TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

TRẦN NGỌC NGUYÊN HẠNH - 52000554
VÕ MINH KHOA - 52000351

PHÁT TRIỂN ỨNG DỤNG IOT ĐỂ BẢO VỆ AN NINH GIA ĐÌNH

DỰ ÁN CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

TRẦN NGỌC NGUYÊN HẠNH - 52000554
VÕ MINH KHOA - 52000351

PHÁT TRIỂN ỨNG DỤNG IOT ĐỂ BẢO VỆ AN NINH GIA ĐÌNH

DỰ ÁN CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

Người hướng dẫn
TS.Bùi Quy Anh

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2025

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn Khoa Công nghệ Thông tin – nơi đã tạo điều kiện thuận lợi về cơ sở vật chất, môi trường học tập và định hướng giúp em có cơ hội được thực hiện môn học Dự án Công nghệ Thông Tin một cách hiệu quả và thực tế.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến Thầy Bùi Quy Anh, người đã tận tình hướng dẫn, truyền đạt những kiến thức quý báu, định hướng và hỗ trợ em trong suốt quá trình thực hiện dự án. Sự tận tâm và chỉ dẫn của thầy đã giúp em hiểu rõ hơn về quy trình phát triển phần mềm, kỹ năng làm việc nhóm cũng như cách áp dụng kiến thức vào thực tiễn.

Qua môn học này, em không chỉ củng cố được kiến thức chuyên môn mà còn học hỏi thêm nhiều kinh nghiệm quý giá, giúp em tự tin hơn trên con đường học tập và nghề nghiệp sau này.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày ... tháng ... năm 20..

Tác giả

(Ký tên và ghi rõ họ tên)

Trần Ngọc Nguyên Hạnh

Võ Minh Khoa

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH
TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS.Bùi Quy Anh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày ... tháng ... năm 2025

Tác giả

(Ký tên và ghi rõ họ tên)

Trần Ngọc Nguyên Hạnh

Võ Minh Khoa

PHÁT TRIỂN ỨNG DỤNG IOT ĐỂ BẢO VỆ AN NINH GIA ĐÌNH

TÓM TẮT

Đề tài tập trung xây dựng một hệ thống IoT giám sát an ninh gia đình thông minh dựa trên việc tích hợp các thiết bị an ninh với nền tảng Blynk, thay vì phát triển ứng dụng giám sát riêng. Hệ thống bao gồm các thành phần phần cứng như camera giám sát, cảm biến chuyển động, cảm biến cửa và bộ Gateway, kết nối trực tiếp đến hạ tầng Cloud của Blynk.

Dữ liệu thu thập từ các thiết bị IoT được truyền qua Gateway, xử lý và lưu trữ trên Blynk Cloud. Khi phát hiện các sự kiện bất thường như chuyển động lạ hoặc xâm nhập trái phép, hệ thống sẽ gửi thông báo cảnh báo tức thì đến người dùng thông qua nền tảng Blynk, cho phép người dùng theo dõi và giám sát từ xa.

Đề tài chú trọng nghiên cứu kiến trúc IoT, cơ chế truyền thông, bảo mật dữ liệu và khả năng tích hợp nền tảng IoT sẵn có, bao gồm mã hóa dữ liệu, xác thực thiết bị và quản lý truy cập người dùng trên hệ sinh thái Blynk.

Kết quả mong đợi của đề tài bao gồm việc xây dựng một hệ thống Smart Home demo hoạt động ổn định và có giao diện quản lý trực quan, giúp người dùng dễ dàng theo dõi và giám sát các thiết bị trong hệ thống. Bên cạnh đó, đề tài hướng đến việc đề xuất một bộ giải pháp an toàn IoT có thể áp dụng cho mô hình nhà thông minh nhằm nâng cao mức độ bảo mật và hạn chế các rủi ro truy cập trái phép. Đồng thời, báo cáo đi kèm sẽ phân tích chi tiết các lỗ hổng có thể phát sinh trong quá trình triển khai, các biện pháp khắc phục đã áp dụng và đánh giá hiệu quả của những biện pháp đó trong thực tế.

Mục tiêu nghiên cứu của đề tài là xây dựng một mô hình hệ thống giám sát an ninh gia đình IoT hoạt động ổn định dựa trên nền tảng Blynk, đồng thời minh họa rõ khả năng tích hợp các dịch vụ Cloud IoT vào mô hình an ninh gia đình nhằm nâng cao khả năng giám sát và phản hồi theo thời gian thực. Bên cạnh việc triển khai hệ thống, đề tài còn hướng đến việc xây dựng báo cáo phân tích chi tiết về kiến trúc

tổng thể, các cơ chế bảo mật được áp dụng và đánh giá hiệu quả hoạt động của hệ thống thông qua quá trình thử nghiệm thực tế.

Phạm vi nghiên cứu của đề tài tập trung vào các thiết bị IoT an ninh gia đình phổ biến có khả năng tích hợp với nền tảng Blynk, bao gồm camera, cảm biến chuyển động và cảm biến cửa. Đề tài không phát triển một ứng dụng giám sát riêng mà sử dụng trực tiếp nền tảng Blynk để thực hiện chức năng giám sát và nhận thông báo cảnh báo. Nội dung nghiên cứu chủ yếu xoay quanh kiến trúc kết nối giữa các thành phần theo mô hình IoT – Gateway – Cloud Blynk, đồng thời áp dụng các cơ chế bảo mật cơ bản trong quá trình truyền thông dữ liệu của hệ thống. Ngoài ra, đề tài không đi sâu vào việc thiết kế phần cứng mới mà tập trung vào tích hợp, triển khai và đánh giá hoạt động của hệ thống trong điều kiện thực nghiệm.

DEVELOPING IOT APPLICATIONS FOR HOME SECURITY

ABSTRACT

This project focuses on developing an IoT-based home security monitoring system integrated with the Blynk platform, instead of building a standalone monitoring application. The system consists of hardware components such as surveillance cameras, motion sensors, door sensors, and a gateway that connects directly to Blynk Cloud infrastructure.

Sensor data is transmitted through the gateway to Blynk Cloud for processing and storage. When abnormal events such as unusual motion or unauthorized access are detected, real-time notifications are delivered to users via the Blynk platform, enabling remote monitoring and security awareness.

The project emphasizes the study of IoT architecture, communication mechanisms, data security, and cloud platform integration, including data encryption, device authentication, and access control within the Blynk ecosystem.

The expected outcomes of the project include the development of a stable IoT-based home security monitoring model integrated with the Blynk platform, enabling reliable real-time monitoring and alerting. The project also aims to demonstrate the feasibility of integrating cloud-based IoT services into a home security system, highlighting the role of cloud infrastructure in data processing, storage, and notification delivery. In addition, the project provides a detailed analysis of the system architecture, implemented security mechanisms, and overall performance through practical evaluation, thereby validating the effectiveness and applicability of the proposed solution.

The research objectives of the project are to design and develop an IoT application capable of connecting and managing home security devices, including sensors and surveillance components, within an integrated system. In addition, the

project aims to propose and implement appropriate security solutions to protect data and prevent unauthorized access during system operation. Finally, the system's performance is evaluated through real-world testing, focusing on stability, responsiveness, and the effectiveness of the implemented security mechanisms.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC CÁC CHỮ VIẾT TẮT

AI: Artificial Intelligence (trí tuệ nhân tạo)

API: Application Programming Interface (giao diện lập trình ứng dụng)

API Key: Khóa xác thực truy cập API

AWS: Amazon Web Services

Blynk: Nền tảng IoT dùng dashboard/thông báo

ESP32: Vi điều khiển ESP32

ESP32-CAM: Module ESP32 tích hợp camera

HTTP: HyperText Transfer Protocol

HTTPS: HTTP Secure (HTTP qua TLS/SSL)

IAM: Identity and Access Management (quản lý định danh & truy cập AWS)

IoT: Internet of Things (Internet vạn vật)

LAN: Local Area Network (mạng nội bộ)

Node.js: Môi trường chạy JavaScript phía server

PIR: Passive Infrared Sensor (cảm biến chuyển động)

S3: Amazon Simple Storage Service

TLS/SSL: Transport Layer Security / Secure Sockets Layer

Wi-Fi: Wireless Fidelity (mạng không dây)

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Trong bối cảnh công nghệ ngày càng phát triển, các thiết bị IoT (Internet of Things) đã và đang trở thành một phần không thể thiếu trong đời sống hiện đại, đặc biệt là trong lĩnh vực an ninh và giám sát nhà ở. Tuy nhiên, cùng với sự gia tăng của các thiết bị kết nối mạng là những rủi ro tiềm ẩn về an toàn thông tin và bảo mật dữ liệu. Các cuộc tấn công mạng, truy cập trái phép có thể gây ảnh hưởng nghiêm trọng đến người dùng.

Chính vì vậy, việc nghiên cứu và xây dựng một phát triển ứng dụng IoT để bảo vệ an ninh gia đình không chỉ đáp ứng nhu cầu thực tế về giám sát, cảnh báo và điều khiển từ xa, mà còn hướng đến việc đảm bảo an toàn và bảo mật dữ liệu cho người sử dụng.

Đề tài được lựa chọn với mong muốn ứng dụng các kiến thức về Internet of Things (IoT), mạng máy tính và bảo mật thông tin vào việc xây dựng một mô hình thực tế có tính ứng dụng cao trong lĩnh vực an ninh gia đình. Thông qua đề tài, nhóm hướng đến việc giải quyết các vấn đề cấp thiết liên quan đến giám sát và bảo vệ an ninh gia đình thông minh, đặc biệt trong bối cảnh xu hướng nhà thông minh đang phát triển mạnh mẽ. Đồng thời, đề tài còn góp phần nâng cao nhận thức và kỹ năng trong việc tích hợp các giải pháp bảo mật vào hệ thống IoT, hướng đến việc xây dựng các sản phẩm công nghệ an toàn, hiệu quả và thân thiện với người dùng.

Từ đó, đề tài không chỉ có ý nghĩa về mặt học thuật mà còn mang giá trị thực tiễn, có thể mở rộng và ứng dụng trong nhiều mô hình Smart Home khác nhau trong tương lai.

1.2 Mục tiêu thực hiện đề tài

Mục tiêu của đề tài là xây dựng một mô hình hệ thống IoT giám sát an ninh gia đình tích hợp với nền tảng Blynk, cho phép kết nối và quản lý các thiết bị an ninh như camera, cảm biến chuyển động và cảm biến cửa thông qua Gateway và hạ tầng Blynk Cloud. Hệ thống được thiết kế nhằm triển khai cơ chế giám sát và cảnh báo theo thời gian thực, trong đó các sự kiện bất thường được phát hiện và thông báo trực tiếp đến người dùng thông qua ứng dụng Blynk. Bên cạnh đó, đề tài tập trung nghiên cứu và áp dụng các giải pháp bảo mật trong hệ thống IoT, bao gồm mã hóa dữ liệu, xác thực thiết bị và kiểm soát truy cập, nhằm đảm bảo an toàn thông tin trong quá trình vận hành. Cuối cùng, hiệu quả hoạt động của hệ thống được đánh giá thông qua các thử nghiệm thực tế, tập trung vào độ ổn định, độ trễ truyền thông và khả năng cảnh báo của mô hình đề xuất.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Internet of Things (IoT)

2.1.1 Khái niệm về IoT

Internet of Things (IoT) hay còn gọi là "Internet vạn vật" là một hệ thống bao gồm các thiết bị vật lý có khả năng thu thập, truyền nhận và xử lý dữ liệu thông qua mạng Internet mà không cần sự can thiệp trực tiếp của con người. Các thiết bị này có thể là cảm biến, camera, công tắc, khóa cửa, hay các bộ điều khiển tự động. Theo định nghĩa của ITU (International Telecommunication Union): "IoT là một hạ tầng toàn cầu cho xã hội thông tin, cho phép các dịch vụ tiên tiến bằng cách liên kết các vật thể vật lý và ảo thông qua việc khai thác các công nghệ thông tin hiện có."

2.1.2 Cấu trúc tổng quát của một hệ thống IoT gồm 3 lớp chính

a. Lớp cảm biến (Perception Layer):

Lớp cảm biến (Perception Layer) là lớp thấp nhất trong kiến trúc hệ thống IoT, có nhiệm vụ trực tiếp thu thập các thông tin từ môi trường xung quanh như chuyển động, nhiệt độ, ánh sáng, âm thanh và các yếu tố vật lý khác. Dữ liệu thu thập được từ lớp này đóng vai trò đầu vào cho toàn bộ hệ thống, làm cơ sở cho quá trình truyền tải, xử lý và phân tích ở các lớp phía trên. Các thiết bị điển hình thuộc lớp cảm biến bao gồm cảm biến chuyển động PIR, cảm biến cửa, cảm biến khói và camera IP, giúp hệ thống giám sát và phát hiện kịp thời các sự kiện bất thường trong môi trường cần bảo vệ.

