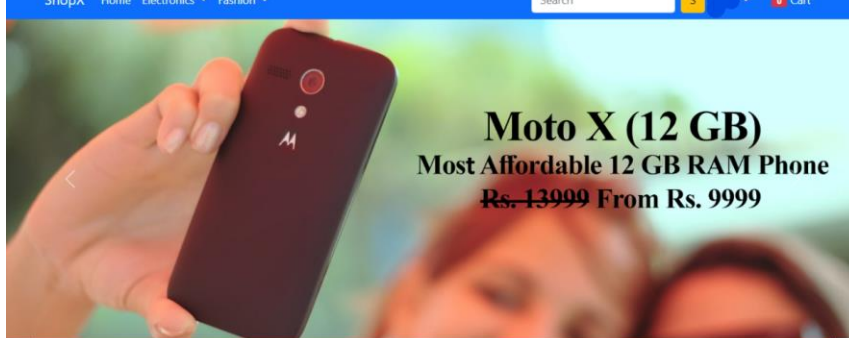
Hit enter and then type **py manage.py createsuperuser**

This will open a CLI Interface to create an admin account. Enter your details and then, type **py manage.py runserver**.

BOOM!

You should see a website like this

ShopX
Home
Electronics
Fashion
Search
S
Cart



**Moto X (12 GB)**  
Most Affordable 12 GB RAM Phone  
~~Rs. 13999~~ From Rs. 9999

**SALE IS LIVE NOW**  
5% Instant Discount for no one  
Terms and Conditions Applied (For details visit Bank's fake Website)

### ShopX Bottom Wear



**Jeans 2**  
Rs. 360.0



**Jeans 3**  
Rs. 550.0



**Jeans 4**  
Rs. 800.0



**Jeans 5**  
Rs. 450.0



**Jeans 6**  
Rs. 305.0






### ShopX Top Wear



**Tshirt 2**  
Rs. 300.0



**Tshirt 3**  
Rs. 450.0



**T Shirt 4**  
Rs. 700.0



**Tshirt 5**  
Rs. 250.0



**Tshirt 6**  
Rs. 150.0

### ShopX Mobile



**Redmi Pro 3**  
Rs. 4999.0



**Samsung NT 2**  
Rs. 10999.0



**Samsung Pro 4**  
Rs. 3999.0



**Redmi 7**  
Rs. 2999.0

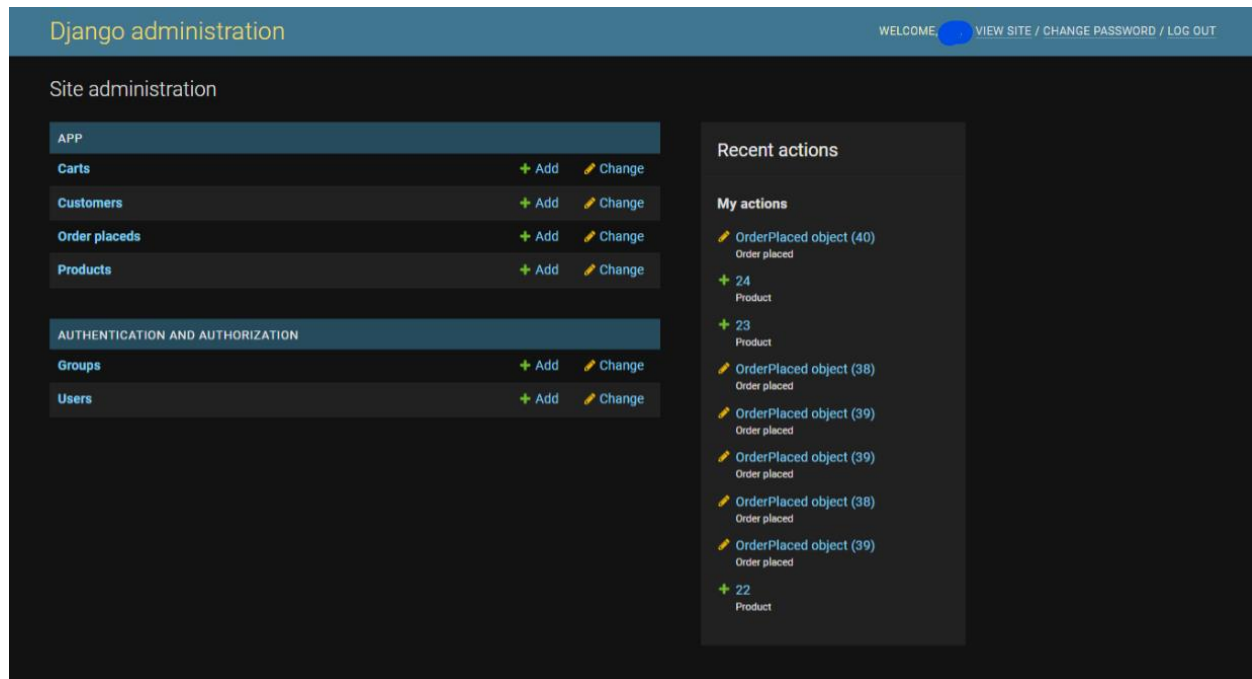


**Redmi 9**  
Rs. 1999.0

## Step 7 | Accessing the admin panel

Now as you have all the stuff loaded up and running, its time to finally add some items to your website, right? So, head over to [localhost:8000/admin](http://localhost:8000/admin).

