Steaven Johann Miranda
BSIT - 3R1

Short Report About Django Rest Framework and FastAPI

Aspect	Django REST Framework	FastAPI
Key Features	It has a built-in admin dashboard or authentication and permission system	Auto-generated interactive API documentation
	Built on top of Django, which is a full-stack framework.	Independent framework, used for APIs
	Highly customizable	Fast development
Advantages	Ideal for building complex, full-stack applications with ORM, admin panel, and other Django features	Extremely fast performance, ideal for I/O-bound tasks and high-traffic APIs
	Built-in support for authentication and permission management.	Automatic OpenAPI and Swagger documentation generation.
	Database migrations are easy with Django's built-in migration system.	Uses Pydantic models for automatic data validation, serialization, and parsing.
Disadvantages	More complex due to the need for serializers, views, and permissions.	Too minimalistic, requires building custom components and manual configurations for larger application
	Slower compared to FastAPI due to its synchronous nature and heavier framework.	Does not provide migration tools; best for use with external databases.
	Dependent on Django, difficult to use outside Django Framework	No Built-In Admin or ORM

Challenges in:

Django REST Framework

One development challenge was configuring the settings.py file for deployment. The settings needed to be adjusted for different environments, such as setting up ALLOWED_HOSTS, handling production database configurations, and managing static files. I solved this by creating separate configuration files for development and deployment and using environment variables to ensure secure and flexible deployment configurations. Another challenge during development was setting up the build command, which was different from the typical Django setup.. To overcome this, I created a custom script that ensured proper handling of static files, migrations, and testing commands before deploying.

FastAPI

The primary challenge was the unfamiliarity with the framework, given its newer nature. To address this, I reviewed the other simpler activity helping me understand its structure and asynchronous capabilities more easily. I also watched several YouTube videos and leveraged ChatGPT to ask about the specific functions used in each line of code, helping me break down and understand what each function does in FastAPI. In terms of deployment, there were no major issues since FastAPI's lightweight nature made deployment straightforward. The minimal number of configuration files and reliance on simple ASGI servers like Uvicorn meant there were fewer complications, and the deployment was more seamless compared to Django.