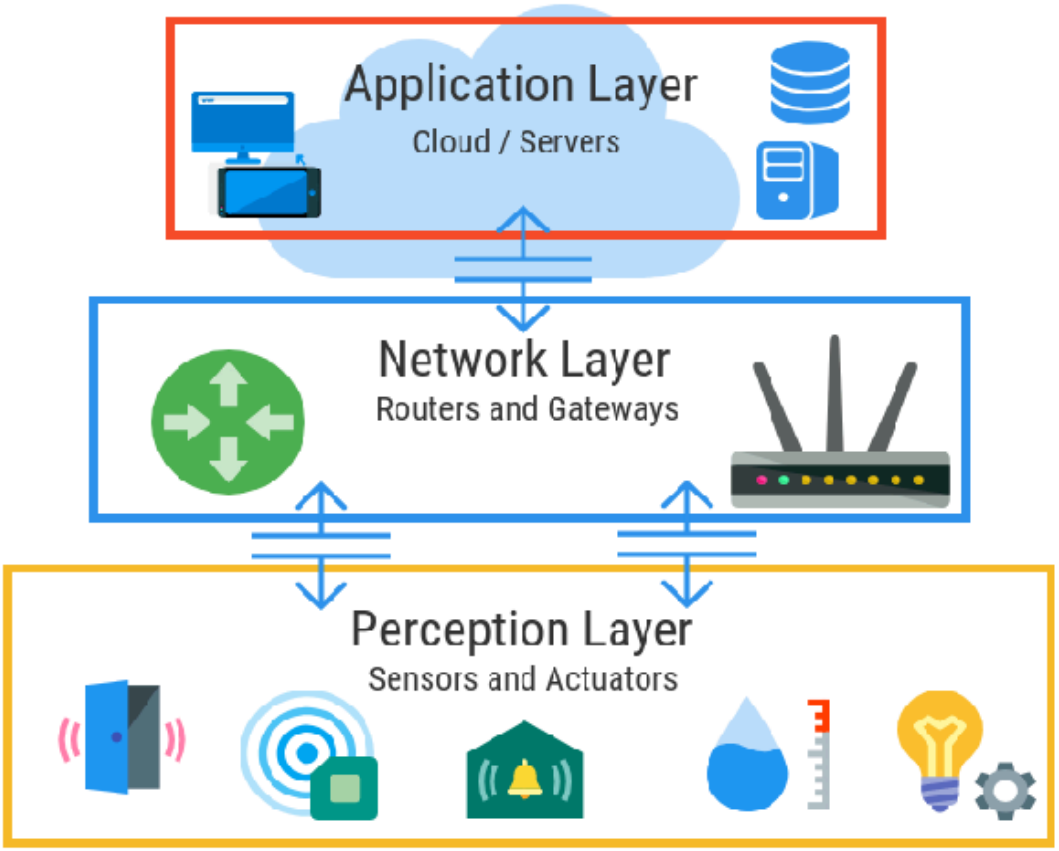
b. Lớp mạng (Network Layer):

Lớp mạng (Network Layer) có nhiệm vụ truyền dữ liệu thu thập được từ lớp cảm biến đến trung tâm xử lý thông qua các giao thức truyền thông khác nhau như Wi-Fi, ZigBee, MQTT, HTTP hoặc LoRa. Tùy vào yêu cầu của hệ thống và điều kiện triển khai, các giao thức này cho phép đảm bảo việc truyền dữ liệu được thực hiện ổn định, kịp thời và hiệu quả. Dữ liệu sau khi được truyền qua lớp mạng sẽ

được gửi đến server hoặc nền tảng Cloud để thực hiện các chức năng xử lý, phân tích và lưu trữ, phục vụ cho việc giám sát, cảnh báo và quản lý hệ thống IoT.

c. Lớp ứng dụng (Application Layer):

Lớp ứng dụng (Application Layer) là nơi người dùng trực tiếp tương tác với hệ thống IoT thông qua các giao diện web hoặc ứng dụng di động. Tại lớp này, các dữ liệu thu thập và xử lý từ những lớp bên dưới được hiển thị một cách trực quan, giúp người dùng theo dõi trạng thái hoạt động của hệ thống, nhận các thông báo cảnh báo khi có sự kiện bất thường, đồng thời thực hiện các thao tác điều khiển thiết bị từ xa nhằm đảm bảo an ninh và thuận tiện trong quá trình sử dụng.



Hình 2.1 : Mô hình IOT 3 lớp

2.1.3 Kiến trúc tổng quát của hệ thống IoT

Một hệ thống IoT hoàn chỉnh trong bảo vệ an ninh gia đình có thể được mô tả bằng công thức:

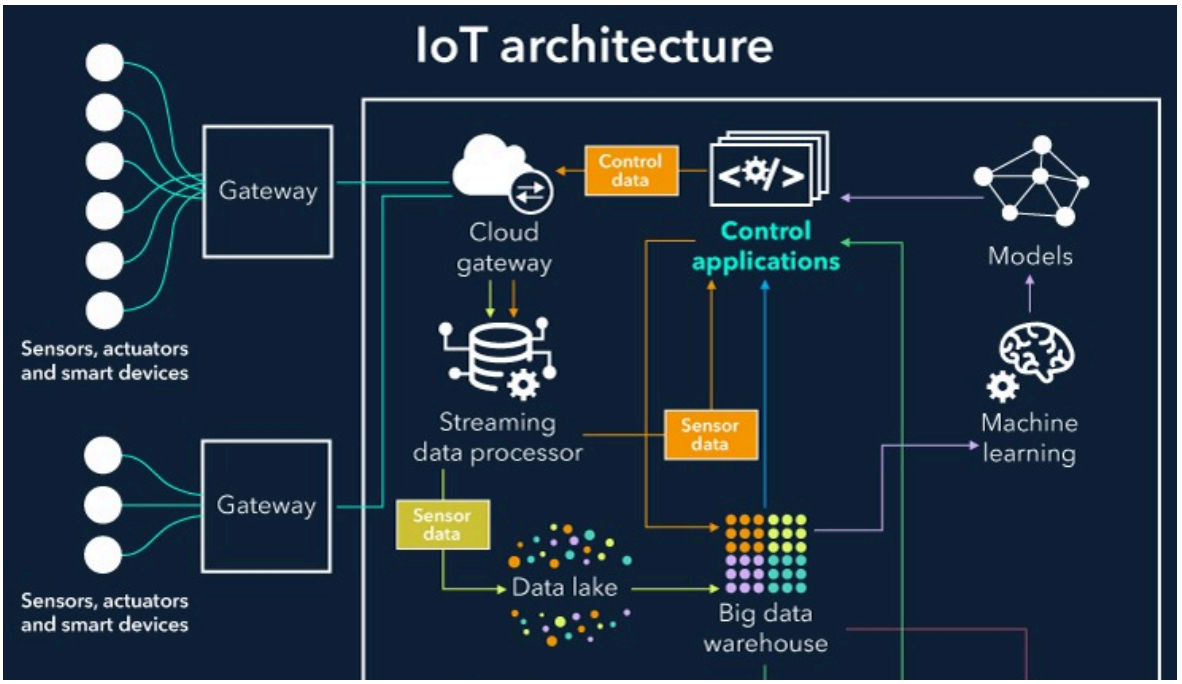
$$Hệthống = \{Thiết\ bị\ cảm\ biến,\ Mạng\ truyền\ thông,\ Ứng\ dụng\ giám\ sát\}$$

Trong đó:

Thiết bị cảm biến: phát hiện sự kiện bất thường như chuyển động, mở cửa, khói,...

Mạng truyền thông: truyền dữ liệu thời gian thực đến trung tâm xử lý.

Ứng dụng giám sát: hiển thị cảnh báo, lưu trữ dữ liệu, điều khiển thiết bị.

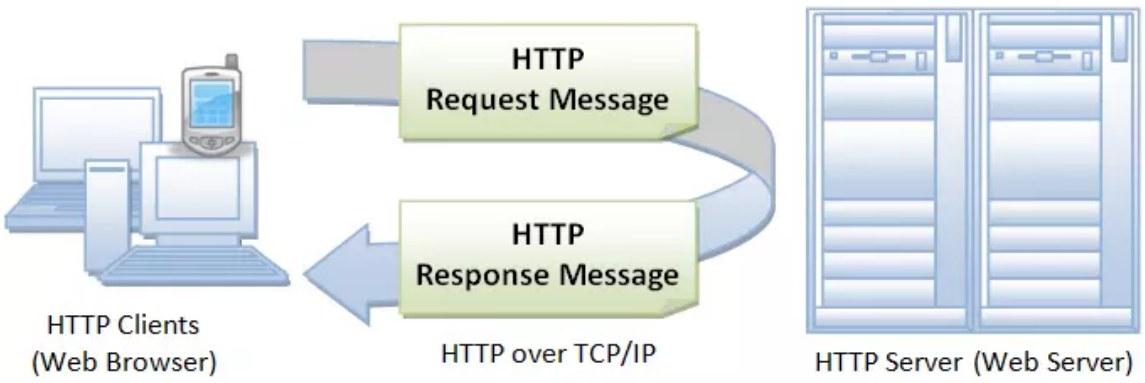


Hình 2.2 : Các khối chức năng chính và cách thức hoạt động của kiến trúc IOT

2.2 Mạng truyền thông trong IoT

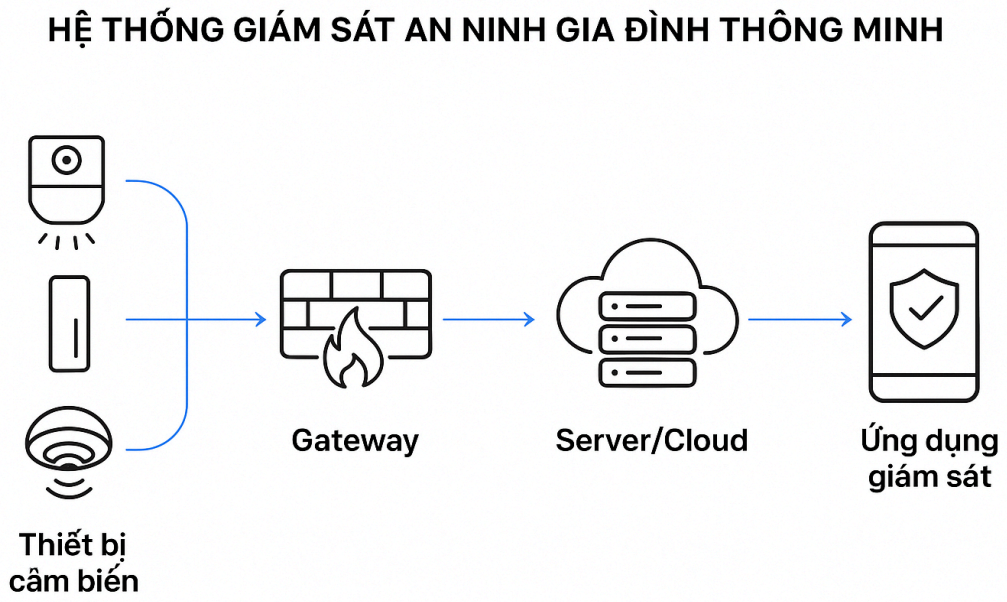
Trong hệ thống IoT bảo vệ an ninh gia đình, giao thức HTTP được sử dụng để truyền dữ liệu giữa thiết bị ESP32-CAM và nền tảng Cloud. Thông qua kết nối Wi-Fi, ESP32-CAM gửi hình ảnh, video và trạng thái hoạt động của hệ thống lên server bằng các yêu cầu HTTP.

Giao thức HTTP có ưu điểm là dễ triển khai, tương thích tốt với các dịch vụ Cloud và phù hợp với các hệ thống IoT quy mô nhỏ. Việc sử dụng HTTP + API Key giúp đảm bảo dữ liệu được truyền tải ổn định và có khả năng bảo mật trong quá trình giao tiếp giữa thiết bị và server.



Hình 2.3 : Mô phỏng giao thức HTTP

2.3 Hệ thống giám sát an ninh gia đình thông minh



Hình 2.4 : Minh họa kiến trúc hệ thống IoT an ninh

Hệ thống IoT bảo vệ an ninh gia đình bao gồm nhiều thành phần phối hợp chặt chẽ với nhau nhằm đảm bảo khả năng giám sát và phản ứng kịp thời trước các sự kiện bất thường. Trong đó, camera giám sát có nhiệm vụ ghi nhận hình ảnh và truyền dữ liệu về server để xử lý. Các cảm biến cửa và cảm biến chuyển động được sử dụng để phát hiện các sự kiện như mở cửa hoặc sự di chuyển của con người trong khu vực giám sát. Bộ xử lý trung tâm, đóng vai trò là Gateway hoặc Server, tiếp nhận dữ liệu từ các thiết bị, thực hiện phân tích và đưa ra các cảnh báo cần thiết. Thông tin cảnh báo và trạng thái hệ thống được hiển thị thông qua ứng dụng di động, cho phép người dùng theo dõi và điều khiển thiết bị từ xa. Ngoài ra, hệ thống còn có các thiết bị phản hồi như còi báo động hoặc khóa cửa thông minh nhằm tăng cường khả năng bảo vệ và phản ứng tự động khi phát hiện xâm nhập.

Công thức khái quát luồng dữ liệu:

$$D_{sensor} \rightarrow Gateway \rightarrow Server_{Cloud} \rightarrow App_{User}$$

Trong đó Dsensor là dữ liệu cảm biến, được xử lý và truyền lên hệ thống để đưa ra phản ứng (báo động, thông báo,...).

2.4 Bảo mật trong hệ thống IoT

2.4.1 Quản lý thông tin nhạy cảm trong hệ thống

a. Trên thiết bị ESP32

ESP32 cần lưu trữ một số thông tin quan trọng để đảm bảo khả năng kết nối và vận hành của hệ thống, bao gồm thông tin Wi-Fi dùng để kết nối vào mạng nội bộ, Blynk Auth Token nhằm xác thực thiết bị với nền tảng Blynk, và API Key được sử dụng để xác thực khi gửi dữ liệu, đặc biệt là hình ảnh, lên server xử lý. Hiện tại, các thông tin này được khai báo trực tiếp trong mã nguồn .ino. Cách triển khai này giúp hệ thống hoạt động ổn định, dễ cấu hình và thuận tiện cho quá trình phát triển, thử nghiệm trong môi trường học tập, đặc biệt khi hệ thống được vận hành trong mạng nội bộ có kiểm soát.

b. Trên server

Các thông tin nhạy cảm của hệ thống, bao gồm AWS Access Key ID, AWS Secret Access Key và API Key, đã được tách hoàn toàn khỏi mã nguồn chương trình và lưu trữ trong file cấu hình .env. Server sẽ đọc các giá trị này thông qua các biến môi trường trong quá trình chạy, qua đó tránh việc khai báo trực tiếp (hard-code) các thông tin quan trọng trong file JavaScript. Cách tiếp cận này giúp tăng cường mức độ bảo mật, hạn chế nguy cơ rò rỉ thông tin khi chia sẻ hoặc quản lý mã nguồn, đồng thời phù hợp với các khuyến nghị bảo mật trong phát triển ứng dụng backend.

