




เอกสารประกอบระบบ

1. เอกสารประกอบระบบ

1.1 คู่มือการติดตั้งและใช้งานแอปพลิเคชัน

เครื่องมือที่จำเป็น

- Python 3.8+ สามารถโหลดได้จาก  Welcome to Python.org
- Node.js + npm สามารถโหลดได้จาก  Node.js — Run JavaScript Everywhere
- PostgreSQL สามารถโหลดได้จาก  PostgreSQL

ขั้นตอนการติดตั้ง

1. Clone หรือดาวน์โหลดโปรเจกต์

1. git clone https://github.com/Lighting1427/LoginSystem_Web.git
2. cd solidiThai_assign1

2. ติดตั้งและรัน Backend

1. cd backend
2. python -m venv venv
3. venv\Scripts\activate
4. pip install annotated-types anyio bcrypt cffi click colorama cryptography ecdsa fastapi greenlet h11 idna passlib pycopg2-binary pyasn1 pycparser pydantic pydantic_core python-jose python-multipart rsa six sniffio SQLAlchemy starlette typing_extensions typing-inspection uvicorn
5. uvicorn app.main:app --reload --host 0.0.0.0 --port 8000

ระบบจะตรวจสอบและสร้าง DB loginDB ให้อัตโนมัติถ้ายังไม่มี

เปิด <http://localhost:8000/docs> เพื่อตรวจสอบ Swagger UI

3. ติดตั้งและรัน Frontend

1. cd ../frontend
2. npm install
3. npm start

เปิด <http://localhost:3000> เพื่อเข้าใช้งานหน้าเว็บ

1.2 วิธีการใช้งานระบบ

1. เข้าใช้งานครั้งแรกจะถูกพาไปหน้า Login
2. ใส่ username และ password (ค่าเริ่มต้น: admin / admin123)
3. เข้าสู่หน้า User Management

- สามารถแสดงข้อมูลผู้ใช้
- สร้าง / แก้ไข / ลบผู้ใช้

1.3 เอกสาร API (Swagger / Postman)

เปิดใช้งาน Backend แล้วเข้า <http://localhost:8000/docs> เพื่อดู API ทั้งหมด:

Method	Endpoint	คำอธิบาย
POST	/token	เมื่อ Login จะส่ง username/password เพื่อรับ JWT
GET	/users	ดึงผู้ใช้ทั้งหมด (ต้องมี JWT)
GET	/users/{user_id}	ดึงผู้ใช้ตาม ID (JWT)
POST	/users	สร้างผู้ใช้งานใหม่ (JWT)
PUT	/users/{user_id}	อัปเดตผู้ใช้ (JWT)
DELETE	/users/{user_id}	ลบผู้ใช้ (JWT)

1.4 แนวทางการออกแบบระบบ

Backend (FastAPI + PostgreSQL)

- โครงสร้าง Module
 - auth.py – จัดการ JWT Authentication
 - crud.py – ฟังก์ชัน CRUD สำหรับ DB
 - database.py – เชื่อมต่อและสร้าง DB
 - models.py – Schema ของ User (SQLAlchemy)
 - schemas.py – Pydantic Schemas
 - main.py – Routing หลัก /users, /token
- ใช้ SQLAlchemy เชื่อมต่อ PostgreSQL
- ป้องกัน Endpoint ด้วย JWT Token
- สร้าง User admin (id=999) อัตโนมัติใน event startup

Frontend (React + Axios + React Router)

- หน้า Login – ใช้ /token เพื่อรับ JWT
- หน้า User Management – เรียก /users ด้วย JWT
- ใช้ React Router แยกเส้นทาง /login กับ /app
- ใช้ Axios ใน userApi.js เรียก REST API
- JWT เก็บใน localStorage + State
- ฟอรั่มแก้ไข User: ถ้าไม่ใส่ password จะใช้ password เดิม