**Overview**

ECommerce shopping website is a panel template built with HTML. Website goes beyond usual templates and provides you with an intuitive programming framework. Server Side Rendering and Django backend will even further speed up development. We can use the ECom web app to build any type of web application Full Product management tools(ordering to placing it), etc.

### 

**Quick start and installation steps**

#clone the repo

* Git clone <https://github.com/chintantutlani/ECommerce-Shopping-Product.git>

#install requirements

* pip install -r .\requirements.txt

#setup mysql database and migrations

* Python manage.py makemigrations
* Python manage.py migrate

#runserverr

* Python manage.py runserver

**Setting Django Project Structure.**

* Create a Project called ECOMWEB.(django-admin startproject ECOMWEB)
* Setting urls for my app in ECOMWEB.
  + path(‘ ‘, include(product.urls)),
* For setting up the image of the product.
  + create a static folder and configure in settings.py
    - Configurations
      * MEDIA\_URL = '/images//'
      * MEDIA\_ROOT = os.path.join(BASE\_DIR, 'static/images')
      * Then we have to mention it in urls as well.
      * urlpatterns=[]+static(settings.MEDIA\_URL,document\_root=settings.MEDIA\_ROOT)

(That's how we store media files,the rest all will be handled by django itself.)

**Setting Django App Structure.**

* Created an App called Product.
* There are my py file
  + Urls , Views , Models, Forms, Fixture
    - Urls

#used to fetch all the products and when we initially start the server.

* + - * path('', views.ShowAllProducts, name='showProducts'),

#used to fetch the particular details of products with the id no.

* + - * path('product/<int:pk>/', views.productDetail, name='product'),

#used to add new products .

* + - * path('addProduct/', views.addProduct, name='addProduct'),

#used to update the particular details of products with the id no.

* + - * path('updateProduct/<int:pk>/', views.updateProduct, name='updateProduct'),

#used to delete the particular details of products with the id no.

* + - * path('deleteProduct/<int:pk>/', views.deleteProduct, name='deleteProduct')
* Views
  + ShowAllProducts,productDetail ,addProduct ,updateProduct ,deleteProduct
    - ShowAllProducts
      * This is used to fetch all the product
        + If the category is not selected it shows all the product in order which is k/last product created and it has pagination in which if the product on the page is more than the set number it will create the next page and likewise.
        + If the category is selected it will filter and show products according to the category selected.
    - productDetail
      * This view fetch all the details of the particular product

With id which we want to see

* addProduct
  + This view is used to add Product
    - When we hit this view it show the form where we can add your details of our product
      * It check all the details and file if it valid it save the data and show our product
* updateProduct
  + This view is used to update particular product
    - When we hit this view it show the details of the product and where we can update details of our product
* deleteProduct
  + This view is used to delete particular product
    - When we hit this view it shows the product which we want to delete
* Models
  + Tales are created in ORM
  + Two tables
  + Product ,Category
    - Product
      * Consisted of field which are used to display image info about what are product consist of
      * And taken category field as foreignKey as a product can have different category can have different product
    - Category
      * It consists of name of the category and foreignkey of product
  + Forms
    - It is built in class which takes model and fields name and Return us a Form Consisting fields which were provided
    - Fixtures
      * I have created a collection of data that Django knows how to import into a database.