2.4.2 Các biện pháp bảo mật đã được triển khai

a. Tách biệt thông tin nhạy cảm phía server

AWS Access Key/Secret và API Key không được khai báo trực tiếp trong mã nguồn server mà được quản lý thông qua các biến môi trường, giúp giảm thiểu nguy cơ lộ thông tin nhạy cảm. Đồng thời, file cấu hình .env đã được thêm vào .gitignore để đảm bảo các thông tin này không bị đẩy lên hệ thống quản lý mã nguồn, góp phần nâng cao mức độ an toàn và tuân thủ các nguyên tắc bảo mật trong quá trình phát triển và triển khai hệ thống.

b. Xác thực truy cập API bằng API Key

Mọi yêu cầu (request) được gửi từ ESP32 lên server đều phải kèm theo khóa xác thực x-api-key trong phần header. Server thực hiện kiểm tra API Key trước khi xử lý các endpoint quan trọng như chức năng phân tích hình ảnh hoặc truy vấn lịch sử sự kiện. Những request không hợp lệ hoặc thiếu thông tin xác thực sẽ bị từ chối ngay từ đầu, qua đó giúp ngăn chặn các truy cập trái phép và tăng cường mức độ bảo mật cho hệ thống.

c. Giới hạn tần suất truy cập (Rate Limiting)

Server được cấu hình để giới hạn số lượng request trong một khoảng thời gian nhất định thông qua cơ chế rate limiting. Biện pháp này giúp giảm nguy cơ các hành vi tấn công như spam, brute-force API hoặc việc gửi yêu cầu liên tục gây quá tải tài nguyên server, từ đó góp phần duy trì sự ổn định và an toàn trong quá trình vận hành hệ thống.

d. Giới hạn phạm vi triển khai

Trong quá trình thử nghiệm, ESP32 và WebServer được triển khai và hoạt động chủ yếu trong mạng nội bộ (LAN), đồng thời không mở cổng truy cập công khai trực tiếp ra Internet. Cách triển khai này giúp hạn chế phạm vi tiếp cận của hệ thống, qua đó giảm đáng kể nguy cơ bị tấn công từ bên ngoài và nâng cao mức độ an toàn trong môi trường thực nghiệm.

CHƯƠNG 3. MÔ HÌNH ĐỀ XUẤT

3.1 Cơ sở đề xuất mô hình

Trong bối cảnh nhu cầu bảo vệ an ninh gia đình ngày càng gia tăng, việc ứng dụng công nghệ Internet of Things (IoT) vào hệ thống giám sát là một hướng đi phù hợp và có tính thực tiễn cao. Các hệ thống camera truyền thống thường có chi phí cao, khó mở rộng và hạn chế trong khả năng giám sát từ xa. Do đó, việc đề xuất một mô hình giám sát an ninh dựa trên IoT giúp khắc phục những hạn chế trên, đồng thời nâng cao hiệu quả bảo vệ an ninh.

ESP32-CAM được lựa chọn làm thiết bị trung tâm của hệ thống nhờ tích hợp sẵn camera và khả năng kết nối Wi-Fi, cùng với ưu điểm về giá thành thấp, kích thước nhỏ gọn và dễ lập trình. Thiết bị có khả năng thu thập hình ảnh từ khu vực giám sát và truyền dữ liệu về server thông qua mạng Internet để phục vụ cho quá trình xử lý và phân tích. Hệ thống được xây dựng theo mô hình Client – Server, trong đó ESP32-CAM đóng vai trò là client thực hiện việc thu thập và gửi dữ liệu, server đảm nhiệm các chức năng xử lý, lưu trữ và phát sinh cảnh báo, còn người dùng truy cập và giám sát hệ thống thông qua giao diện web hoặc ứng dụng trên thiết bị di động. Mô hình này giúp hệ thống hoạt động ổn định, dễ quản lý và thuận tiện cho việc mở rộng trong tương lai.

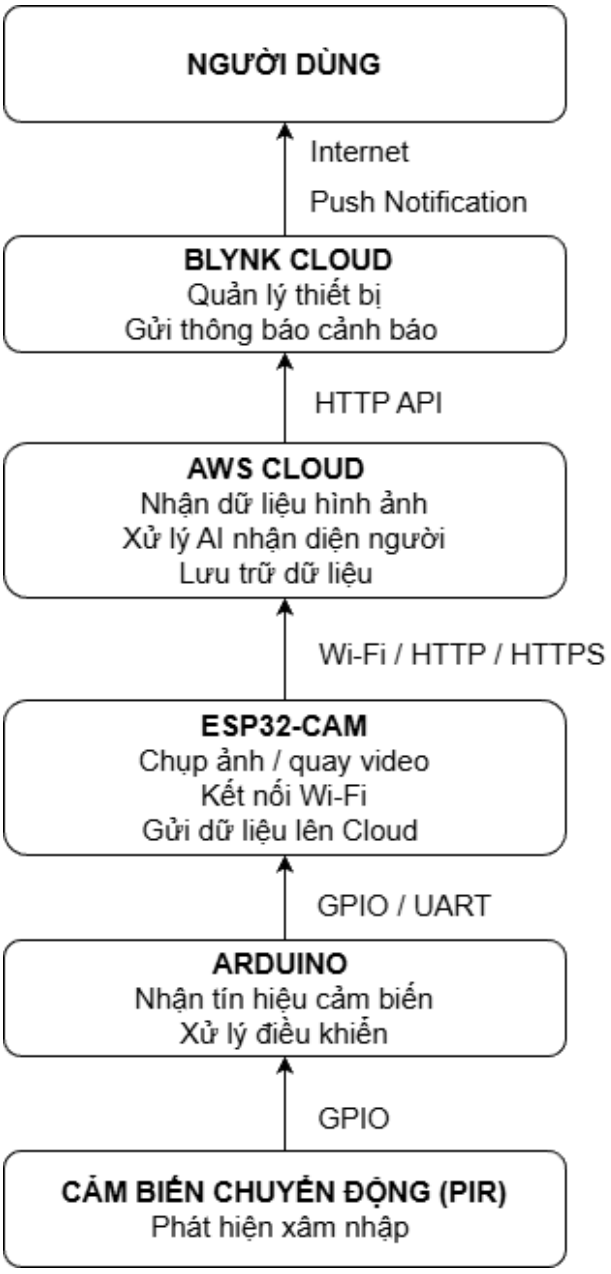
Từ các phân tích trên, mô hình IoT bảo vệ an ninh gia đình sử dụng ESP32-CAM là phù hợp về mặt công nghệ, có tính khả thi cao và đáp ứng tốt mục tiêu nghiên cứu của đề tài.

3.2 Mục tiêu

Mục tiêu của đề tài “Phát triển ứng dụng IoT để bảo vệ an ninh gia đình” là nghiên cứu, thiết kế và xây dựng một hệ thống giám sát an ninh gia đình dựa trên nền tảng Internet of Things. Hệ thống cho phép thu thập hình ảnh từ camera ESP32-CAM, truyền dữ liệu qua mạng Internet, giám sát từ xa theo thời gian thực và gửi cảnh báo khi phát hiện các hành vi xâm nhập bất thường.

Bên cạnh đó, đề tài hướng đến việc áp dụng các kiến thức về IoT, mạng máy tính và lập trình vào thực tế, xây dựng hệ thống có chi phí thấp, dễ triển khai và có khả năng mở rộng, đáp ứng yêu cầu học tập và nghiên cứu của sinh viên ngành Công nghệ thông tin.

3.3 Kiến trúc tổng thể



Hình 3.1 : Overall System Architecture Diagram

3.3.1 Lớp thiết bị (Device Layer)

Lớp thiết bị bao gồm các phần cứng IoT trong đó thiết bị chính là:

ESP32-CAM tích hợp camera OV2640 có nhiệm vụ thu thập hình ảnh hoặc video từ môi trường giám sát. Thiết bị hoạt động liên tục, ghi nhận các thay đổi trong khu vực giám sát và có thể thực hiện các xử lý cơ bản như phát hiện chuyển động ban đầu.

Hình 3.2 : ESP32-CAM

Arduino là bo mạch vi điều khiển được sử dụng để điều khiển và xử lý tín hiệu từ các cảm biến trong hệ thống. Arduino có nhiệm vụ nhận dữ liệu từ cảm biến chuyển động, xử lý tín hiệu và thực hiện các hành động tương ứng như gửi tín hiệu điều khiển hoặc kích hoạt cảnh báo.

Hình 3.3 : Arduino

Cảm biến chuyển động (thường sử dụng cảm biến PIR) có chức năng phát hiện sự thay đổi chuyển động trong khu vực giám sát. Khi có chuyển động của con người hoặc vật thể, cảm biến sẽ phát ra tín hiệu gửi về Arduino để xử lý.

Hình 3.4 : Cảm biến chuyển động

Dây jumper được sử dụng để kết nối các linh kiện phần cứng với nhau như Arduino và cảm biến chuyển động. Các loại dây jumper gồm dây đực–đực, đực–cái và cái–cái, giúp việc lắp ráp mạch điện trở nên linh hoạt và thuận tiện trong quá trình thử nghiệm và triển khai hệ thống.

Hình 3.5 : Dây jumper

Buzzer là một linh kiện điện tử dùng để phát ra âm thanh cảnh báo hoặc thông báo trong các mạch điện tử. Buzzer thường được sử dụng trong các dự án Arduino, ESP32, hệ thống báo động, thiết bị gia dụng và mạch điều khiển để tạo tín hiệu âm thanh như tiếng “beep” khi có sự kiện xảy ra. Có hai loại buzzer phổ biến là buzzer chủ động (active buzzer) và buzzer thụ động (passive buzzer), giúp người dùng linh hoạt lựa chọn theo nhu cầu sử dụng. Nhờ kích thước nhỏ gọn, dễ kết nối và lập trình, buzzer là linh kiện không thể thiếu trong quá trình thử nghiệm và triển khai các hệ thống điện tử.

Hình 3.6 : Còi buzzer

3.3.2 Lớp mạng truyền thông (Network Layer)

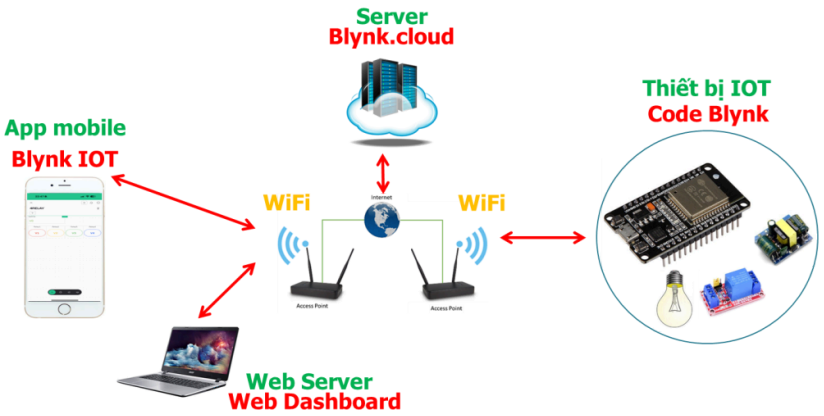
Lớp mạng truyền thông đóng vai trò kết nối giữa các thiết bị phần cứng trong hệ thống IoT và các thành phần xử lý phía trên. Lớp này đảm bảo việc truyền dữ liệu từ thiết bị giám sát đến server và ứng dụng người dùng được diễn ra ổn định, kịp thời và an toàn.

Trong hệ thống của đề tài, lớp mạng truyền thông sử dụng mạng Wi-Fi làm hạ tầng kết nối chính để đảm bảo khả năng truyền dữ liệu ổn định và thuận tiện trong môi trường gia đình. ESP32-CAM được tích hợp sẵn module Wi-Fi, cho phép kết nối trực tiếp với router gia đình và truy cập Internet. Thông qua kết nối này, các dữ liệu hình ảnh, tín hiệu phát hiện chuyển động và trạng thái hoạt động của thiết bị được truyền từ ESP32-CAM đến server xử lý trung tâm. Bên cạnh đó, Arduino nhận tín hiệu từ cảm biến chuyển động và thực hiện xử lý cục bộ, sau đó truyền thông tin này sang ESP32-CAM thông qua kết nối dây hoặc giao tiếp nối tiếp. ESP32-CAM tiếp tục đảm nhiệm vai trò gửi các dữ liệu đã tổng hợp lên server thông qua mạng truyền thông để phục vụ cho quá trình giám sát và phân tích.

Các giao thức truyền thông phổ biến như HTTP được sử dụng để truyền dữ liệu giữa ESP32-CAM và server. Giao thức này dễ triển khai, phù hợp với các hệ thống IoT quy mô nhỏ và đáp ứng tốt yêu cầu giám sát theo thời gian thực.

