**BACK-END SCRIPT**

1. Introduction.

* Uses NestJS framework and using TypeScript.
* Using Restful API to communicate between back-end and front-end.
* Architecture:
  + PostgreSQL for the main database.
  + 2 VMs to operate this project, one is API server, one is Redis, NGINX and n8n servers.
* Uses varies libraries to develop this project:
  + TypeORM to standardize the SQL for NestJS.
  + Class-validator for DTO.
  + Passport Strategy for JWT authentication.
  + Cloudinary for third-party image storage.
  + Redis for caching.
  + Socket.io for real-time chat, notification.

2. User.

* CRUD for user.
* Talk every single one of the API.

3. JWT and auth.

* Uses 2 layers of tokens (access and refresh).
* Log in 2 ways (google or manual).
  + Manual, make the manual way -> email, password.
  + Google, use the Google account to create the new account.
    - The google.strategy.ts get the gmail’s information such as email and nickname.
    - The findOrCreateUserGoogle method uses that information to create the new account or return the existing account.
    - Sign using the login method as usual.
* If the users want to use the API, they must have the access token.
  + UseGuards(AuthGuards(jwt)) requires user to has the valid access token in order to use the API.
  + The payload is validated by the jwt.strategy.ts and store the payload in the req.user.
* Talk every single remaining one of the API.

4. Error handler.

* The error system for the application.
* Makes the front-end easier to detect the error.