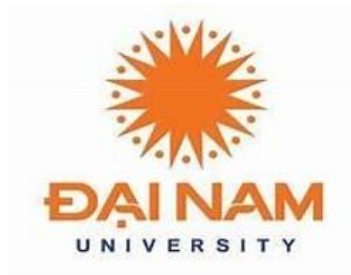


**TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN TỐT NGHIỆP
XÂY DỰNG HỆ THỐNG GHI HÌNH
QUA INTERNET

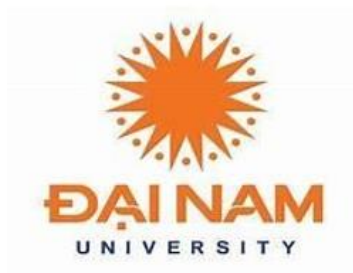
SINH VIÊN THỰC HIỆN : NGUYỄN XUÂN VIỄN

MÃ SINH VIÊN : 1451020273

KHOA : CÔNG NGHỆ THÔNG TIN

Hà Nội - 2024

**TRƯỜNG ĐẠI HỌC ĐẠI NAM
KHOA CÔNG NGHỆ THÔNG TIN**



NGUYỄN XUÂN VIỄN

**XÂY DỰNG HỆ THỐNG GHI HÌNH
QUA INTERNET**

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

MÃ SỐ : 74.80.201

NGƯỜI HƯỚNG DẪN: TS. TRẦN ĐĂNG CÔNG

Hà Nội - 2024

[illegible]

Ký và ghi họ tên

LỜI CAM ĐOAN

Tôi xin cam đoan rằng báo cáo có tên: “Xây dựng hệ thống ghi hình qua internet” là kết quả của công việc nghiêm túc và tự chủ của bản thân. Trong quá trình nghiên cứu và phát triển dự án này, tôi đã tuân thủ các nguyên tắc và chuẩn mực đạo đức nghiêm ngặt.

Mọi thông tin, số liệu và kết quả được trình bày trong báo cáo đều tuân thủ nguyên tắc trung thực và minh bạch nhất có thể. Tôi đã sử dụng tài liệu tham khảo và nguồn thông tin từ các nguồn uy tín, và những phần này đã được dẫn chứng rõ ràng và nêu rõ trong phần tài liệu tham khảo.

Đồng thời, tôi cam kết rằng nếu có bất kỳ sai sót nào trong thông tin trình bày, tôi sẽ chịu trách nhiệm hoàn toàn và tuân theo mọi quy định và kỷ luật do bộ môn và nhà trường áp đặt.

Tôi cũng xác nhận rằng quy trình kiểm tra, xác minh thông tin và kiểm soát chất lượng đã được thực hiện để đảm bảo tính chính xác và đáng tin cậy của mọi thông tin được trình bày trong báo cáo này.

Hà Nội, ngày 14 tháng 06 năm 2024

DANH MỤC KÝ HIỆU HOẶC CHỮ VIẾT TẮT

STT	Chữ viết tắt	Chữ viết đầy đủ (tiếng Anh)	Chữ viết đầy đủ (tiếng Việt)
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	Api	Application programming interface	Giao diện lập trình ứng dụng
3	CSS	Cascading Style Sheets	Các tập tin định kiểu theo tầng
4	CNTT		Công nghệ thông tin
5	DOM	Document Object Model	Mô hình đối tượng tài liệu
6	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
7	HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản
8	IDE	Integrated development environment	Môi trường phát triển tích hợp
9	IoT	Internet of thing	Internet vạn vật
10	LAN	Local area network	Mạng cục bộ
11	TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
12	USB	Universal Serial Bus	Cổng giao tiếp đa năng

DANH MỤC HÌNH ẢNH

Hình 1.1: Mô hình hoạt động của hệ thống IoT	3
Hình 2.1: Mô hình hoạt động của Flask	11
Hình 2.2: Thành phần chính của bo mạch Arduin.....	13
Hình 2.3: Quy trình làm việc để tải mã nguồn vào bo mạch arduino	15
Hình 2.4: Mapping cho một module chip esp32	17
Hình 2.5: ESP32-CAM.....	18
Hình 2.6: Sơ đồ chân ESP32-CAM.....	19
Hình 2.7: Mối quan hệ giữa các dự án, ứng dụng và sản phẩm Firebase.....	21
Hình 2.8: Xử lý yêu cầu trang JavaScript.....	22
Hình 2.9: Các ứng dụng của JavaScript	25
Hình 4.1: Kiến trúc tổng quan của hệ thống ghi hình qua Internet	30
Hình 4.2: Module ESP32-CAM OV2640	31
Hình 4.3: Module ESP32-CAM-MB.....	32
Hình 4.4: Dây cáp Micro USB	33
Hình 4.5: Sơ đồ lắp đặt.....	33
Hình 4.6: Lắp đặt thực tế	34
Hình 4.7: Sơ đồ dự án.....	35
Hình 4.8: Mã nguồn camera_pins.h	36
Hình 4.9: Mã nguồn camera_index.h	37
Hình 4.10: Mã nguồn app_httpd.cpp.....	37
Hình 4.11: Mã nguồn CameraWebServer.ino.....	38
Hình 4.12: Hàm tạo luồng stream.....	39
Hình 4.13: Hàm tạo tệp video mp4	39
Hình 4.14: Hàm chụp ảnh màn hình.....	40
Hình 4.15: Cấu hình kết nối với firebase.....	40
Hình 4.16: Khởi tạo firebase và xây dựng hàm upload, download	40
Hình 4.17: Thiết kế giao diện web	41
Hình 4.18: Nội dung của index chính.....	42