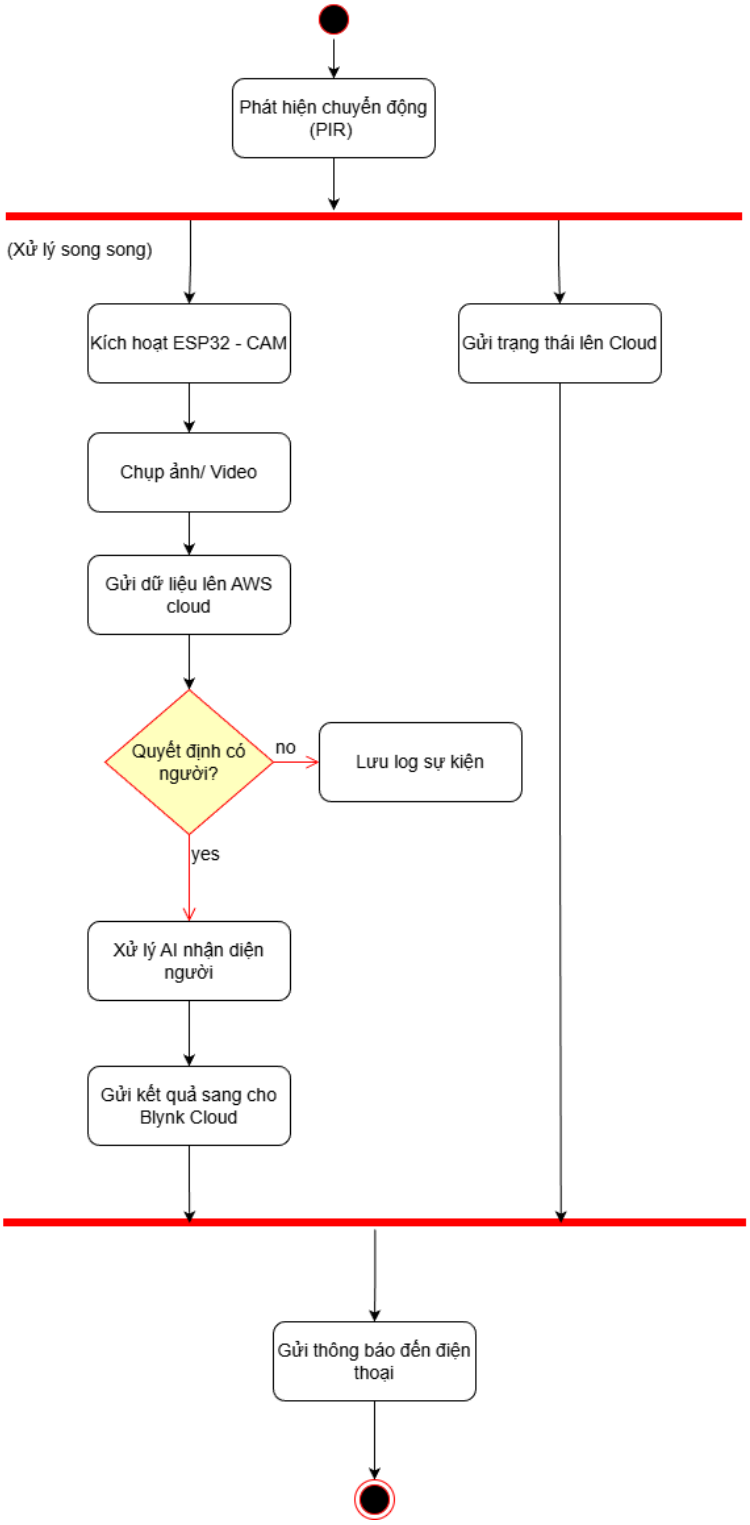
3.3.3 Lớp ứng dụng người dùng (Application Layer)

Blynk đóng vai trò là lớp Application Layer trong hệ thống, cho phép người dùng theo dõi trạng thái của các thiết bị IoT, nhận thông báo cảnh báo và điều khiển hệ thống thông qua ứng dụng di động có sẵn. Cụ thể, dữ liệu từ các thiết bị IoT như ESP32-CAM, cảm biến chuyển động và cảm biến cửa sau khi được xử lý tại Gateway và trên nền tảng Cloud sẽ được gửi lên Blynk Cloud. Nền tảng này chịu trách nhiệm hiển thị thông tin trạng thái hệ thống, các hình ảnh hoặc sự kiện giám sát, đồng thời phát hiện và thông báo các hành vi bất thường đến người dùng theo thời gian thực. Thông qua ứng dụng Blynk trên điện thoại thông minh, người dùng có thể nhận thông báo tức thì khi phát hiện chuyển động hoặc xâm nhập trái phép, theo dõi trạng thái hoạt động của các thiết bị trong hệ thống, bật/tắt hoặc cấu hình các chức năng giám sát cơ bản, cũng như giám sát toàn bộ hệ thống từ xa thông qua kết nối Internet.



Hình 3.7 : Mô phỏng cách hoạt động của Blynk

3.4 Mô hình hoạt động



Hình 3.8 : Activity Diagram

Hệ thống IoT bảo vệ an ninh gia đình được thiết kế dựa trên mô hình xử lý sự kiện kết hợp với xử lý song song nhằm đảm bảo khả năng phản hồi nhanh và hoạt động ổn định. Khi cảm biến chuyển động PIR phát hiện có sự di chuyển trong khu vực giám sát, tín hiệu sẽ được gửi đến bộ điều khiển trung tâm để kích hoạt hệ thống. Tại thời điểm này, hệ thống triển khai hai luồng xử lý song song. Ở luồng xử lý thứ nhất, ESP32-CAM được kích hoạt để chụp ảnh hoặc quay video tại khu vực xảy ra sự kiện, sau đó dữ liệu hình ảnh hoặc video được truyền lên nền tảng AWS Cloud thông qua kết nối Wi-Fi sử dụng giao thức HTTP/HTTPS. Tại Cloud, hệ thống tiến hành phân tích dữ liệu để xác định sự xuất hiện của con người; nếu không phát hiện người, thông tin sự kiện chỉ được lưu lại nhằm phục vụ mục đích giám sát và thống kê, trong khi nếu phát hiện có người, dữ liệu sẽ được xử lý bởi mô-đun AI nhận diện và kết quả phân tích được gửi sang Blynk Cloud. Song song với đó, ở luồng xử lý thứ hai, trạng thái hoạt động của thiết bị được gửi lên Cloud để cập nhật tình trạng hệ thống theo thời gian thực. Sau khi Blynk Cloud nhận được kết quả xử lý, hệ thống sẽ gửi thông báo cảnh báo đến điện thoại của người dùng thông qua ứng dụng di động, giúp người dùng kịp thời nắm bắt và phản ứng trước các tình huống an ninh xảy ra tại khu vực giám sát.

3.5 Ưu điểm của mô hình đề xuất

Mô hình đề xuất có kiến trúc đơn giản, dễ triển khai và thuận tiện cho việc bảo trì trong quá trình vận hành. Hệ thống có chi phí thấp, phù hợp với điều kiện và nhu cầu sử dụng trong phạm vi hộ gia đình. Nhờ cơ chế xử lý sự kiện kết hợp xử lý song song, hệ thống có khả năng phản hồi nhanh khi phát hiện các tình huống bất thường, đồng thời đảm bảo hoạt động ổn định. Bên cạnh đó, mô hình có tính linh hoạt cao, dễ mở rộng và thuận lợi cho việc tích hợp các chức năng AI trên nền tảng Cloud nhằm nâng cao khả năng phân tích và nhận diện. Ngoài ra, hệ thống hỗ trợ giám sát từ xa và gửi cảnh báo kịp thời đến người dùng, giúp tăng cường hiệu quả bảo vệ an ninh trong môi trường gia đình.

3.6 Mô hình triển khai thực tế

Trong mô hình triển khai thực tế, ESP32-CAM được lắp đặt tại các vị trí cần giám sát như cửa ra vào, hành lang hoặc khu vực sân trước nhằm đảm bảo khả năng quan sát hiệu quả. Cảm biến chuyển động PIR được bố trí gần khu vực giám sát để kịp thời phát hiện sự di chuyển của con người hoặc vật thể. Các thiết bị trong hệ thống được kết nối với nhau thông qua dây jumper và sử dụng nguồn điện ổn định để đảm bảo hoạt động liên tục. ESP32-CAM kết nối trực tiếp với mạng Wi-Fi gia đình để truyền dữ liệu lên nền tảng Cloud, nơi dữ liệu hình ảnh và trạng thái hoạt động của hệ thống được xử lý và lưu trữ. Người dùng có thể theo dõi tình trạng hệ thống và nhận các thông báo cảnh báo thông qua ứng dụng trên điện thoại thông minh, giúp giám sát an ninh gia đình một cách thuận tiện và kịp thời.

Hình 3.9 : Mô hình triển khai thực tế

CHƯƠNG 4. THỰC NGHIỆM

4.1 Dữ liệu thực nghiệm

4.1.1 Mục tiêu

Dữ liệu hoạt động thực tế của hệ thống được ghi nhận thông qua quá trình vận hành liên tục trong các điều kiện sử dụng bình thường, bao gồm trạng thái kết nối, thời gian xử lý sự kiện và phản hồi cảnh báo. Kết quả cho thấy hệ thống hoạt động ổn định, không xảy ra lỗi treo hay gián đoạn trong quá trình giám sát. Độ trễ từ thời điểm phát hiện sự kiện đến khi gửi thông báo ở mức thấp, đáp ứng yêu cầu giám sát thời gian gần thực. Khả năng phản hồi của hệ thống nhanh và nhất quán, đảm bảo người dùng nhận được thông tin kịp thời khi có sự kiện xảy ra.

4.1.2 Loại dữ liệu thu thập

Hệ thống ghi nhận trạng thái kết nối ổn định thông qua mạng Wi-Fi và nền tảng Blynk, đảm bảo việc truyền dữ liệu không bị gián đoạn trong quá trình hoạt động. Dữ liệu hình ảnh từ camera được truyền dưới dạng luồng thời gian thực, cho phép người dùng theo dõi trực tiếp khu vực giám sát. Khi có truy cập hoặc phát hiện chuyển động, dữ liệu sự kiện được xử lý và gửi thông báo kịp thời đến người dùng. Thời gian phản hồi được xác định từ thời điểm sự kiện xảy ra đến khi cảnh báo được nhận, cho thấy hệ thống đáp ứng nhanh, phù hợp với yêu cầu giám sát an ninh trong thời gian thực.

4.1.3 Kịch bản thực nghiệm

Quá trình vận hành của hệ thống bắt đầu khi ESP32 khởi động và tự động kết nối vào mạng Wi-Fi đã cấu hình. Sau khi kết nối thành công, người dùng có thể truy cập WebServer để theo dõi hình ảnh camera giám sát theo thời gian thực. Khi hệ thống phát sinh sự kiện giám sát như phát hiện chuyển động hoặc truy cập bất thường, dữ liệu được xử lý và gửi đến nền tảng Blynk để tạo thông báo cảnh báo trên điện thoại người dùng. Trong quá trình thử nghiệm, việc ngắt kết nối Wi-Fi và

kết nối lại cho thấy hệ thống có khả năng tự khôi phục kết nối, tiếp tục hoạt động bình thường mà không cần can thiệp thủ công.

4.2 Cài đặt thực nghiệm

4.2.1 Mục tiêu

Quy trình triển khai được thực hiện bằng cách thiết lập đầy đủ môi trường phần cứng và phần mềm theo đúng cấu hình thiết kế, bao gồm kết nối các thiết bị và chuẩn bị nền tảng vận hành. Sau đó, chương trình được nạp cho ESP32 và tiến hành cấu hình các dịch vụ liên quan như Wi-Fi, WebServer và Blynk để đảm bảo khả năng truyền dữ liệu và gửi cảnh báo. Cuối cùng, hệ thống được kiểm tra tổng thể nhằm xác nhận trạng thái hoạt động ổn định, từ đó đảm bảo sẵn sàng thu thập dữ liệu thực nghiệm trong quá trình vận hành.

4.2.2 Môi trường cài đặt

a. Phần cứng

Hệ thống phần cứng bao gồm ESP32 hoặc ESP32-CAM đóng vai trò là bộ điều khiển trung tâm và thiết bị thu thập hình ảnh. Router Wi-Fi nội bộ được sử dụng để tạo mạng kết nối giữa các thiết bị trong hệ thống. Máy tính cá nhân cài đặt Arduino IDE đảm nhiệm việc nạp chương trình và cấu hình cho ESP32. Điện thoại thông minh được cài đặt ứng dụng Blynk dùng để giám sát trạng thái hệ thống và nhận thông báo cảnh báo khi có sự kiện xảy ra.

b. Phần mềm

Môi trường phần mềm được xây dựng trên Arduino IDE, cho phép lập trình, biên dịch và nạp mã nguồn cho ESP32. Các thư viện WiFi, WebServer và Camera được sử dụng để thiết lập kết nối mạng, xây dựng máy chủ web và xử lý luồng hình ảnh từ camera. Nền tảng Blynk được cấu hình thông qua dashboard và chức năng notification, giúp hiển thị dữ liệu giám sát và gửi cảnh báo thời gian thực đến điện thoại người dùng.

