

Primera parte

Proyecto: Shein.

Lenguaje: Python.

Framework: Django.

Editor: VS code.

1 Procedimiento para crear carpeta del Proyecto: UIII_Shein_0630

2 procedimiento para abrir vs code sobre la carpeta UIII_Shein_0630

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_Shein sin duplicar carpeta.

9 procedimiento para ejecutar servidor en el puerto 8036

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

11 procedimiento para crear aplicación app_Shein

12 Aquí el modelo models.py

=====

=====

=====

MODELO: USUARIOS (Basado en la imagen proporcionada)

=====

=====

```
class Usuarios(models.Model):
    nombre = models.CharField(max_length=255)
    Apellido = models.CharField(max_length=255)
    Correo_Electronico = models.CharField(max_length=255, unique=True)
    Numero_Telefonico = models.CharField(max_length=20)
    id_usuario = models.IntegerField(primary_key=True) # Asumiendo PK aquí
```

MODELO: PRODUCTOS (Basado en la imagen proporcionada)

```
class Productos(models.Model):
    producto_id = models.IntegerField(primary_key=True)
    nombre_producto = models.CharField(max_length=255)
    descripcion = models.TextField(blank=True, null=True)
    precio = models.FloatField()
    cantidad = models.IntegerField()
```

MODELO: PEDIDOS (Basado en la imagen proporcionada)

```
class Pedidos(models.Model):
    id_pedido = models.IntegerField(primary_key=True)
    Estado_pedido = models.CharField(max_length=50)
    direccion = models.CharField(max_length=255)
    fecha_pedido = models.DateTimeField()
    id_usuario = models.IntegerField() # FK a Usuarios
    id_producto = models.IntegerField() # FK a Productos
```

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

13 primero trabajamos con el MODELO: USUARIOS

14 En view de app_Shein crear las funciones con sus códigos correspondientes

(inicio_cinepolis, agregar_usuario, actualizar_usuario, realizar_actualizacion_usuario, borrar_usuario)

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

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 Shein”, “Inicio”, “Usuarios”, en submenu de Usuarios(Agregar Usuario, Ver Usuarios, Actualizar Usuario, Borrar Usuario), “Productos” en submenu de Productos(Agregar Producto, Ver Productos, Actualizar Productos, Borrar Productos), “Pedidos” en submenu de Pedidos(Agregar Pedido, Ver Pedidos, Actualizar Pedidos, Borrar Pedidos), 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 Ing. Cesar Loya, 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 Shein.

21 Crear la subcarpeta carpeta usuarios dentro de app_Shein\templates.

22 crear los archivos html con su codigo correspondientes de (agregar_usuario.html, ver_usuarios.html mostrar en tabla con los botones ver, editar y borrar, actualizar_usuario.html, borrar_usuario.html) dentro de app_Shein\templates\usuarios.

23 No utilizar forms.py.

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

25 procedimiento para agregar app_Shein en settings.py de backend_Shein

26 realizar las configuraciones correspondiente a urls.py de backend_Shein para enlazar con app_Shein

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

27 por lo pronto solo trabajar con “usuarios” dejar pendiente # MODELO: PRODUCTOS y # MODELO: PEDIDOS

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 8030.
