Despliegue del proyecto

Notas:

- En este documento me referiré a la carpeta Django que contiene el BackEnd como carpeta BackEnd, y a la carpeta React como carpeta FrontEnd.
- Para el desarrollo de este proyecto lo hice desde una maquina con sistema operativo Linux Mint 21.3 x86_64, con Python 3.10.12 y Node v22.5.1.
- Usuario administrador
  - Nombre de usuario: admin
  - Password: Admin123
- Usuario de ejemplo
  - Nombre de usuario: rosalio
  - Password: rosalio

Despliegue de BackEnd

Para desplegar el proyecto BackEnd es necesario usar el lenguaje de programación de Python, yo uso la versión de **Python 3.10.12**, en mi recomendación personal es mejor hacerlo desde un entorno virtual para tener un entorno aislado, luego instalar paquetes y desplegar migraciones.

Pasos

Crear entorno virtual:

Desde la cmd (Windows) o terminal (Sistemas operativos basados en Unix) ejecutar el comando, este paso solo es necesario sino tenemos nuestro entorno virtual que usualmente solo se adra la primera vez que iniciemos:
Windows
python -m venv venv
Sistemas operativos basados en Unix
python3 -m venv venv

Activar entorno virtual

Cada vez que instalemos paquetes o iniciemos nuestra aplicación debemos tener activo nuestro entorno virtual.
Para activarlo desde la carpeta de nuestro proyecto BackEnd y creado el entorno virtual se debe ejecutar el siguiente comando:
Windows:
venv\Scripts\activate.bat
Sistemas operativos basados en Unix:
source venv/bin/activate

Instalar paquetes

Este paso es necesario para la primera vez que iniciamos nuestro proyecto, o algún colega del código agrega un nuevo paquete, recordar que es necesario tener nuestro entorno virtual activo.
pip install -r requirements.txt

Esto instalara nuestros paquetes.

Configurar de base de datos.

Para esta prueba técnica he dejado la base de datos de SQLite que he usado, para el desarrollo del proyecto ya que tiene configurado usuarios, grupos, origen de movimiento, tipos de metas, tipos de movimiento, tipos de ahorros.
De lo contrario habría que agregar registros en la base de datos, esto se puede hacer desde un gestor de base de datos, en mi caso use Dbeaver.

Inserciones en la base de datos

Origen de movimiento
INSERT INTO finanzas_origenmovimiento (id, nombre) VALUES(1, 'Ingreso');
INSERT INTO finanzas_origenmovimiento (id, nombre) VALUES(2, 'Gasto');

Tipo de movimiento
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(1, 'Apertura', '2024-08-04', '2024-08-04', NULL, 1, NULL);
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(2, 'Salario', '2024-08-04 00:00:00', '2024-08-04 13:35:29.463654', NULL, 1, NULL);
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(3, 'Comida', '2024-08-04', '2024-08-04', NULL, 2, NULL);
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(4, 'Transporte', '2024-08-04', '2024-08-04', NULL, 2, NULL);
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(5, 'Vivienda', '2024-08-04', '2024-08-04', NULL, 2, NULL);
INSERT INTO finanzas_tipomovimiento (id, nombre, created_at, updated_at, deleted_at, origen_id, usuario_id) VALUES(6, 'Regalias', '2024-08-04 18:52:41.832853', '2024-08-04 18:52:41.832879', NULL, 1, NULL);

Tipo de meta
INSERT INTO finanzas_tipometa (id, nombre) VALUES(1, 'Corriente');
INSERT INTO finanzas_tipometa (id, nombre) VALUES(2, 'Por cantidad');
INSERT INTO finanzas_tipometa (id, nombre) VALUES(3, 'Por fecha');

Tipo de ahorro
INSERT INTO finanzas_tipoahorro (id, nombre, created_at, updated_at, deleted_at, usuario_id, tipo_meta_id) VALUES(1, 'Apertura de cuenta', '2024-08-04', '2024-08-04', NULL, NULL, 1);

Grupo
INSERT INTO auth_group (id, name) VALUES(1, 'admin');

Usuario
INSERT INTO core_app_user (first_name, last_name, id, password, last_login, is_superuser, username, email, is_staff, is_active, date_joined) VALUES('', '', 1, 'pbkdf2_sha256$720000$zXtRcnOCilGgiHgyMgtrma$VS2EQ/6n7zisO1TgOB03aKjDpPgtMqQ89UPvWcolSGA=', '2024-08-04 02:04:50', 1, 'admin', 'admin@admin.com', 1, 1, '2024-08-03 22:31:03.357806');

Grupo Usuario
INSERT INTO core_app_user_groups (id, user_id, group_id) VALUES(1, 1, 1);

Estos registros estan ligados a un archivo de constantes en el BackEnd y FrontEnd.

