

Claire Marie Butong
BSIT – 3R1

Difference between Django REST Framework (DRF) and FastAPI (highlight key features, advantages, disadvantages).

- As a student learning APIs, I find FastAPI more flexible as it provides better user interface and kinda easy to use. Meanwhile the DRF complicates me, I've encountered more issues with it than FastAPI. Based on features FastAPI is better as it provides better UI and its not complicated unlike DRF which required more complexity. But if we're talking about security, I think DRF is better as it has Django admin and is a time-saver for data management and has built-in tools like authentication. Above all both frameworks have its weaknesses and strengths, it just depends on what type of project you're building. FastAPI is fast, lightweight, and fun — but takes some skill to use. DRF is versatile and sturdy, great if you already know Django.

Challenges you encountered in development and deployment and how you solved them for both DRF and FastAPI.

- While doing this project, I've encountered the following challenges:
 - o Pushing my codes in my github repo - Git refused to push my local changes because the remote already had different commits. To fix this, I tried using **git pull --rebase** command but it only overwrote my files and removing my local files. After using **git reflog**, I've had retrieved my files back and use **git push origin main --force** to push my files to my repo.
 - o The CORS policy errors - while deploying my fronted it encountered blocking access from accessing my backend due to CORS restrictions. To fix this, I correctly configured the **CORSMiddleware** in my **main.py**.
 - o Typos – I didn't notice the typos from my frontend which caused the deployment and functions to not work properly.
 - o Frontend API URLs not working - Incorrect or blocked API requests due to missing CORS headers. To fix it I ensured my React frontend used the correct API base URL, and updated CORS settings on the FastAPI backend.
- Afterall, I've resolved my issues and successfully deployed both my backend and frontend.