



PROGRAMACIÓN WEB 2



Profesor(a):

Carlo Jose Luis Corrales Delgado

Estudiantes:

Mamani Anahua, Victor Narciso
Cuno Cahuari, Armando Steven

Repositorio GitHub:

<https://github.com/VictorMA18/Lab06-Django>

Video:

28 de mayo, 2024

Clases en Models.py

Este código define tres modelos en Django: Alumno, Curso, y Nota. El modelo Alumno tiene un campo nombre de tipo CharField con un máximo de 255 caracteres. El modelo Curso contiene un campo curso que es un CharField de hasta 100 caracteres y debe ser único. El modelo Nota establece relaciones con los modelos Alumno y Curso mediante claves foráneas (ForeignKey), y tiene tres campos nota, nota2, y nota3, todos ellos de tipo DecimalField con un máximo de cuatro dígitos y dos decimales, inicializados en 0. Además, cada modelo tiene un método str que retorna una representación legible del objeto.

Poner la imagen del código

Funciones en Views.py

El código proporciona seis funciones de vista en Django para gestionar un sistema escolar. La función crear-alumno(request) maneja la creación de nuevos registros de alumnos, mientras que lista-alumnos(request) recupera y muestra todos los registros de alumnos. Por otro lado, crear-curso(request) gestiona la creación de nuevos registros de cursos, y lista-cursos(request) muestra todos los cursos existentes. Además, crear-nota(request) permite la creación de nuevas notas, y lista-notas(request) muestra todas las notas registradas. Cada función valida y procesa los formularios correspondientes, redirigiendo a las páginas de lista respectivas después de guardar los datos en la base de datos. En conjunto, estas funciones proporcionan una interfaz completa para gestionar los datos de alumnos, cursos y notas en el sistema de notas.

Poner la imagen del código

Funciones en Forms.py

El código proporciona tres formularios en Django para los modelos de datos relacionados con un sistema escolar. El formulario AlumnoForm está asociado al modelo Alumno, permitiendo la creación de nuevos registros de alumnos con un campo para el nombre del alumno. CursoForm está vinculado al modelo Curso, facilitando la creación de nuevos registros de cursos. Por último, NotaForm se asocia al modelo Nota, ofreciendo campos para ingresar información sobre la nota de un alumno en un curso específico, incluyendo múltiples notas si es necesario. Cada formulario está diseñado para interactuar directamente con su respectivo modelo, simplificando el proceso de captura y almacenamiento de datos relacionados con el sistema escolar.

Poner la imagen del código

Funciones en Urls.py en la aplicación

El bloque de código define un conjunto de rutas en Django que asignan solicitudes HTTP a funciones de vista específicas dentro de la aplicación. Cada ruta, configurada mediante el método path(), está asociada a una función de vista correspondiente en el módulo de vistas. Por ejemplo, /crearAlumno/ dirige las solicitudes a la función crear-alumno, encargada de la creación de nuevos registros de alumnos, mientras que /listaAlumno/ muestra la lista de todos los alumnos registrados utilizando la función lista-alumnos. Además, /crearNota/ dirige las solicitudes a la función crear-nota para crear nuevas notas, y /listaNota/ muestra todas las notas registradas utilizando la función lista-notas. Este enfoque permite a los usuarios interactuar con diversas funcionalidades del sistema de notas, como la gestión de alumnos, cursos y notas, a través de URLs específicas y funciones de vista asociadas.

Poner la imagen del código

Funciones en Urls.py en el proyecto

El bloque de código configura las rutas principales de la aplicación Django. La ruta /admin/ dirige a la interfaz de administración predeterminada de Django, mientras que la ruta /notas/ es manejada por la aplicación colegio.urls, que a su vez incluye un conjunto de rutas definidas en el archivo urls.py de la aplicación colegio. Esto permite una estructura de URL modular, donde las rutas relacionadas con las notas (definidas en colegio.urls) se pueden gestionar de manera independiente de las rutas de administración u otras partes de la aplicación.

Poner la imagen del código

Los archivos HTML

crear_alumno.html

Poner la imagen del código

crear_curso.html

Poner la imagen del código

crear_nota.html

Poner la imagen del código

lista_alumnos.html

Poner la imagen del código

lista_cursos.html

Poner la imagen del código

lista_notas.html

Poner la imagen del código

Ejecutar el servidor

El comando `python manage.py runserver` se utiliza en Django para iniciar el servidor de desarrollo local. Una vez ejecutado, el servidor se activa y permite acceder a la aplicación web en desarrollo a través de un navegador en la dirección `http://localhost:8000/` por defecto. Es una herramienta fundamental durante el proceso de desarrollo de aplicaciones web con Django, ya que proporciona un entorno de prueba para probar y depurar el código antes de desplegar la aplicación en un entorno de producción.

```
python manage.py runserver
```

Acceder a la aplicación en el navegador

Crear un alumno en <http://127.0.0.1:8000/notas/crearAlumno/>

Crear un curso en <http://127.0.0.1:8000/notas/crearCurso/>

Crear una nota en <http://127.0.0.1:8000/ventas/crearNota/>

URL de video de explicación:

URL de repositorio de GitHub: <https://github.com/VictorMA18/Lab06-Django>