4.2.3 Quy trình cài đặt

a. Arduino IDE và ESP32

Môi trường phát triển cho ESP32 được thiết lập bằng cách cài đặt Arduino IDE phiên bản 2.x nhằm đảm bảo tương thích với các board và thư viện mới. Sau đó, URL của ESP32 Board Manager được thêm vào mục Preferences để mở rộng danh sách phần cứng hỗ trợ. Board "esp32 by Espressif Systems" được cài đặt để cho phép biên dịch và nạp chương trình cho ESP32/ESP32-CAM. Các thư viện cần thiết gồm Blynk, ArduinoJson và WiFiManager được cài đặt nhằm phục vụ kết nối IoT, xử lý dữ liệu JSON và quản lý cấu hình Wi-Fi động cho thiết bị.

b. Node.js Server

Máy chủ backend được triển khai trên nền tảng Node.js với phiên bản LTS từ v18.x trở lên để đảm bảo tính ổn định và bảo mật. Một thư mục dự án riêng được tạo và khởi tạo bằng npm để quản lý gói phụ thuộc. Các thư viện express, multer, aws-sdk, cors, dotenv và express-rate-limit được cài đặt nhằm xây dựng API, xử lý upload dữ liệu, kết nối dịch vụ AWS, hỗ trợ CORS, quản lý biến môi trường và giới hạn truy cập, đảm bảo hệ thống hoạt động an toàn và hiệu quả.

c. AWS Services

Hệ thống sử dụng các dịch vụ của AWS bằng cách tạo tài khoản AWS và thiết lập một IAM User với quyền truy cập S3 và Rekognition. Một S3 Bucket được tạo tại khu vực ap-southeast-2 để lưu trữ dữ liệu hình ảnh từ hệ thống giám sát. Access Key ID và Secret Access Key được lưu trữ an toàn và sử dụng cho việc cấu hình kết nối giữa Node.js Server và các dịch vụ AWS, phục vụ cho quá trình lưu trữ và xử lý dữ liệu trên nền tảng cloud.

d. Blynk IoT

Nền tảng Blynk được cấu hình bằng cách đăng ký tài khoản và tạo một Template mang tên "ESP32CAM Security" để quản lý thiết bị IoT. Trong template, hệ thống thiết lập 9 Virtual Pins (V0–V8) dùng cho việc hiển thị dữ liệu và điều khiển, đồng

thời cấu hình 3 Events để gửi thông báo khi có sự kiện giám sát. Auth Token được lấy từ Device và tích hợp vào chương trình ESP32, cho phép thiết bị kết nối với Blynk Cloud và gửi dữ liệu, cảnh báo đến điện thoại người dùng theo thời gian thực.

4.2.4 Cấu hình hệ thống

a. Cấu hình Server

Tiến hành tạo file server.js để xây dựng dịch vụ backend, trong đó khai báo AWS credentials phục vụ kết nối các dịch vụ liên quan. Tạo file .env chứa thông tin AWS IAM và thiết lập các giá trị này dưới dạng Environment Variables của hệ điều hành để tăng mức độ bảo mật. Khai báo tên S3 Bucket theo biến S3_BUCKET=esp32cam-security-khoahanh, đồng thời tạo API Key ngẫu nhiên bằng PowerShell với lệnh -join ((48..57) + (65..90) + (97..122) | Get-Random -Count 32 | ForEach-Object {[char]$_}) và lưu vào .env để xác thực truy cập API. Tiếp theo, thiết lập danh sách IP được phép truy cập bằng ALLOWED_IPS=192.168.100.167,192.168.100.114 nhằm giới hạn nguồn truy cập từ camera và laptop cá nhân. Dự án được khởi tạo bằng cách tạo file package.json, cài đặt các dependencies bằng npm install, tạo file .gitignore để loại trừ các thông tin nhạy cảm khi quản lý mã nguồn, và tạo thư mục uploads để lưu tạm dữ liệu tải lên trước khi xử lý.

b. Cấu hình thiết bị ESP32

Trên ESP32, cấu hình Wi-Fi được thực hiện tự động thông qua WiFiManager, giúp thiết bị kết nối mạng mà không cần hardcode SSID/password trong mã nguồn. Sau đó, khai báo Blynk Auth Token để thiết bị liên kết với Blynk Cloud và thực hiện chức năng gửi dữ liệu hoặc cảnh báo. API Key trên ESP32 phải được khai báo trùng khớp với giá trị API_Key trong file .env của server để đảm bảo xác thực hợp lệ khi giao tiếp giữa thiết bị và backend. Cuối cùng, kiểm tra đúng địa chỉ IP của server và cổng kết nối đang sử dụng để đảm bảo ESP32 có thể gửi dữ liệu và nhận phản hồi ổn định trong quá trình vận hành.

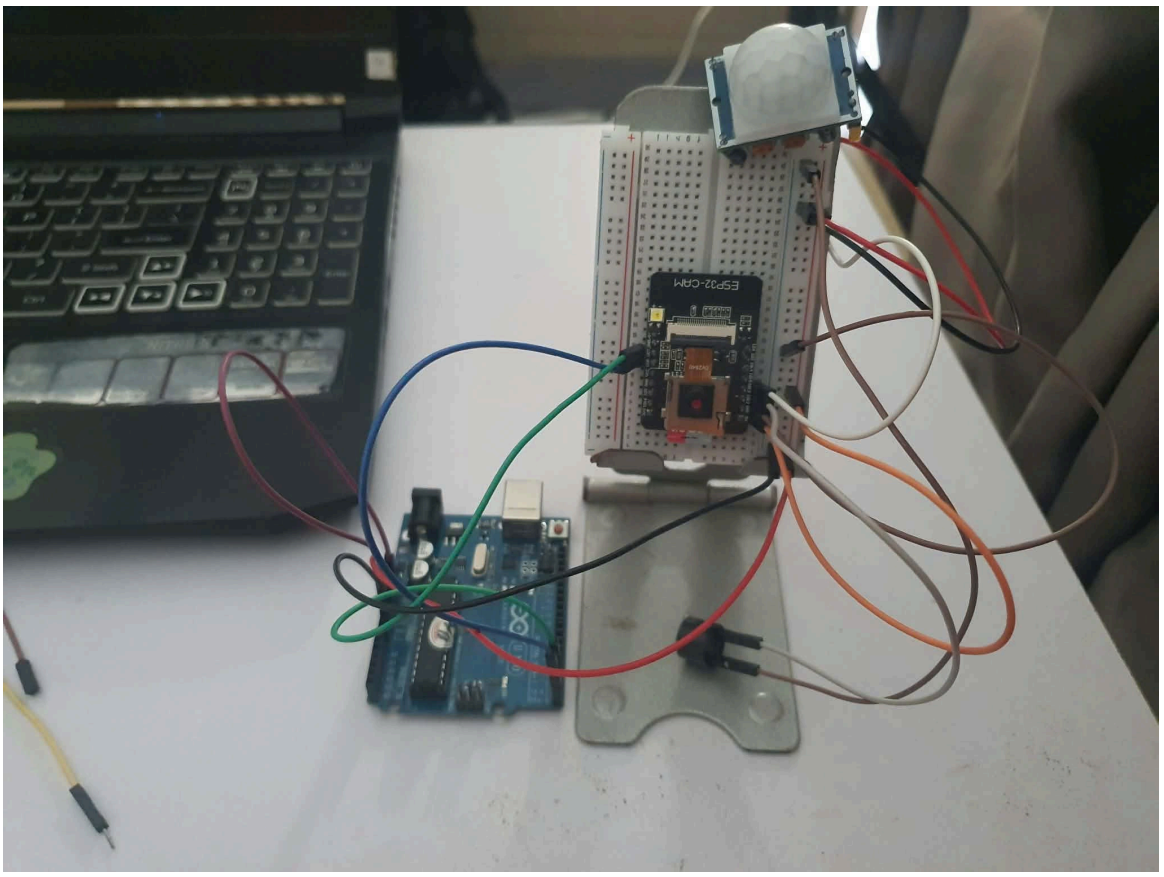
4.2.4.1 Kết nối phần cứng

Kết nối PIR Sensor được thực hiện theo đúng chân tín hiệu: PIR VCC nối vào ESP32 5V, PIR GND nối vào ESP32 GND và PIR OUT nối vào ESP32 GPIO13 để đọc tín hiệu phát hiện chuyển động. Buzzer được đấu với chân điều khiển để phát âm báo khi có sự kiện: Buzzer (+) nối vào ESP32 GPIO12 và Buzzer (-) nối vào ESP32 GND. Thẻ SD được chuẩn bị bằng cách format định dạng FAT32 nhằm đảm bảo tương thích khi lưu trữ dữ liệu hình ảnh hoặc log sự kiện trong quá trình hệ thống hoạt động.

4.3 Kết quả thực nghiệm

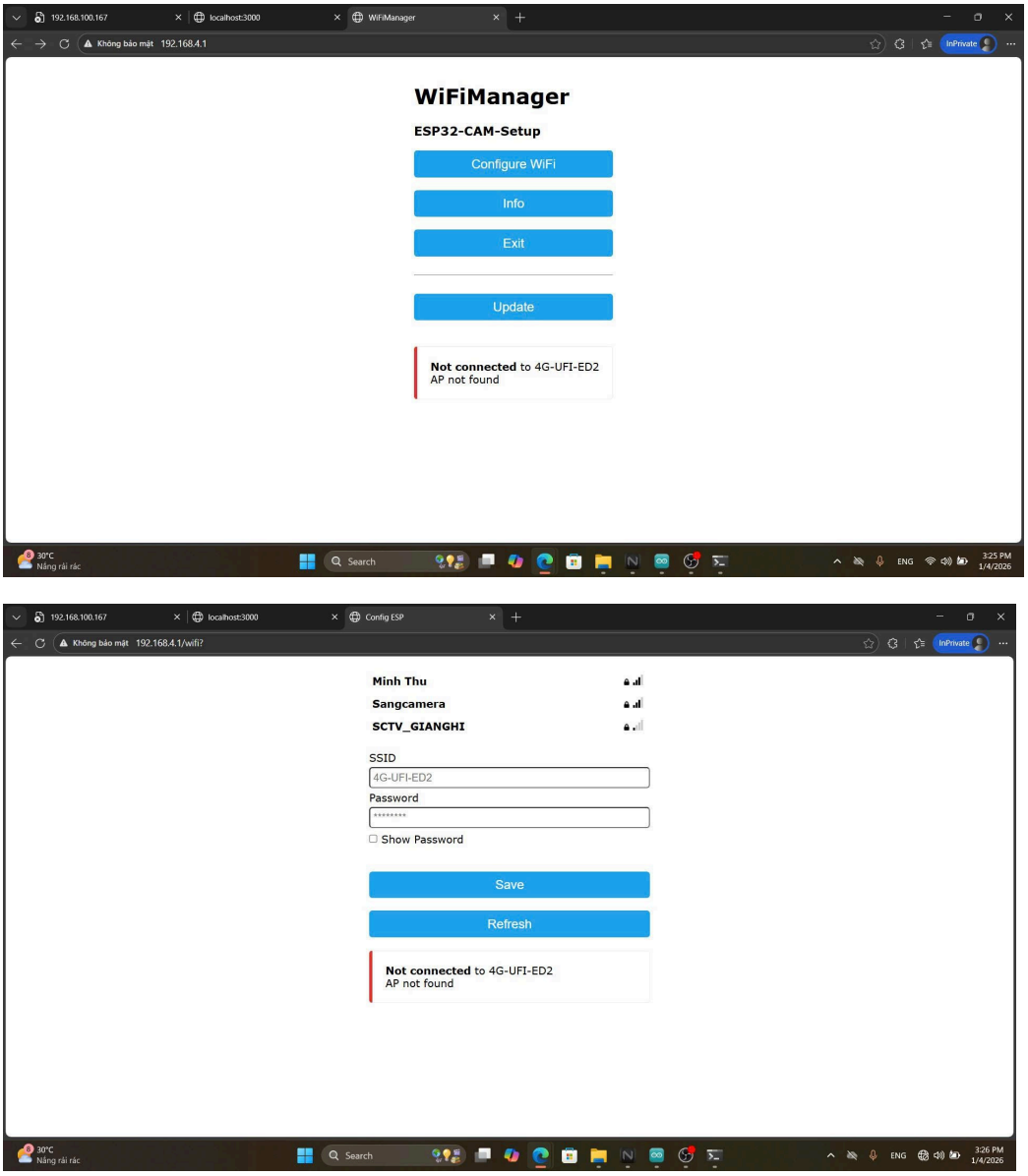
4.3.1 Kết quả kết nối và vận hành hệ thống

Trong quá trình thực nghiệm, hệ thống được khởi động và vận hành liên tục trong môi trường mạng LAN gia đình. Hình ảnh thiết bị thực tế được trình bày nhằm minh họa mô hình triển khai, bao gồm ESP32-CAM, cảm biến và hệ thống kết nối mạng trong điều kiện sử dụng thực tế. Kết quả cho thấy ESP32-CAM duy trì kết nối Wi-Fi ổn định, thời gian kết nối ban đầu ngắn và hầu như không xảy ra hiện tượng mất mạng. Khi xảy ra gián đoạn Wi-Fi, thiết bị có khả năng tự động kết nối lại mà không cần can thiệp thủ công. Server Node.js hoạt động ổn định trong suốt quá trình thử nghiệm, không ghi nhận tình trạng treo hoặc dừng dịch vụ. Kết nối giữa ESP32-CAM và server được duy trì liên tục thông qua các yêu cầu HTTP có xác thực API Key, đảm bảo tính liên tục và độ tin cậy của hệ thống. Kết quả này chứng minh mô hình triển khai đáp ứng tốt yêu cầu vận hành trong môi trường thực tế với quy mô gia đình.



Hình 4.1 : Thiết bị ESP32-CAM và các thành phần hệ thống trong quá trình thực nghiệm tại môi trường mạng LAN gia đình.

ESP32 kết nối Wi-Fi ổn định trong môi trường LAN.