Archivo .env
Es necesario configurar nuestro archivo .env dentro de la carpeta del BackEnd contiene las claves valor validas para nuestro proyecto, para esta prueba tecnica he deja
SIGNING_KEY = 'rosalio'
JWT_ALGORITHM = 'HS256'

Correr nuestro proyecto

Para correr nuestro proyecto, luego de haberlo configurado y con nuestro entorno virtual activo, desde terminal o cmd ejecutar el comando

Windows:
python manage.py runserver

Sistemas operativos basados en Unix, el comando python tambien puede servir en Unix si se ha configurado previamente:
python3 manage.py runserver

Si todo se ha realizado correctamente, la terminal nos mostrara

System check identified no issues (0 silenced).
August 05, 2024 - 15:09:11
Django version 5.0.7, using settings 'core.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

Podemos acceder desde un navegado a la ruta mencionada http://127.0.0.1:8000/
En esa ruta también tendremos la documentacion y el panel administrador
Documentación: http://127.0.0.1:8000/*docs*
Panel administrador: *http://127.0.0.1:8000/admin*

Despliegue del FrontEnd

Para el despliegue del FrontEnd es necesario tener instalado node, en mi caso yo use la versión de node v22.5.1

Pasos

Instalar paquetes

Desde la cmd de Windows o la terminal de sistemas operativos Unix, entrando en la carpeta del FrontEnd ejecutar el comando:
npm i
Eso instalara los paquetes necesarios para nuestro proyecto

Configuración

Por defecto he dejado que nuestro proyecto FrontEnd conecte con nuestro BackEnd a través de la ruta http://localhost:8000 pero si nuestro proyecto esta configurado en un puerto o ruta distinto, es necesario ajustar nuestra ruta para ellos de cualquier editor de texto entramos al archivo de la carpeta FrontEnd "**/src/libs/axios.js**" y en la variable «const ruta = "http://localhost:8000"; » ajustamos nuestra ruta del BackEnd.

Desplegar proyecto

Para desplegar el proyecto he configurado dos opciones:

Modo desarrollo:
Para ejecutar el modo desarrollo desde la cmd o terminal de nuestro carpeta BackEnd ejecutamos el comando
npm run dev
Esto hará que nuestro proyecto al iniciarse se cargue todos los archivos, y en cualquier cambio que realicemos hará un recarga automática de nuestros cambios.

Despues de ejecutar nuestro comando nos devolvera

```
  VITE v5.3.5  ready in 389 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Donde el puerto http://localhost:5173/ es nuestra ruta donde estara desplegado nuestro FrontEnd en modo desarrollo, podemos cambiar el puerto de nuestro FrontEnd pero si hacemos esto tambien sera necesario ajustar los **CORS_ALLOWED_ORIGINS** de el archivo "**/django/core/settings.py**" agregando también nuestra ruta.

Modo despliegue:
Para ejecutar el modo desarrollo desde la cmd o terminal de nuestro carpeta BackEnd ejecutamos el comando
npm run deploy

Después de ejecutarse nuestra terminal nos devolverá

vite v5.3.5 building for production...
✓built in 8.96s
➜ Local: http://localhost:4173/
➜ Network: http://192.168.1.44:4173/
➜ press h + enter to show help

Donde el puerto http://localhost:4173/ es nuestra ruta donde estara desplegado nuestro FrontEnd en modo despliegue, podemos cambiar el puerto de nuestro FrontEnd pero si hacemos esto tambien sera necesario ajustar los **CORS_ALLOWED_ORIGINS** de el archivo "**/django/core/settings.py**" agregando también nuestra ruta.

Esto hará que nuestro proyecto al iniciarse solo se cargue los archivos necesarios, por ende es mas eficiente, y en cualquier cambio que realicemos no hará un recarga de nuestros cambios, solo la hará si volvemos a ejecutar el comando de despliegue.

Existen estas dos alternativas, una para nuestro servidor con vista a los clientes y la otra para nuestros compañeros de código y nosotros cuando estemos desarrollando.