**1. โครงสร้างโฟลเดอร์แนะนำ**

src/

main.ts // จุดเริ่มต้นแอป

app.module.ts // รวมทุก module

app.controller.ts // API หลัก (ถ้ามี)

app.service.ts // Service หลัก (ถ้ามี)

config/ // การตั้งค่า (เช่น config.service.ts)

database/ // การเชื่อมต่อ DB, migration, entity

entities/ // Entity ของแต่ละ table

migrations/ // Migration script

database.module.ts // Module สำหรับ DB

modules/ // Feature หลักแต่ละกลุ่ม (แยกตาม business)

auth/ // Auth (login, JWT, user)

auth.controller.ts

auth.service.ts

auth.module.ts

dto/

guards/

strategies/

user/ // User CRUD

user.controller.ts

user.service.ts

user.module.ts

dto/

entities/

log/ // Login log, audit log

log.controller.ts

log.service.ts

log.module.ts

entities/

token/ // User token, refresh token

token.controller.ts

token.service.ts

token.module.ts

entities/

common/ // ตัวช่วย, utility, pipe, filter, interceptor

pipes/

filters/

interceptors/

utils/

types/ // Type definition

## 2. ****หลักการตั้งชื่อ****

* **Module**: ตาม business เช่น [user](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), auth, [log](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), [token](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
* **Controller**: จัดการ API route เช่น [user.controller.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
* **Service**: จัดการ logic เช่น [user.service.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
* **Entity**: โครงสร้าง table เช่น [user.entity.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
* **DTO**: รับ/validate ข้อมูล เช่น [create-user.dto.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), [login.dto.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)
* **Common**: ตัวช่วยที่ใช้ซ้ำหลายที่ เช่น utils, pipes, filters

## 3. ****ตัวอย่างการแยก module แบบมืออาชีพ****

### User Module

modules/user/

user.controller.ts

user.service.ts

user.module.ts

dto/

create-user.dto.ts

update-user.dto.ts

entities/

user.entity.ts

### Auth Module

modules/auth/

auth.controller.ts

auth.service.ts

auth.module.ts

dto/

login.dto.ts

guards/

jwt-auth.guard.ts

strategies/

jwt.strategy.ts

### Database Module

database/

entities/

user.entity.ts

login-log.entity.ts

user-token.entity.ts

database.module.ts

**4. ข้อดีของโครงสร้างนี้**

* **ขยายง่าย**: เพิ่ม feature ใหม่แค่เพิ่ม module
* **ดูแลง่าย**: แต่ละ module แยกกันชัดเจน
* **ขายงานได้**: โครงสร้างนี้เป็นมาตรฐานที่บริษัทใหญ่ใช้
* **รองรับทีมงาน**: หลายคนทำงานพร้อมกันได้ ไม่ชนกัน

**5. ตัวอย่างชื่อไฟล์/โฟลเดอร์**

* modules/user/user.controller.ts → API สำหรับ user
* [auth.service.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) → Logic สำหรับ auth
* [user.entity.ts](vscode-file://vscode-app/c:/Users/user/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) → โครงสร้าง table user
* common/utils/hash.util.ts → ตัวช่วย hash password

**6. คำแนะนำเพิ่มเติม**

* ใช้ชื่อภาษาอังกฤษทั้งหมด
* ตั้งชื่อให้สื่อความหมายตามหน้าที่
* ถ้า feature ใหญ่มาก แยก module ย่อยได้อีก
* ทุก module มี controller, service, dto, entity ของตัวเอง

## 7. ****ตัวอย่างโครงสร้างจริง (พร้อมขายงาน)****

src/

main.ts

app.module.ts

config/

database/

entities/

migrations/

database.module.ts

modules/

auth/

user/

log/

token/

common/

utils/

pipes/

filters/

interceptors/

types/