Hình 4.19: Các hàm xây dựng trong script.....	42
Hình 4.20: Giao diện chương trình.....	43
Hình 4.21: Thực hiện ghi hình.....	43
Hình 4.22: Chức năng upload.....	44
Hình 4.23: Chức năng download.....	44

LỜI MỞ ĐẦU

1. Tính cấp thiết của đề tài

Tính cấp thiết của đề tài "xây dựng hệ thống ghi hình qua internet" nằm ở việc nó mang lại nhiều lợi ích và ứng dụng quan trọng trong nhiều lĩnh vực khác nhau. Đầu tiên, hệ thống này cung cấp tính tiện ích và linh hoạt cao, cho phép người dùng truy cập và xem các video từ xa thông qua internet, đáp ứng nhu cầu giám sát và quản lý từ xa một cách thuận tiện. Thứ hai, tính an ninh và bảo mật được tăng cường thông qua khả năng quản lý việc ghi lại video và lưu trữ an toàn, giúp bảo vệ nhà cửa, doanh nghiệp hoặc các nơi công cộng. Ngoài ra, hệ thống ghi hình qua internet cũng hỗ trợ quản lý tài nguyên và nhân viên hiệu quả hơn trong doanh nghiệp và tổ chức, giúp họ kiểm soát hoạt động trong các khu vực khác nhau và đưa ra các quyết định dựa trên thông tin thu thập được. Cuối cùng, việc nghiên cứu và phát triển hệ thống này cũng đóng vai trò quan trọng trong việc phát triển công nghệ, tạo ra cơ hội cho sự tiên bộ và sáng tạo trong lĩnh vực này. Tóm lại, tính cấp thiết của đề tài này nằm ở khả năng cung cấp các giải pháp hiệu quả cho nhu cầu giám sát, bảo vệ và quản lý từ xa, đồng thời đóng góp vào sự phát triển của công nghệ và ứng dụng thực tiễn.

Đề tài này có thể ứng dụng trong các lĩnh vực cụ thể như: hệ thống nhà thông minh, giao thông vận tải, y tế, giáo dục... Nó mang đến nhiều lợi ích thiết thực trong đời sống. Ví dụ như việc có các màn hình giám sát giúp gia đình theo dõi được tài sản và giúp tăng cường được khả năng an ninh của bản thân, tránh được một số những rủi ro nhất định; các camera hành trình ghi lại quá trình của người tham gia giao thông từ đó giúp giải quyết được nhiều tình huống phức tạp; theo dõi được tình trạng sức khỏe của bệnh nhân, giúp các bác sĩ, y tá, nhân viên chăm sóc có những hành động kịp thời để đảm bảo an toàn sức khỏe của bệnh nhân trong nhiều tình huống khẩn cấp; ghi hình, giám sát thi cử để từ đó nâng cao chất lượng giáo dục, tạo môi trường cạnh tranh công bằng, lành mạnh.

2. Mục đích nghiên cứu

Mục đích của nghiên cứu này là xây dựng và phát triển một hệ thống ghi hình qua internet với các tính năng và chức năng hiện đại, nhằm đáp ứng nhu cầu ngày càng tăng về

giám sát và quản lý từ xa trong các môi trường khác nhau. Cụ thể ở bài báo cáo này tôi xin trình bày về một hệ thống an ninh cung cấp cho đối tượng sử dụng là đội ngũ an ninh của doanh nghiệp hay căn hộ, khách sạn. Ngoài ra, mục đích của nghiên cứu cũng là tối ưu hóa hiệu suất và tiết kiệm tài nguyên, đặc biệt là băng thông và lưu trữ, thông qua việc áp dụng các kỹ thuật nén dữ liệu và quản lý tài nguyên thông minh. Điều này giúp giảm chi phí vận hành và duy trì hệ thống trong thời gian dài, đồng thời tăng cường tính cạnh tranh của hệ thống trên thị trường.

Tóm lại, mục đích của nghiên cứu là xây dựng một hệ thống ghi hình qua internet đa chức năng, linh hoạt và hiệu suất cao, kết hợp công nghệ tiên tiến để đáp ứng nhu cầu ngày càng tăng về giám sát và quản lý từ xa trong các môi trường khác nhau.