Hình 4.2 : Giao diện kết nối Wi-fi

4.3.2 Kích hoạt phát hiện chuyển động và ghi nhận hình ảnh, video

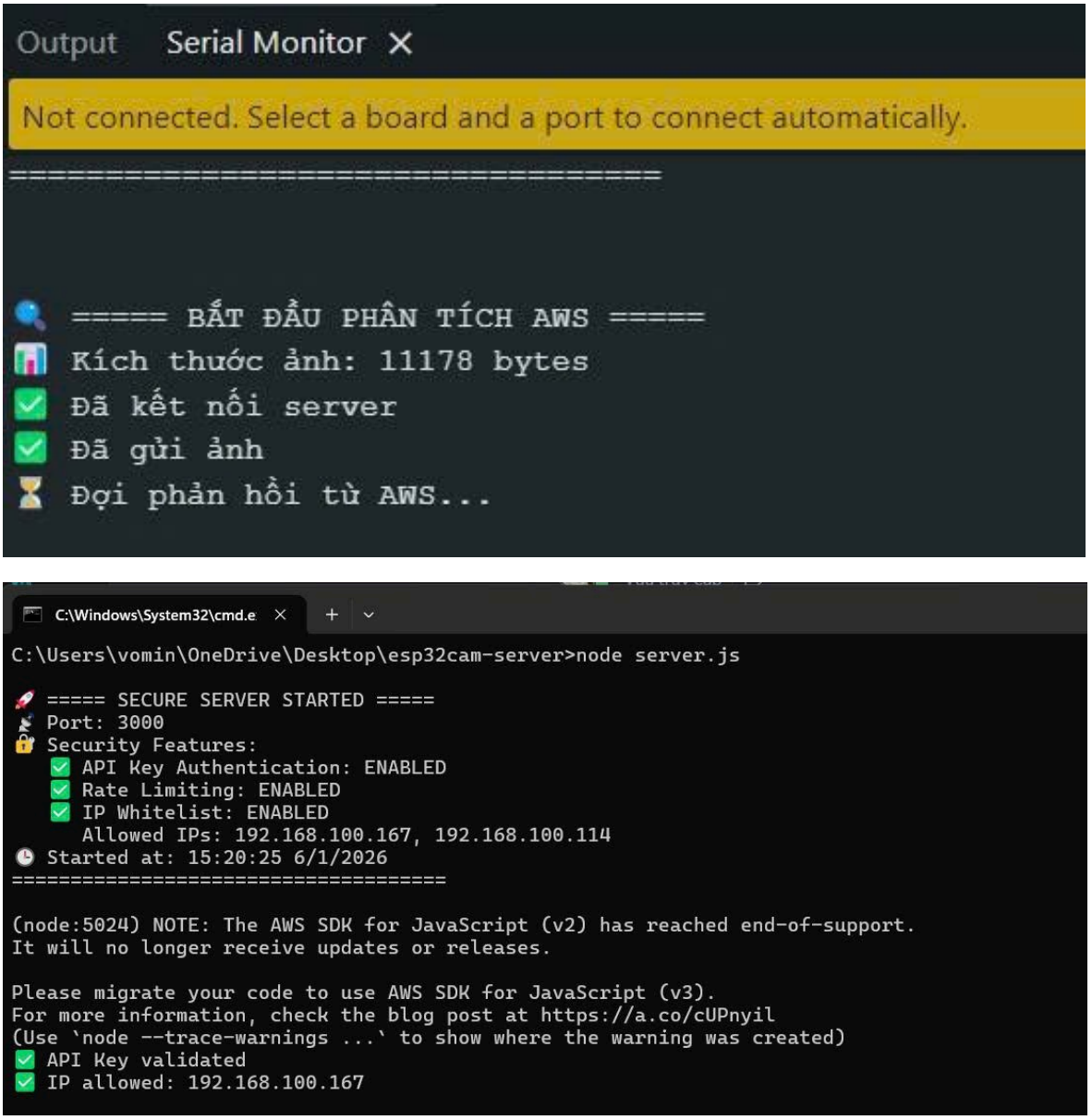
Khi cảm biến chuyển động PIR phát hiện sự di chuyển trong khu vực giám sát, tín hiệu kích hoạt được gửi đến bộ điều khiển trung tâm. Hệ thống tiến hành kiểm tra và xác nhận trạng thái chuyển động hợp lệ nhằm hạn chế các kích hoạt giả. Sau khi chuyển động được xác nhận, ESP32-CAM tự động khởi động quá trình ghi nhận dữ liệu, bao gồm chụp ảnh tức thời để phục vụ phân tích và đồng thời ghi video trong khoảng thời gian xác định. Các dữ liệu hình ảnh và video sau khi được

tạo sẽ được lưu trữ và sử dụng cho các bước xử lý, phân tích và giám sát tiếp theo của hệ thống. Kết quả thực nghiệm cho thấy cơ chế kích hoạt này hoạt động ổn định và đáp ứng nhanh khi có sự kiện chuyển động xảy ra.

Hình 4.3 : Nhật ký Serial Monitor khi cảm biến PIR phát hiện chuyển động và kích hoạt chụp ảnh, ghi video

4.3.3 Thiết lập kết nối HTTP và xác thực yêu cầu từ thiết bị IoT

Khi ESP32-CAM hoàn tất việc thu thập dữ liệu hình ảnh, thiết bị tạo một yêu cầu HTTP POST gửi dữ liệu lên server xử lý trung tâm. Trước khi tiếp nhận và xử lý nội dung yêu cầu, server thực hiện các bước kiểm tra bảo mật nhằm đảm bảo tính hợp lệ của kết nối. Cụ thể, server tiến hành xác thực API Key được gửi kèm trong header của request, kiểm tra địa chỉ IP nguồn dựa trên IP Whitelist, đồng thời áp dụng cơ chế Rate Limiting để giới hạn số lượng yêu cầu trong một khoảng thời gian nhất định. Chỉ khi tất cả các điều kiện bảo mật được đáp ứng, server mới chấp nhận request và tiếp tục xử lý dữ liệu, qua đó đảm bảo an toàn và ổn định cho hệ thống trong quá trình vận hành.

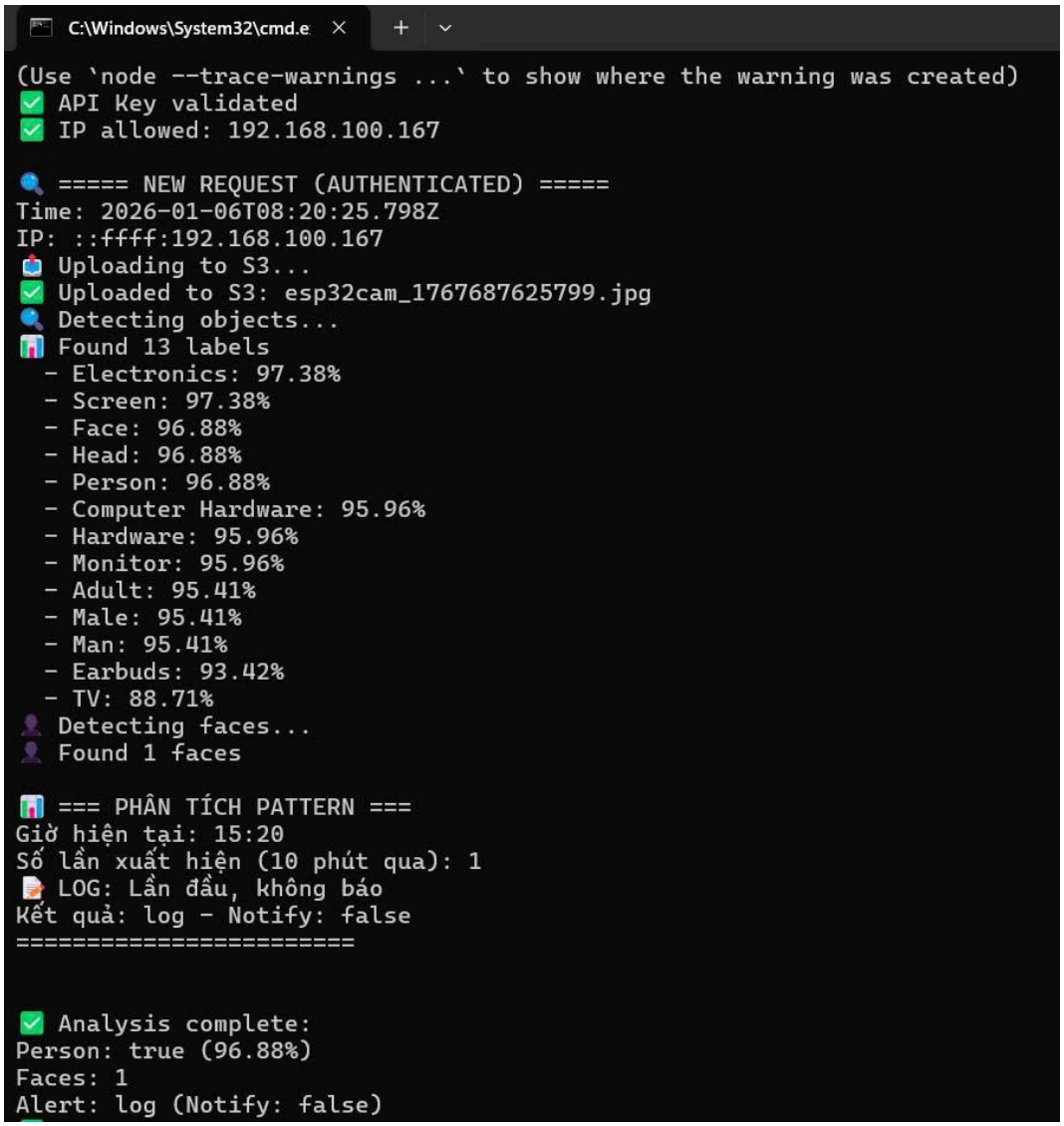


Hình 4.4 : ESP32-CAM gửi HTTP POST request và server xác thực API Key, IP Whitelist và Rate Limiting

4.3.4 Xử lý dữ liệu hình ảnh và phân tích hành vi trên nền tảng Cloud

Sau khi server xác thực thành công yêu cầu từ thiết bị IoT, dữ liệu hình ảnh do ESP32-CAM gửi lên được tải lên dịch vụ lưu trữ Amazon S3 để phục vụ quá trình xử lý. Tiếp theo, server sử dụng dịch vụ AWS Rekognition để phân tích nội dung hình ảnh nhằm phát hiện các đối tượng xuất hiện, đặc biệt là đối tượng con người. Kết quả phân tích bao gồm danh sách các nhãn nhận diện kèm theo độ tin cậy, số lượng khuôn mặt được phát hiện và các thông tin liên quan.

Dựa trên kết quả nhận diện, hệ thống tiếp tục thực hiện phân tích hành vi (pattern analysis) bằng cách đánh giá thời điểm xuất hiện và tần suất xuất hiện của đối tượng trong một khoảng thời gian xác định. Trên cơ sở đó, hệ thống phân loại mức độ sự kiện và quyết định có kích hoạt cảnh báo hay không. Cơ chế này giúp hạn chế các cảnh báo không cần thiết, đồng thời nâng cao độ chính xác và hiệu quả của hệ thống giám sát an ninh gia đình.

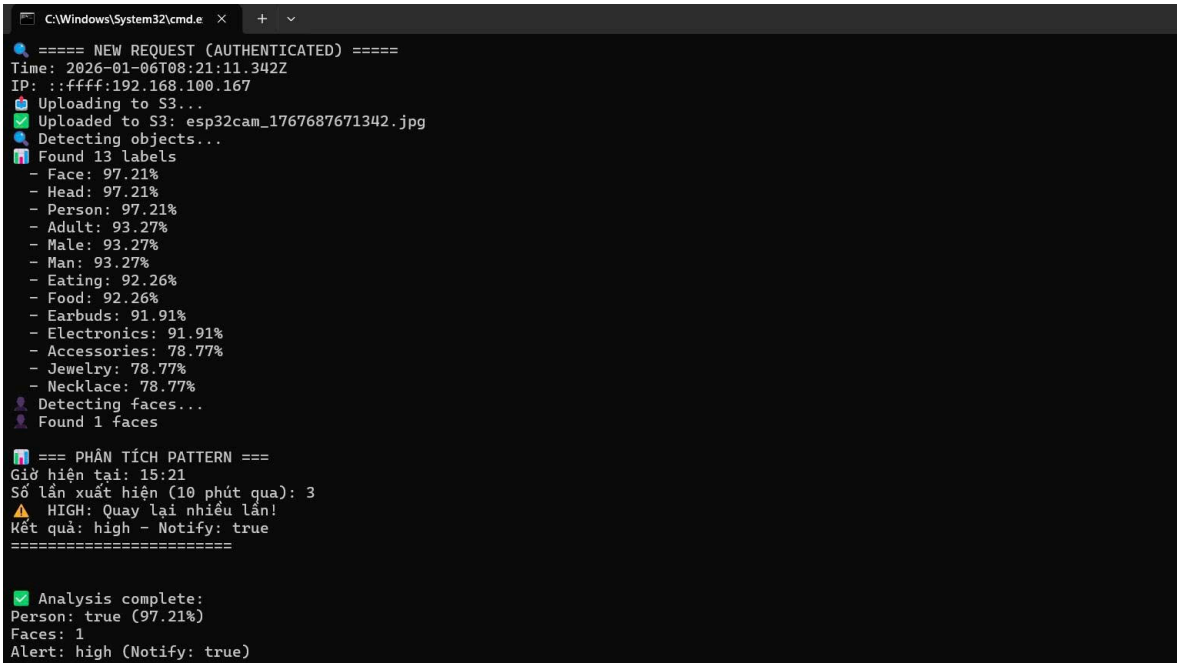


Hình 4.5 : Server tải ảnh lên AWS và thực hiện phân tích đối tượng, hành vi bằng AWS Rekognition

