Backend requests:

1. init nest js project in backend folder
2. add module named video for streaming
3. init React project in frontend folder
4. Complete README to start BE and FE
5. Add nestjs typeorm and config postgres with variable in .env
6. Add video entity
7. Add API to find all videos from DB
8. Add video module to app module
9. Add category with enum and cover image to entity
10. Add filters for find all video
11. Add DTO for filters and update swagger
12. Instead of title and description, add search as query
13. Create docker compose that includes postgres and minio
14. add storage module and service to upload S3
15. Upload file to S3 with public access
16. Seed video fake data as a typeorm migration
17. Add API for find one video
18. Add test for video module
19. Add access-level free and premium

Frontend requests:

1. add api base url to .env
2. Add side menu
3. Update items to :Dashboard, Smart Calendar, Qrated, Expert, Ask AI, Belly-safe, Mamma Moves
4. Create page for Mamma Moves and call video API from BE
5. update base url from .env
6. Based on the categories of each video from BE, put them in the seprate section
7. Create video page and play each video
8. If accessLevel is not "free" when user click on it redirect to plan price page
9. Blur the thumnail image if video item is premium and mention on it a premium title
10. All item of each section should be in on row, if they are more than width, it shold be scrolable

Admin requests:

1. add admin js in nestjs and postgres
2. read admin credential from .env
3. add video entity as new resource of admin

Cursor issues:

1. Cursor can’t modify files with format \*.ipynb or package.json
2. In some cases, Cursor can’t modify even simple file and stuck in a loop
3. In some cases, Cursor removes some important part of code to solve the problem
4. Log out, it seems For more than 2 or 3 devices, Cursor will log out

Cursor advantages:

1. Great for Documentation
2. Implement structure, naming