Modelo (Model):

El Modelo representa la capa de datos y la lógica de la aplicación. los modelos son clases que representan tablas de una base de datos.

Vista (View):

las vistas son responsables de generar la interfaz de usuario que se presenta al usuario. Pueden ser páginas HTML

Template (Plantilla):

Los template son archivos que contienen el código HTML

Flujo de Funcionamiento:

* El usuario realiza una solicitud a la aplicación web, por ejemplo, al acceder a una URL específica.
* El framework (Django) examina la URL y determina qué vista se debe llamar.
* La vista realiza cualquier lógica de negocio necesaria, que podría implicar la interacción con el modelo para recuperar o modificar datos.
* La vista utiliza una plantilla para generar la respuesta HTML final.
* La respuesta HTML se envía de vuelta al navegador del usuario.

Al elegir el enfoque MVT de Django en lugar del Modelo de 3 Capas, optamos por una solución que no solo hace que el desarrollo sea más fácil, sino que también mejora nuestras posibilidades de crear aplicaciones web exitosas. Django proporciona herramientas y convenciones que facilitan el trabajo del desarrollador y simplifican la creación de aplicaciones web robustas y eficientes. Su diseño específico para el desarrollo web y la fuerte comunidad que respalda este enfoque hacen que sea una elección poderosa para llevar nuestros proyectos a buen término.

las convenciones se refieren a las prácticas comunes y acordadas que los desarrolladores de Django siguen al estructurar sus aplicaciones. Por ejemplo, la forma en que se nombra y organiza el código

Python:

Propósito: Python es el lenguaje principal en el que está escrito Django. Se utiliza para desarrollar la lógica del servidor, los modelos de datos, las vistas y las demás partes del backend de la aplicación web.

HTML (Hypertext Markup Language):

Propósito: HTML se utiliza para estructurar el contenido y la presentación de las páginas web. En Django, las plantillas HTML

CSS (Cascading Style Sheets):

Propósito: CSS se utiliza para aplicar estilos y diseñar el diseño de las páginas web.

JavaScript:

Propósito: JavaScript se utiliza para agregar interactividad y dinamismo a las páginas web.

obstaculos

"Uno de los desafíos más significativos que enfrentamos durante el desarrollo fue la presión del tiempo. Con plazos ajustados, nos vimos obligados a optimizar nuestro enfoque y priorizar tareas para cumplir con los objetivos establecidos. Esta limitación temporal nos desafió a ser más eficientes en nuestra planificación y ejecución, pero también nos brindó la oportunidad de desarrollar habilidades de gestión del tiempo y trabajo bajo presión que resultaron ser activos valiosos en el proceso."

resultado

"A pesar de las limitaciones de tiempo, logramos obtener resultados significativos en nuestro desarrollo. Nos enfocamos en maximizar la eficiencia y la calidad del trabajo, priorizando las tareas críticas para cumplir con los plazos establecidos. Esta estrategia nos permitió cumplir con los objetivos clave del proyecto.