
Proyecto: Boutique

Lenguaje: Python

Framework: Django

Editor: VS code

1 Procedimiento para crear carpeta del Proyecto: UIII_Boutique_0777

2 procedimiento para abrir vs code sobre la carpeta UIII_Boutique_0777

3 procedimiento para abrir terminal en vs code

4 Procedimiento para crear carpeta entorno virtual “.venv” desde terminal de vs code

5 Procedimiento para activar el entorno virtual.

6 procedimiento para activar intérprete de python.

7 Procedimiento para instalar Django

8 procedimiento para crear proyecto backend_Boutique sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8390

10 procedimiento para copiar y pegar el link en el navegador.

11 procedimiento para crear aplicación app_Boutique

12 Aquí el modelo models.py

```
===== ===== ===== ===== ===== ===== =====  
from django.db import models  
  
===== ===== ===== ===== ===== ===== =====
```

MODELO: CATEGORÍA

```
===== ===== ===== ===== ===== ===== =====
```

```
class Categoria(models.Model):  
    idcategoria = models.AutoField(primary_key=True, unique=True)  
    nombre = models.CharField(max_length=45)  
    descripcion = models.TextField()  
    fechacreacion = models.DateField()  
    activo = models.BooleanField(default=True)  
    tipo = models.CharField(max_length=20)  
    slug = models.SlugField(max_length=100, unique=True)  
    def str(self):  
        return self.nombre  
  
===== ===== ===== ===== ===== ===== =====
```

MODELO: PRODUCTO

```
=====
class Producto(models.Model):
    idproducto = models.AutoField(primary_key=True, unique=True)
    nombre = models.CharField(max_length=100)
    descripcion = models.TextField()
    precio = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.IntegerField()
    fechaagregado = models.DateTimeField(auto_now_add=True)
    imagen = models.CharField(max_length=255, blank=True, null=True)
    idcategoria2 = models.ForeignKey(
        Categoria,
        on_delete=models.CASCADE,
        db_column='idcategoria2',
        related_name='productos'
    )
    def str(self):
        return self.nombre
=====
```

MODELO: PROVEEDOR

```
=====
class Proveedor(models.Model):
    idproveedor = models.AutoField(primary_key=True, unique=True)
    nombre = models.CharField(max_length=100)
    email = models.EmailField(max_length=100)
    telefono = models.CharField(max_length=20)
    direccion = models.CharField(max_length=150)
    ciudad = models.CharField(max_length=100)
    pais = models.CharField(max_length=100)
    id_producto2 = models.ForeignKey(
        Producto,
        on_delete=models.CASCADE,
        db_column='id_producto2',
        related_name='proveedores'
    )
    def str(self):
        return self.nombre
=====
```

12.5 Procedimiento para realizar las migraciones(makemigrations y migrate).

13 primero trabajamos con el MODELO: CATEGORÍA

14 En view de app_Boutique crear las funciones con sus códigos correspondientes
(inicio_boutique, agregar_categoria, actualizar_categoria, realizar_actualizacion_categoria, borrar_categoria)

15 Crear la carpeta “templates” dentro de “app_Boutique”.

16 En la carpeta templates crear los archivos html (base.html, header.html, navbar.html, footer.html, inicio.html).

17 En el archivo base.html agregar bootstrap para css y js.

18 En el archivo navbar.html incluir las opciones (“Sistema de Administración Boutique”, “Inicio”, “categoria”, en submenu de categorias(Agregar Categoria, ver categoria, actualizar categoria, borrar categoria), “Productos” en submenu de Productos(Agregar producto, ver productos, actualizar productos, borrar productos), “Proveedores” en submenu de Proveedores(Agregar proveedor, ver proveedores, actualizar proveedores, borrar proveedores), incluir iconos a las opciones principales, no en los submenu.

19 En el archivo footer.html incluir derechos de autor, fecha del sistema y “Creado por Sofia Vazquez, Cbtis 128” y mantenerla fija al final de la página.

20 En el archivo inicio.html se usa para colocar información del sistema más una imagen tomada desde la red sobre boutique.

21 Crear la subcarpeta carpeta categoria dentro de app_Boutique\templates.

22 crear los archivos html con su codigo correspondientes de (agregar_categoria.html, ver_categorias.html mostrar en tabla con los botones ver, editar y borrar, actualizar_categoria.html, borrar_categoria.html) dentro de app_Boutique\templates\categoria.

23 No utilizar forms.py.

24 procedimiento para crear el archivo urls.py en app_Boutique con el código correspondiente para acceder a las funciones de views.py para operaciones de crud en categorias.

25 procedimiento para agregar app_Boutique en settings.py de backend_Boutique

26 realizar las configuraciones correspondiente a urls.py de backend_Boutique para enlazar con app_Boutique

27 procedimiento para registrar los modelos en admin.py y volver a realizar las migraciones.

27 por lo pronto solo trabajar con “categoría” dejar pendiente # MODELO: PRODUCTO y # MODELO: PROVEEDOR

28 Utilizar colores suaves, atractivos y modernos, el código de las páginas web sencillas.

28 No validar entrada de datos.

29 Al inicio crear la estructura completa de carpetas y archivos.

30 proyecto totalmente funcional.

31 finalmente ejecutar servidor en el puerto puerto 8390.