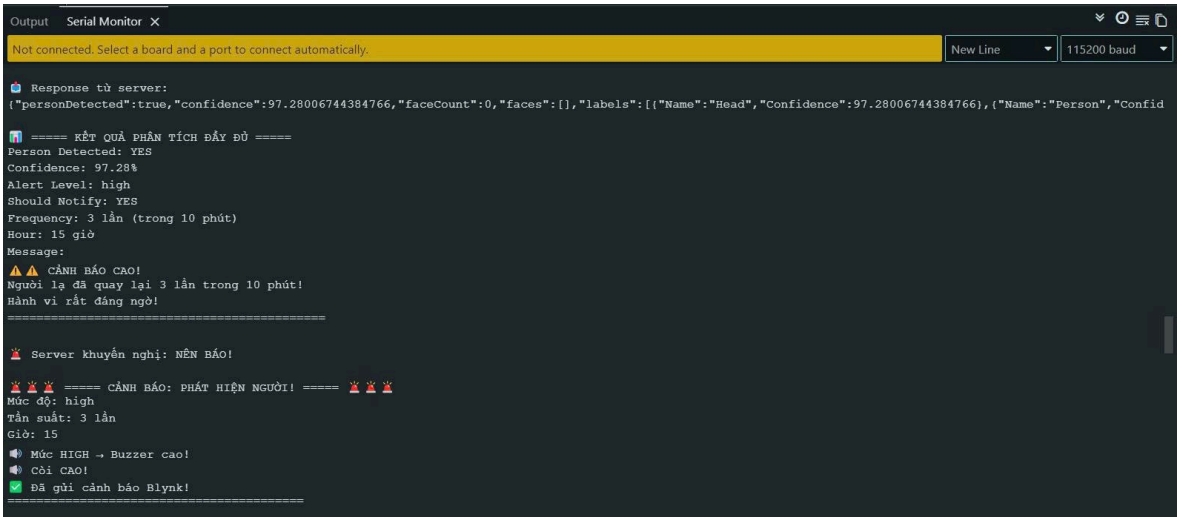
4.3.5 Phản hồi kết quả phân tích và kích hoạt cảnh báo từ thiết bị IoT

Sau khi hoàn tất quá trình phân tích hình ảnh và hành vi trên nền tảng Cloud, server tổng hợp kết quả và trả về dữ liệu phản hồi dưới dạng JSON cho thiết bị ESP32-CAM. Nội dung phản hồi bao gồm trạng thái phát hiện đối tượng con người, độ tin cậy của kết quả nhận diện, số lượng khuôn mặt được phát hiện, mức độ cảnh báo và quyết định có gửi thông báo hay không.

ESP32-CAM tiếp nhận dữ liệu phản hồi từ server, tiến hành phân tích nội dung JSON và thực hiện các hành động tương ứng. Trong trường hợp mức cảnh báo được xác định ở mức cao, thiết bị kích hoạt cơ chế cảnh báo như phát còi báo động và đồng thời gửi thông báo cảnh báo đến người dùng thông qua nền tảng Blynk. Cơ chế phản hồi hai chiều này giúp hệ thống đưa ra quyết định kịp thời, nâng cao khả năng phản ứng và đảm bảo hiệu quả giám sát an ninh trong môi trường thực tế.



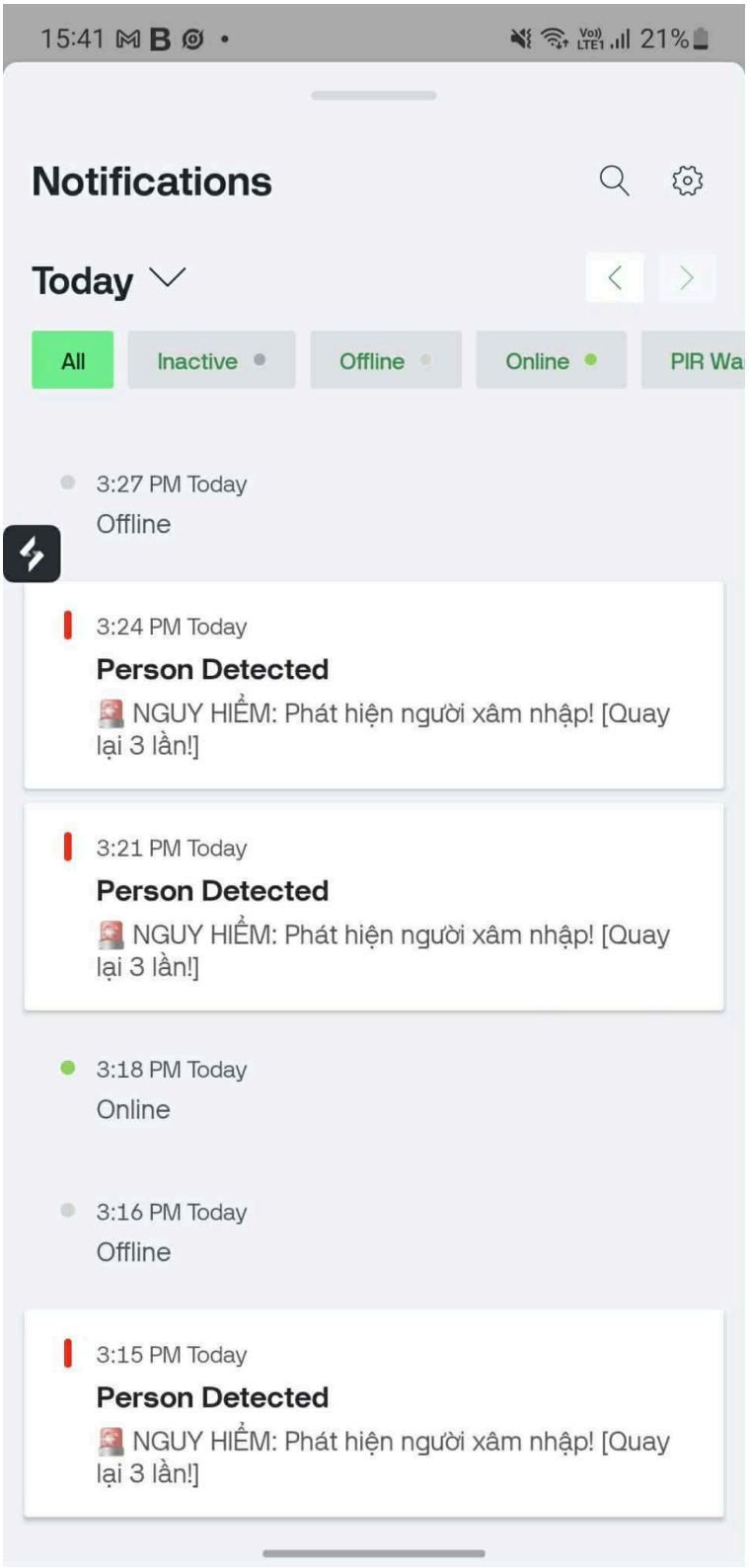
Hình 4.6 : Quá trình phản hồi kết quả phân tích và phát cảnh báo



Hình 4.7 : ESP32-CAM nhận kết quả phân tích từ server và gửi cảnh báo Blynk

4.3.6 Cập nhật Dashboard Blynk gửi thông báo cảnh báo theo kết quả phân tích

Trong quá trình vận hành, hệ thống sử dụng nền tảng Blynk để hiển thị trạng thái và kết quả giám sát theo thời gian thực. Cụ thể, các thông tin từ ESP32-CAM sau khi nhận phản hồi phân tích từ server sẽ được cập nhật lên Dashboard thông qua các Virtual Pin từ V0 đến V8, giúp người dùng theo dõi trực quan các trạng thái quan trọng (ví dụ: trạng thái thiết bị, phát hiện người, mức cảnh báo, tần suất xuất hiện, thời gian, và các thông tin liên quan). Đồng thời, khi biến shouldNotify = true, hệ thống sẽ kích hoạt cơ chế gửi thông báo cảnh báo lên ứng dụng Blynk, thông báo ngay lập tức đến điện thoại người dùng về sự kiện bất thường (ví dụ: phát hiện người với mức cảnh báo cao hoặc xuất hiện lặp lại nhiều lần trong thời gian ngắn). Cơ chế này đảm bảo việc giám sát từ xa được thực hiện liên tục và giúp người dùng phản ứng kịp thời trước các tình huống có nguy cơ xâm nhập.

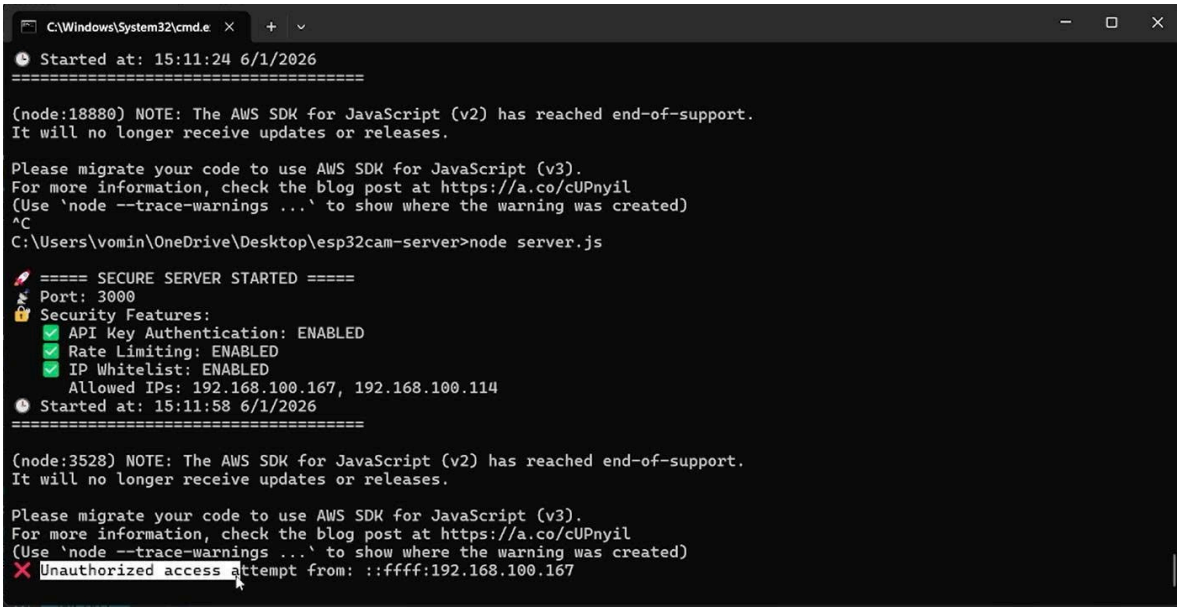


Hình 4.8 : Thông báo cảnh báo "Person Detected" được gửi đến người dùng thông qua Blynk

4.3.7 Kiểm thử cơ chế bảo mật và xác thực truy cập hệ thống

Trong quá trình thực nghiệm, hệ thống tiến hành kiểm thử các cơ chế bảo mật đã triển khai nhằm đánh giá khả năng ngăn chặn truy cập trái phép. Cụ thể, server được cấu hình kiểm tra API Key, IP Whitelist và Rate Limiting trước khi chấp nhận xử lý bất kỳ yêu cầu nào từ thiết bị IoT. Kết quả kiểm thử cho thấy, khi một request được gửi đến server với API Key không hợp lệ hoặc thiếu thông tin xác thực, server ngay lập tức từ chối yêu cầu và ghi nhận sự kiện truy cập trái phép. Thông báo "Unauthorized access attempt" được hiển thị trên console server, đồng thời request không được xử lý tiếp theo. Cơ chế này giúp ngăn chặn các truy cập không hợp lệ ngay từ lớp đầu tiên, đảm bảo an toàn cho hệ thống trong quá trình vận hành.

Kết quả thực nghiệm chứng minh rằng các biện pháp bảo mật đã triển khai hoạt động hiệu quả, góp phần bảo vệ hệ thống khỏi các nguy cơ tấn công như giả mạo thiết bị, gửi dữ liệu trái phép hoặc khai thác API không được cấp quyền.

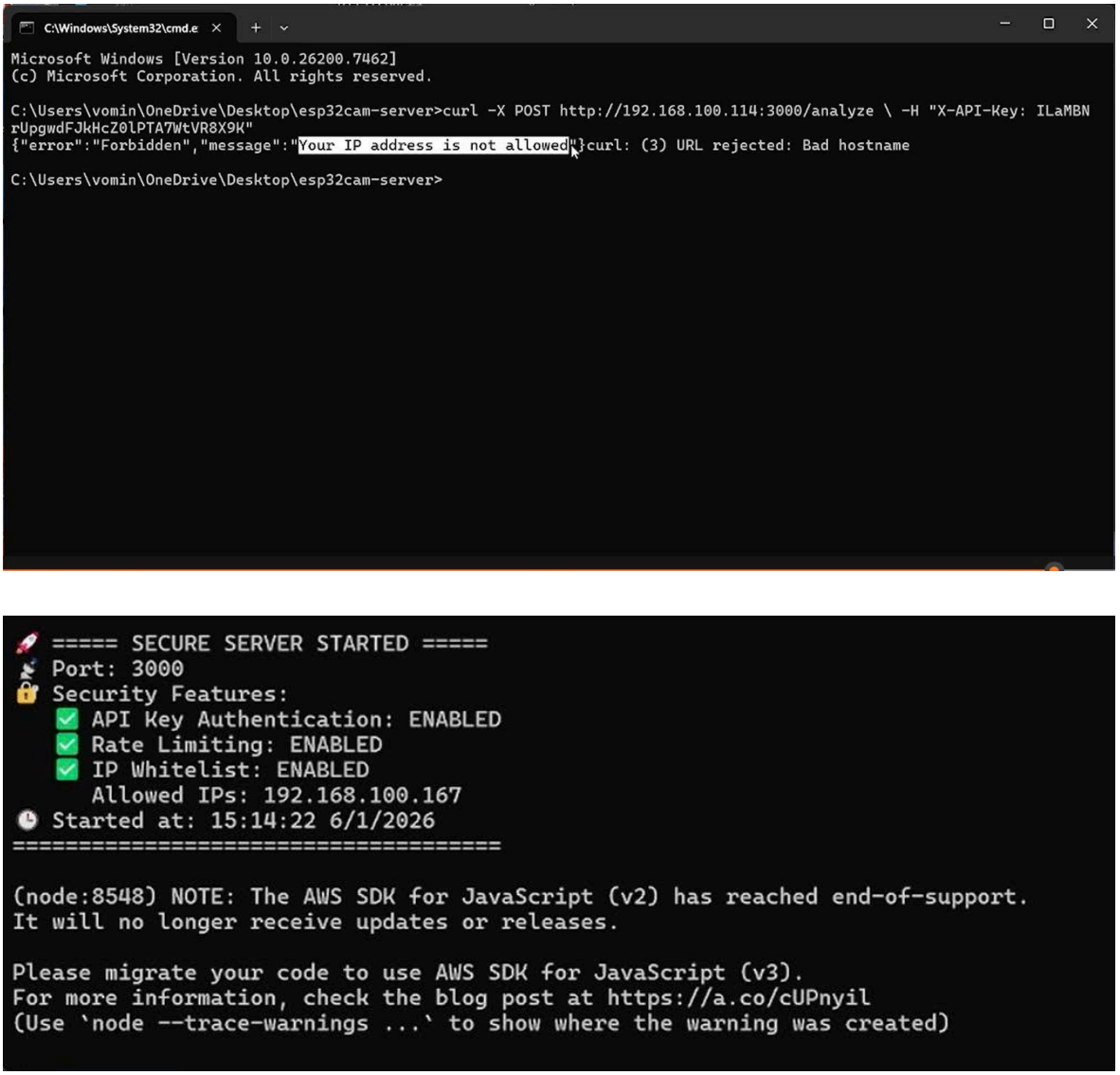


Hình 4.9 : Server từ chối truy cập khi API Key không hợp lệ và ghi nhận "Unauthorized access attempt"