This will take you to an interface similar to this



Here you can click **Products** and it will take you to a product addition wizard! Pretty neat huh!

### Troubleshooting

Any database errors?

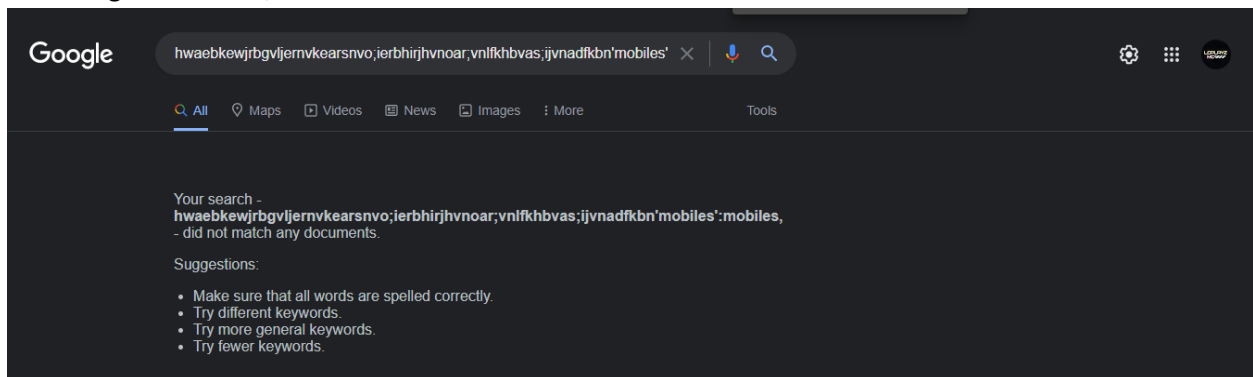
- I recommend you to take the **db.sqlite3** file from my [GitHub](#), and try replacing it with original one if that works. It will also provide you with some default products.

### Note

Be sure to grab **media** folder from my [GitHub](#) as it contains all the image files and paste it in the main directory.

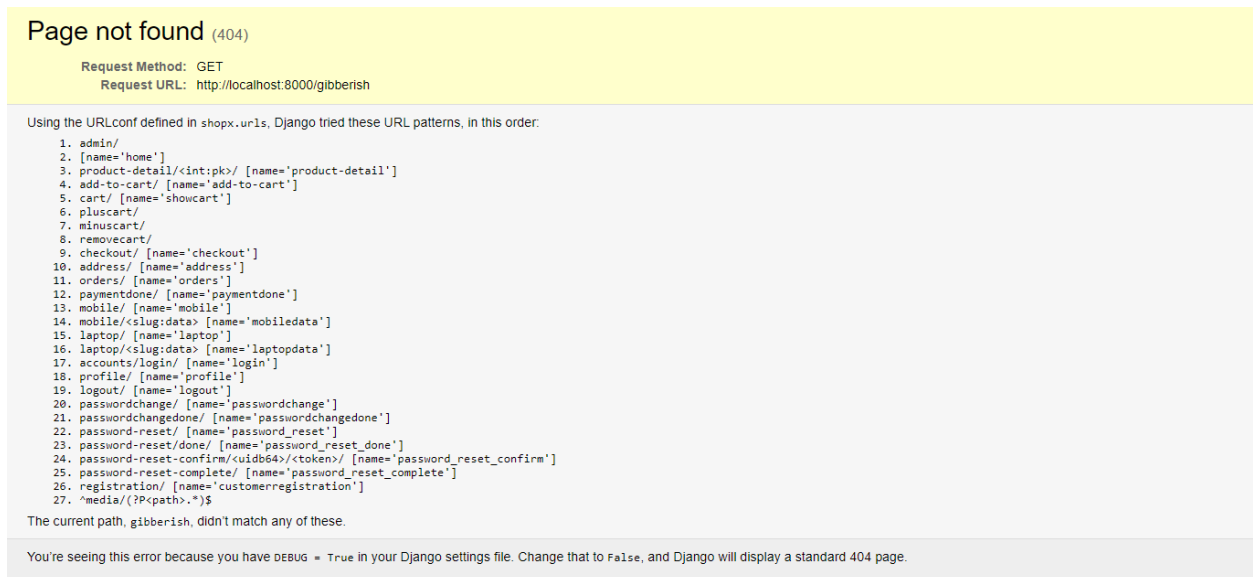
## Bonus | Adding 404 Error Page

Every website isn't perfect nor complete. Let's take an example. On [Google](#), search some random gibberish like,



Yea? See that page? Here Google returned with *Your search did not match any documents.*

You don't want customers to go anywhere and see this,



We need to make it a subtle error page with a button to take those lost users to the home page right?

