Informe Primer Trabajo Práctico
Web II 2024

Alumno: Lucas Moraga
Profesor: Lic. Julio César Casco

Descripción del Proyecto

El proyecto consiste en el desarrollo de una aplicación web para la gestión de mensajes utilizando el framework Django. La aplicación permite a los usuarios enviar y recibir mensajes, así como buscar mensajes por destinatario.

Explicación del Modelo Creado

El modelo principal de la aplicación es el modelo `Mensaje`, que contiene los siguientes campos:
-remitente: El usuario que envía el mensaje.
-destinatario: El usuario que recibe el mensaje.
-texto: El contenido del mensaje
-fecha_envio: La fecha y hora en que se envió el mensaje.

Proceso de Implementación de la Vista y Cómo se Vinculó a la URL

Se creó una vista para mostrar los mensajes recibidos por el usuario. La vista se implementó en el archivo views.py y se vinculó a la URL correspondiente en el archivo urls.py. La vista utiliza el método GET para recibir parámetros de búsqueda y filtrar los mensajes por destinatario.

Descripción de la Plantilla HTML Utilizada

La plantilla HTML utilizada para mostrar los mensajes recibidos (mensajes_recibidos.html) incluye un encabezado con el título "Mensajes Recibidos", un formulario de búsqueda y una lista de mensajes. Cada mensaje se muestra en un recuadro blanco con detalles como la fecha de envío, remitente, destinatario y el contenido del mensaje.

Explicación sobre la Configuración del Entorno Virtual y la Creación del Archivo requirements.txt

Se configuró un entorno virtual utilizando venv para gestionar las dependencias del proyecto. El archivo requirements.txt se generó utilizando el comando pip freeze > requirements.txt, que lista todas las dependencias instaladas en el entorno virtual.

Detalle del Script de Datos de Prueba

SCRIPT:
Se guardó en el archivo "datosPrueba.py" en la carpeta raiz del proyecto.

Se creó un script para generar datos de prueba y poblar la base de datos con mensajes ficticios. El script utiliza el ORM de Django para crear instancias del modelo Mensaje y guardarlas en la base de datos.

Reflexión sobre las Dificultades Encontradas y Cómo se Resolvieron

-Mensaje sobre todos los imports (from) del proyecto.

Mensaje: "La importación "django.db" no se ha podido resolver desde el origenPylancereportMissingModuleSource" resaltado en amarillo, en distintos archivos de configuración de la aplicación.
La ide de Vscode sugirió otro intérprete. Al parecer se produjo por un error inesperado con el entorno virtual desde vs code
Se pudo corregir al seleccionar de forma manual el intérprete (python 3.10) con el path del entorno virtual (venv).

-Con estilos de css
Se tuvo que configurar Django en el archivo settings.py para que tenga las siguientes configuraciones:

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
BASE_DIR / "static",
]
```

Se cargó el archivo CSS en la plantilla HTML agregando {% load static %} al principio del archivo y el enlace al archivo CSS dentro de la etiqueta <head>.

Fuente: https://programacionfacil.org/blog/django-aplicar-estilos-css/#google_vignette

-Dificultad en hacer push en Github:
Basandome en los apuntes subidos, avancé hasta la parte donde se debe implementar “git add .” en la carpeta raíz del proyecto. El mensaje de error que surgió fue por la autenticación: “Identidad del autor desconocido”
Con ‘git config --global user.email’ y ‘git config --global user.name’ configure mi correo y usuario globalmente en la terminal. La inicialización del repositorio local ya estaba realizada, se habían agregado los archivos con éxito al igual que el primer commit, pero al introducir: **git push -u origin main** tuve de nuevo los mensaje de error:
“error: src refspec main no concuerda con ninguno”
“error: falló el empuje de algunas referencias a (enlace del proyecto.git)
Solución: El error ocurrió por intentar empujar (push) a una rama llamada `main`, pero mi rama al inicializar el repositorio era `master`.” Se solucionó al introducir el comando especificando la rama correcta: **“git push -u origin master”**

Métodos de envío y recepción de mensajería

Funcion de Búsqueda dentro de la aplicación:

Funcionalidad extra realizada con ayuda de copilot para conseguir una búsqueda más dinámica con el método: '__ icontains'
Explicación: es un filtro específico que permite buscar registros en el campo destinatario que contengan una subcadena dada, sin importar mayúsculas o minúsculas.
Código:
destinatario__icontains=query:
destinatario: Es el nombre del campo en el modelo Mensaje donde se guarda el destinatario.
__icontains: Es una expresión de consulta de Django. El doble guion bajo (__) seguido de icontains le indica a Django que queremos hacer una búsqueda insensible a mayúsculas y minúsculas de una cadena en el campo destinatario.
query: Es el término que el usuario ingresó en la barra de búsqueda para filtrar los mensajes.

Envío de mensajes dentro de la aplicación:

Se implementó una vista en Django que permite a los usuarios enviar mensajes a otros usuarios dentro de la aplicación. La vista maneja tanto la visualización del formulario de envío como el procesamiento de los datos.