4.3.8 Kiểm thử cơ chế IP Whitelist và ngăn chặn truy cập trái phép

Trong quá trình kiểm thử bảo mật, hệ thống tiến hành đánh giá hiệu quả của cơ chế IP Whitelist nhằm kiểm soát nguồn truy cập đến server. Cụ thể, địa chỉ IP của máy tính thực hiện gửi request thử nghiệm được loại bỏ khỏi danh sách IP được phép (Whitelist) trên server. Sau khi cấu hình lại whitelist, máy tính tiếp tục gửi yêu cầu HTTP POST đến endpoint phân tích của server. Kết quả cho thấy server đã từ chối request ngay lập tức và trả về thông báo lỗi "Forbidden – Your IP address is not allowed". Đồng thời, request không được xử lý và không phát sinh bất kỳ thao tác tải ảnh hay phân tích dữ liệu nào trên hệ thống.

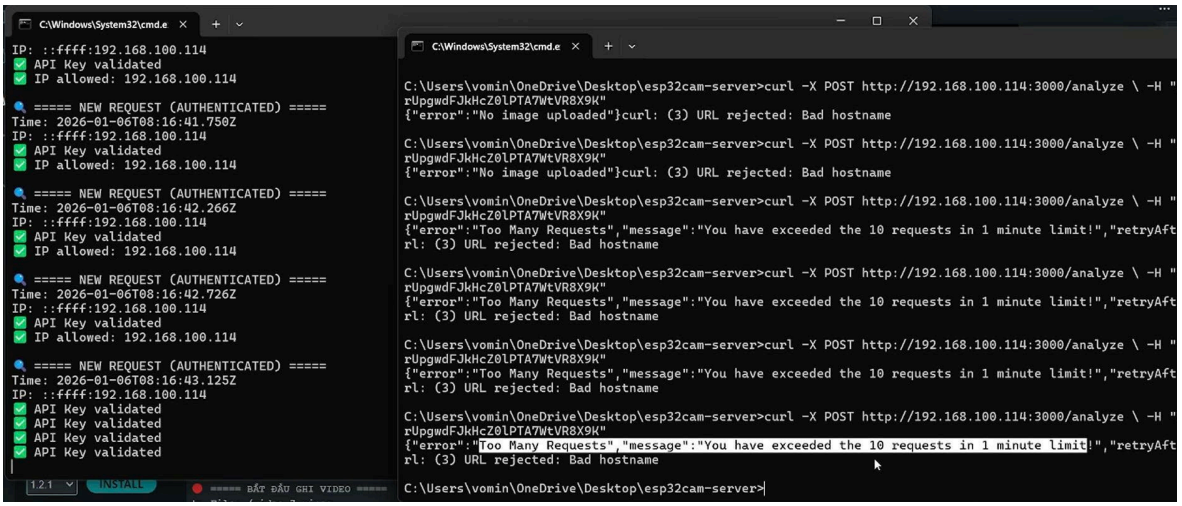
Kết quả kiểm thử chứng minh rằng cơ chế IP Whitelist hoạt động đúng như thiết kế, giúp hạn chế hiệu quả các truy cập trái phép ngay cả khi request có kèm API Key hợp lệ. Biện pháp này góp phần nâng cao mức độ an toàn cho hệ thống, đặc biệt trong môi trường triển khai nội bộ hoặc mạng LAN có kiểm soát.



Hình 4.10 : Kiểm thử IP Whitelist – server từ chối truy cập khi địa chỉ IP không nằm trong danh sách cho phép

4.3.9 Kiểm thử cơ chế Rate Limiting và phòng chống tấn công spam

Để đánh giá khả năng bảo vệ hệ thống trước các hành vi gửi yêu cầu liên tục hoặc tấn công từ chối dịch vụ (DoS) ở mức cơ bản, hệ thống đã triển khai và kiểm thử cơ chế Rate Limiting trên server. Trong kịch bản kiểm thử, một thiết bị hợp lệ (đã được xác thực API Key và nằm trong danh sách IP Whitelist) được sử dụng để gửi liên tiếp nhiều yêu cầu HTTP POST đến endpoint phân tích trong thời gian ngắn. Kết quả cho thấy, khi số lượng request vượt quá ngưỡng cho phép (10 request trong vòng 1 phút), server tự động từ chối các yêu cầu tiếp theo và trả về thông báo lỗi "Too Many Requests". Đồng thời, các request vượt ngưỡng không được xử lý, không phát sinh thao tác tải ảnh lên Cloud hay phân tích dữ liệu.



Hình 4.11 : Kiểm thử Rate Limiting – server từ chối request khi vượt quá số lượng cho phép

Kết quả kiểm thử chứng minh rằng cơ chế Rate Limiting hoạt động hiệu quả, giúp hạn chế nguy cơ tấn công spam API, giảm tải cho server và bảo vệ tài nguyên hệ thống trong quá trình vận hành thực tế.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

5.1.1 Đánh giá hệ thống

a. Ưu điểm

Hệ thống hoạt động ổn định trong môi trường mạng nội bộ, đáp ứng tốt yêu cầu giám sát liên tục với độ tin cậy cao. Quá trình triển khai đơn giản, sử dụng các thiết bị phổ biến nên chi phí đầu tư thấp, phù hợp với mô hình gia đình. Việc áp dụng xác thực API và cơ chế kiểm soát truy cập cơ bản giúp hạn chế truy cập trái phép vào server. Ngoài ra, hệ thống hỗ trợ gửi thông báo thời gian thực thông qua nền tảng Blynk, đảm bảo người dùng nhận được cảnh báo kịp thời khi có sự kiện xảy ra.

b. Hạn chế

Mức độ bảo mật của kênh truyền dữ liệu chưa được nâng cao, chủ yếu dừng ở các cơ chế cơ bản. Hệ thống phụ thuộc nhiều vào kết nối Wi-Fi, do đó hiệu năng và tính ổn định có thể bị ảnh hưởng khi mạng không ổn định. Bên cạnh đó, dữ liệu giám sát chưa được lưu trữ dài hạn, gây hạn chế trong việc tra cứu và phân tích lịch sử hoạt động của hệ thống.

5.1.2 Mức độ đáp ứng yêu cầu đề tài

Hệ thống đã hoàn thành đầy đủ các chức năng giám sát an ninh IoT theo mục tiêu đề ra, đảm bảo khả năng theo dõi và phát hiện sự kiện trong môi trường thực tế. Dữ liệu được thu thập từ các cảm biến và camera, sau đó được xử lý và hiển thị cho người dùng thông qua các nền tảng giám sát phù hợp. Hệ thống tích hợp lớp bảo mật cơ bản, đáp ứng yêu cầu an toàn trong mô hình thử nghiệm và hạn chế truy cập trái phép. Giao diện người dùng được thiết kế đơn giản, trực quan, giúp người dùng dễ dàng thao tác và theo dõi trạng thái hệ thống.

5.1.3 Kết luận

Hệ thống IoT được triển khai đã đáp ứng đầy đủ các mục tiêu của đề tài, thể hiện khả năng vận hành ổn định trong môi trường thực nghiệm mạng LAN gia đình. Thông qua quá trình thử nghiệm, hệ thống cho thấy khả năng thu thập dữ liệu từ cảm biến và camera, xử lý sự kiện giám sát và truyền thông tin đến người dùng một cách liên tục và tin cậy. Các cơ chế kết nối, xác thực và thông báo được tích hợp đồng bộ, giúp hệ thống duy trì hoạt động ổn định ngay cả khi xảy ra gián đoạn tạm thời của mạng.

Kết quả thực nghiệm khẳng định tính khả thi của mô hình triển khai IoT trong phạm vi gia đình với chi phí thấp và cấu hình đơn giản, đồng thời vẫn đảm bảo các yêu cầu cơ bản về an ninh và khả năng phản hồi thời gian thực. Bên cạnh đó, kiến trúc hệ thống được xây dựng theo hướng mở, cho phép dễ dàng mở rộng trong tương lai như nâng cao mức độ bảo mật, tích hợp lưu trữ dữ liệu dài hạn, phân tích thông minh hoặc kết nối với các nền tảng cloud nâng cao. Do đó, mô hình này không chỉ phù hợp cho mục đích học tập và thử nghiệm mà còn là nền tảng cho các nghiên cứu và phát triển tiếp theo trong lĩnh vực giám sát an ninh IoT.

5.2 Hướng phát triển

Hướng phát triển tiếp theo có thể tập trung nâng cao tính an toàn và khả năng mở rộng của hệ thống. Trước hết, áp dụng HTTPS/TLS cho toàn bộ kênh truyền giữa ESP32-CAM và server nhằm mã hóa dữ liệu, giảm nguy cơ bị nghe lén hoặc can thiệp trong quá trình truyền thông. Song song đó, cơ chế quản lý khóa truy cập cần được thiết kế theo hướng động thay vì dùng một API Key tĩnh, ví dụ tạo token theo phiên, có thời hạn (expire), hỗ trợ thu hồi khóa và cấp lại khi nghi ngờ bị lộ, đồng thời phân quyền theo thiết bị/người dùng để kiểm soát truy cập chặt chẽ hơn.

Về lưu trữ, hệ thống có thể mở rộng theo hướng ghi nhận và lưu trữ dữ liệu dài hạn trên Cloud, bao gồm ảnh chụp khi có sự kiện, video ngắn, log truy cập và

lịch sử cảnh báo. Dữ liệu được tổ chức theo cấu trúc thời gian và theo thiết bị, kết hợp cơ chế sao lưu và chính sách vòng đời (lưu bao lâu, tự xóa sau N ngày) để tối ưu chi phí. Khi có kho dữ liệu ổn định, hệ thống có thể tích hợp AI nhận diện thông minh như phát hiện người lạ, phân loại đối tượng (người/xe/thú cưng), nhận diện khuôn mặt theo danh sách cho phép, lọc cảnh báo giả do chuyển động nhỏ, đồng thời đưa ra mức độ cảnh báo theo ngữ cảnh để tăng độ chính xác và trải nghiệm người dùng.

Cuối cùng, để mở rộng cho nhiều thiết bị và nhiều người dùng, kiến trúc cần hỗ trợ quản lý đồng thời nhiều ESP32-CAM trong cùng một dashboard, mỗi thiết bị có định danh riêng và cấu hình độc lập. Hệ thống cũng nên bổ sung cơ chế tài khoản người dùng, phân quyền (admin/user/guest), chia sẻ quyền xem camera theo nhóm, đồng bộ thông báo theo từng khu vực giám sát, và hỗ trợ truy cập từ xa an toàn. Các hướng phát triển này giúp hệ thống chuyển từ mô hình thử nghiệm sang mô hình triển khai thực tế quy mô lớn hơn, ổn định hơn và thông minh hơn.

TÀI LIỆU THAM KHẢO

Tiếng Việt

Espressif Systems, *Tài liệu kỹ thuật ESP32*, Espressif Systems, 2023.

Espressif Systems, *Tài liệu bo mạch ESP32-CAM*, Espressif Systems.

Arduino, *Tài liệu Arduino IDE*, https://docs.arduino.cc

Arduino,

Thư viện WiFi cho ESP32,https://www.arduino.cc/reference/en/libraries/wifi

Blynk, *Tài liệu nền tảng IoT Blynk*, https://docs.blynk.io

Tiếng Anh

Node.js Foundation, *Node.js Official Documentation*, https://nodejs.org/en/docs

Express.js, *Express – Node.js Web Framework*, https://expressjs.com

Amazon Web Services,

AWS Security Best Practices, https://docs.aws.amazon.com/security

Amazon Web Services, *AWS IAM User Guide*, https://docs.aws.amazon.com/iam

OWASP Foundation, *OWASP Top 10 – Web Application Security Risks*, https://owasp.org/www-project-top-ten

Kevin Ashton, *That "Internet of Things" Thing*, RFID Journal, 2009.

S. Sicari et al., *Security, Privacy and Trust in Internet of Things: The Road Ahead*, Computer Networks, Elsevier, 2015.