**1. Khởi động hệ thống**

Trước khi sử dụng UI, cần đảm bảo hệ thống đã được chạy đầy đủ.

**Bước 1.1: Chạy NanoMQ**

* Mở terminal và chạy NanoMQ bằng Docker:

bash

CollapseWrapCopy

docker run -d -p 1883:1883 nanomq/nanomq:0.6.6-slim

* Kiểm tra container:

bash

CollapseWrapCopy

docker ps

**Bước 1.2: Chạy Redis**

* Mở terminal khác và chạy Redis:

bash

CollapseWrapCopy

docker run -d -p 6379:6379 redis

* Kiểm tra:

bash

CollapseWrapCopy

docker ps

**Bước 1.3: Chạy các service Python**

* Trong thư mục dự án (đã có venv và cài đặt requirements.txt):
  + Terminal 1:

bash

CollapseWrapCopy

venv\Scripts\activate *# Windows*

python listener.py

* + Terminal 2:

bash

CollapseWrapCopy

venv\Scripts\activate

python processor.py

* + Terminal 3:

bash

CollapseWrapCopy

venv\Scripts\activate

python publisher.py

* + Terminal 4:

bash

CollapseWrapCopy

venv\Scripts\activate

python ui.py

* Đợi ui.py khởi động, bạn sẽ thấy thông báo như Running on http://0.0.0.0:5000.

**2. Truy cập UI**

**Bước 2.1: Mở trình duyệt**

* Mở trình duyệt web (Chrome, Firefox, v.v.) và nhập:

text

CollapseWrapCopy

http://localhost:5000

* Bạn sẽ được chuyển đến trang đăng nhập.

**Bước 2.2: Đăng nhập**

* **Trang Login**:
  + **Username**: Nhập admin.
  + **Password**: Nhập admin123.
  + Nhấn nút "Login".
* Nếu thông tin đúng, bạn sẽ được chuyển đến **Dashboard**. Nếu sai, sẽ thấy thông báo "Invalid credentials".

**3. Sử dụng các chức năng trong UI**

**3.1. Xem Dashboard**

* **Mục đích**: Xem tổng quan hệ thống thời gian thực.
* **Cách sử dụng**:
  1. Sau khi đăng nhập, bạn tự động vào trang Dashboard (http://localhost:5000/).
  2. Quan sát các thông tin:
     + **System Status**: Số Gateway, Gateway Online, Beacons Detected, trạng thái Broker/Redis/Listener/Processor/Publisher, và Alerts (nếu có).
     + **Messages/Second**: Số tin nhắn trong beacon\_data.
     + **AWS Updates/Second**: Số tin nhắn trong aws\_queue.
  3. Dữ liệu tự động cập nhật mỗi giây qua WebSocket (biểu đồ và số liệu thay đổi theo thời gian thực).

**3.2. Quản lý Gateways**

* **Mục đích**: Xem danh sách Gateway, số Beacon liên kết, và điều khiển Gateway (bật/tắt).
* **Cách sử dụng**:
  1. Nhấn vào mục "Gateways" trên thanh điều hướng (hoặc truy cập http://localhost:5000/gateways).
  2. **Xem danh sách**:
     + Thấy danh sách Gateway (GW1, GW2, GW3) với IP, trạng thái (Online/Offline), và số Beacon.
     + Có thể lọc theo trạng thái (Online/Offline) hoặc ID bằng thanh tìm kiếm.
  3. **Điều khiển Gateway**:
     + Bên cạnh mỗi Gateway, nhấn nút "On" hoặc "Off".
     + Xác nhận hành động (nếu giao diện yêu cầu).
     + Trạng thái Gateway sẽ thay đổi (Online → Offline hoặc ngược lại), và lệnh được gửi tới NanoMQ qua topic control/<gateway\_id>.

**3.3. Xem thông tin Beacons**

* **Mục đích**: Xem danh sách Beacon, Gateway liên kết, RSSI, và trạng thái phát hiện.
* **Cách sử dụng**:
  1. Nhấn vào mục "Beacons" trên thanh điều hướng (hoặc http://localhost:5000/beacons).
  2. **Xem danh sách**:
     + Thấy các Beacon với beacon\_id, gateway, rssi, last\_seen, và detected (1 = phát hiện, 0 = không).
     + Có thể lọc theo Gateway, trạng thái phát hiện, hoặc ID qua các bộ lọc.
  3. **Quan sát biểu đồ**:
     + Biểu đồ RSSI thay đổi theo dữ liệu gần nhất (100 bản ghi cuối từ beacon\_data).