We are going to implement just that right now.

See when you execute the website right now, it will show an error. That's because we have some stuff related to 404 page already done! Now its time to finish it.

## Adding 404 Page

You might remember that we have added line 9 at <project>'s urls.py. [Reference](#)

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('', include('app.urls')),
7 ]
8
9 handler404 = "shopx.views.page_not_found_view"
```

This line was for 404 page. Now the **handler404** you might see is a default **Django** term. And **shopx.views.page\_not\_found\_view** refers a view in <project>'s views.py . But now you might think, why should we add views.py in <project>'s folder? Isn't it supposed that we should add it in app folder? Well, you are right, but here we want to address each and every link which is not defined, such as <http://localhost:8000/gibberish> and all, to redirect to a specific page which will have a button right? So basically, Think it this way...

<project> is the *parent* of **app** folder, so, any invalid links should redirect to the parent so parent handles it.

So first, let's create a file named **views.py** in <project> folder

Your directory should look something like this,

```
.
├── <project_name>
│   └── <project>
│       ├── __pycache__
│       ├── __init__.py
│       ├── asgi.py
│       ├── settings.py
│       └── urls.py
```



```
└─ views.py <- Create this file
└─ wsgi.py
```

Now in this file, add this bit of code,

```
1 from django.shortcuts import render
2
3 def page_not_found_view(request, exception):
4     return render(request, '404.html', status=404)
```

Now you might notice that we have called 404.html file, now we are going to make that file.

Create a new folder named **templates** and create a new file named **404.html** in the <project\_name> folder. Your directory should look like this,

```
.
└─ <project_name>
    └─ <project>
        └─ app
            └─ templates <- Create this folder
                └─ 404.html
```

Now in **404.html**, we are going to add this bit of code,

Note that you can add any kind of code here so it isn't necessary to use just my code but, well ... here I go anyways :D

```
1 <html>
2 <head>
3     <link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
4     <script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
5     <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
6 </head>
7 <!-- Include the above in your HEAD tag -->
8
9 <div class="container">
10     <div class="row">
11         <div class="col-md-12">
12             <div class="error-template">
13                 <h1>
14                     Oops!</h1>
15                 <h2>
16                     404 Not Found</h2>
17                 <div class="error-details">
18                     Sorry, an error has occurred, Requested page not found!
19                 </div>
20                 <div class="error-actions">
21                     <a href="/" class="btn btn-primary btn-lg"><span class="glyphicon glyphicon-home"></span>
22                         Take Me Home </a>
23                 </div>
24             </div>
25         </div>
26     </div>
27 </div>
28 </html>
```

Now that the 404.html is ready, let's also make some changes to **settings.py**. Django itself recommends to turn off **DEBUG=TRUE** in settings.py when in production.

## Page not found (404)

Request Method: GET  
Request URL: http://localhost:8000/gibberish

Using the URLconf defined in shopx.urls, Django tried these URL patterns, in this order:

```
1. admin/
2. [name='home']
3. product-detail/<int:pk>/ [name='product-detail']
4. add-to-cart/ [name='add-to-cart']
5. cart/ [name='showcart']
6. pluscart/
7. minuscart/
8. removecart/
9. checkout/ [name='checkout']
10. address/ [name='address']
11. orders/ [name='orders']
12. paymentdone/ [name='paymentdone']
13. mobile/ [name='mobile']
14. mobile/<slug:data> [name='mobiledata']
15. laptop/ [name='laptop']
16. laptop/<slug:data> [name='laptopdata']
17. accounts/login/ [name='login']
18. profile/ [name='profile']
19. logout/ [name='logout']
20. passwordchange/ [name='passwordchange']
21. passwordchangedone/ [name='passwordchangedone']
22. password-reset/ [name='password_reset']
23. password-reset/done/ [name='password_reset_done']
24. password-reset-confirm/<uidb64>/<token>/ [name='password_reset_confirm']
25. password-reset-complete/ [name='password_reset_complete']
26. registration/ [name='customerregistration']
27. ^media/(?P<path>.*)$
```

The current path, gibberish, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

Here if you take a look at the footer, you might see:

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page

We want to change this,

Head over to **settings.py** and change these lines

```
26 DEBUG = False
27
28 ALLOWED_HOSTS = ['.localhost', '127.0.0.1', '[:,1]']
```

Note that you can change **ALLOWED\_HOSTS** to your domain, for example you've set your home router, and in the router settings you've set a domain, now you want your app to use that domain, then you should set it here as a string and voila! Nevermind... so then you want to set **DEBUG = False**. Now when you visit any invalid link, it will redirect you to this page.

# Oops!

## 404 Not Found

Sorry, an error has occurred. Requested page not found!

[🏠 Take Me Home](#)

So that's it! Now your lost customers will find a way to home! Hurray!

**So that's it for this documentation, I hope you liked it! If you find any errors or doubts, be sure to visit my GitHub's issues page and file an issue there. I will do my best to answer those.**

## References

[GitHub](#)

[Django](#)

[Project Provider](#)

[Author](#)