**3.4. Quản lý cấu hình**

* **Mục đích**: Xem và chỉnh sửa các tham số hệ thống.
* **Cách sử dụng**:
  1. Nhấn vào mục "Config" (hoặc http://localhost:5000/config).
  2. **Xem cấu hình**:
     + Thấy danh sách tham số như WINDOW\_SIZE, RSSI\_THRESHOLD, FREQ\_THRESHOLD, v.v.
  3. **Chỉnh sửa**:
     + Nhập giá trị mới vào ô tương ứng (ví dụ: đổi WINDOW\_SIZE từ 10 thành 20).
     + Nhấn "Save" để lưu vào Redis.
     + Trang sẽ hiển thị thông báo "Success" nếu lưu thành công.

**3.5. Xem Logs**

* **Mục đích**: Theo dõi hoạt động hệ thống qua log.
* **Cách sử dụng**:
  1. Nhấn vào mục "Logs" (hoặc http://localhost:5000/logs).
  2. **Xem log**:
     + Thấy 50 bản ghi log gần nhất từ logs trong Redis (thời gian, service, message).
     + Có thể lọc theo service (Listener, Processor, v.v.) hoặc thời gian.
  3. **Xem số liệu**:
     + Thấy thông tin processing\_time và latency (giá trị mẫu: 50ms, 200ms).

**3.6. Đổi mật khẩu**

* **Mục đích**: Cập nhật mật khẩu tài khoản.
* **Cách sử dụng**:
  1. Nhấn vào mục "Change Password" (hoặc http://localhost:5000/change-password).
  2. Nhập:
     + **Old Password**: admin123.
     + **New Password**: Mật khẩu mới (ví dụ: newpass456).
  3. Nhấn "Change".
     + Nếu mật khẩu cũ đúng, mật khẩu được cập nhật trong Redis, chuyển về Dashboard.
     + Nếu sai, thấy thông báo "Old password incorrect".

**3.7. Đăng xuất**

* **Mục đích**: Thoát khỏi phiên làm việc.
* **Cách sử dụng**:
  1. Nhấn "Logout" trên thanh điều hướng (hoặc http://localhost:5000/logout).
  2. Bạn sẽ được đưa về trang đăng nhập.

**4. Quy trình mẫu sử dụng UI**

1. **Khởi động hệ thống**:
   * Chạy NanoMQ, Redis, listener.py, processor.py, publisher.py, ui.py.
2. **Truy cập và đăng nhập**:
   * Mở http://localhost:5000, đăng nhập với admin/admin123.
3. **Kiểm tra trạng thái hệ thống**:
   * Vào Dashboard, thấy 3 Gateway, 0 Beacon (nếu chưa có dữ liệu).
4. **Mô phỏng dữ liệu**:
   * Chạy script gửi dữ liệu giả lập:

python

CollapseWrapCopy

import paho.mqtt.client as mqtt

import json

client = mqtt.Client()

client.connect("localhost", 1883, 60)

client.publish("bluetooth/GW1/data", json.dumps({

"beacon\_id": "Beacon1",

"rssi": -70,

"timestamp": "2025-03-11T10:00:00"

}))

* + Quay lại Dashboard, thấy "Beacons Detected: 1".

1. **Điều khiển Gateway**:
   * Vào Gateways, tắt GW1 (nhấn "Off"), kiểm tra trạng thái đổi thành "Offline".
2. **Xem chi tiết**:
   * Vào Beacons, thấy "Beacon1" với Gateway "GW1", RSSI -70.
   * Vào Logs, thấy log từ Listener: "Pushed: Beacon1, GW1".
3. **Cấu hình**:
   * Vào Config, đổi RSSI\_THRESHOLD thành -80, nhấn "Save".
4. **Đăng xuất**:
   * Nhấn "Logout", quay lại trang đăng nhập.

**5. Lưu ý khi sử dụng**

* **Yêu cầu**: NanoMQ và Redis phải chạy để UI hoạt động đầy đủ.
* **Debug**: Nếu UI không hiển thị dữ liệu, kiểm tra terminal của ui.py hoặc log trong Redis.
* **Thời gian thực**: Dashboard cần WebSocket (đảm bảo trình duyệt hỗ trợ).