3. Phạm vi nghiên cứu

Phạm vi của nghiên cứu này sẽ tập trung vào các khía cạnh quan trọng của việc xây dựng hệ thống ghi hình qua internet, bao gồm:

- Phát triển hệ thống phần cứng và phần mềm: Nghiên cứu sẽ tiến hành phát triển cả phần cứng và phần mềm của hệ thống ghi hình. Phần cứng bao gồm việc chọn lựa và tích hợp các cảm biến hình ảnh, vi xử lý, mạch giao tiếp mạng.
- Kết nối và giao tiếp qua internet: Một phần quan trọng của nghiên cứu là phát triển giao thức truyền thông và giao diện mạng cho phép hệ thống ghi hình kết nối và giao tiếp qua internet. Điều này bao gồm việc nghiên cứu các công nghệ mạng hiện đại như TCP/IP, HTTP để đảm bảo tính ổn định và an toàn trong truyền tải dữ liệu.
- Tích hợp công nghệ mới: Nghiên cứu sẽ khám phá và áp dụng các công nghệ mới như Internet of Things (IoT) để kết nối các thiết bị thông minh và cảm biến.
- Thử nghiệm và đánh giá hiệu suất: Cuối cùng, nghiên cứu sẽ tiến hành thử nghiệm và đánh giá hiệu suất của hệ thống ghi hình qua internet trong các điều kiện thực tế. Điều này bao gồm việc đo lường và đánh giá các chỉ số như độ trễ, băng thông, độ phân giải hình ảnh, và độ tin cậy của hệ thống.

- Tóm lại, phạm vi của nghiên cứu sẽ bao gồm cả các khía cạnh kỹ thuật và ứng dụng của việc xây dựng hệ thống ghi hình qua internet, với mục tiêu là tạo ra một hệ thống linh hoạt, hiệu suất và an toàn đáp ứng được nhu cầu của người dùng trong các môi trường khác nhau.

4. Phương pháp nghiên cứu

Phương pháp nghiên cứu sẽ bao gồm các bước từ nghiên cứu thị trường và thiết kế kiến trúc đến phát triển, kiểm thử và đánh giá, và cuối cùng là tối ưu hóa và cải tiến hệ thống ghi hình qua internet. Quá trình này nhằm mục đích đảm bảo rằng hệ thống đạt được hiệu suất và tính ổn định cao nhất để đáp ứng nhu cầu của người dùng.

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn các thầy cô trong khoa CNTT cũng như các thầy cô giảng dạy trong trường đã giúp tôi có được những kiến thức tổng quan để hoàn thành bài báo cáo này. Đặc biệt, tôi muốn gửi lời cảm ơn sâu sắc nhất đến TS. Trần Đăng công và GVHD. Nguyễn Văn Nhân là những người đã đồng hành và hướng dẫn tôi suốt quãng thời gian thực hiện đồ án tốt nghiệp của bản thân. Trong quá trình này, sự tận tâm và sự quan tâm từ các thầy đã là nguồn động viên lớn, giúp tôi tiếp cận kiến thức một cách sâu sắc và tự tin hơn khi tiếp cận dự án thực tế của mình.

Tôi rất mong nhận được sự đánh giá và góp ý chân thành từ thầy để bài báo cáo của tôi có thể được cải thiện và hoàn thiện hơn. Lời hướng dẫn và ý kiến từ thầy sẽ giúp tôi nắm bắt rõ hơn những khía cạnh cần điều chỉnh, từ đó phát triển kỹ năng và hiểu biết của mình một cách toàn diện hơn.

Cuối cùng, tôi xin gửi lời cảm ơn đến tất cả những người đã đóng góp ý kiến, phản hồi và ý tưởng quan trọng. Sự sáng tạo và đóng góp của các bạn đã làm phong phú và nâng cao chất lượng của dự án.

Bài báo cáo của tôi có thể chưa tránh khỏi những sai sót do những hạn chế trong quá trình thực hiện. Mong rằng thầy và quý độc giả có thể xem xét và góp ý để giúp tôi hoàn thiện bài báo cáo một cách tốt nhất. Sự chỉ dẫn và phản hồi từ thầy sẽ giúp tôi cải thiện kiến thức và kỹ năng một cách toàn diện hơn.

Tôi xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	1
1.1. Kiến thức chung về IoT	1
1.2. Cấu trúc và thành phần của một hệ thống IoT.....	2
1.3. Mô hình hoạt động của hệ thống IoT	3
1.3.1. Thu thập dữ liệu.....	3
1.3.2. Truyền tải dữ liệu.....	4
1.3.3. Xử lý dữ liệu ban đầu	4
1.3.4. Lưu trữ và phân tích dữ liệu	4
1.3.5. Hành động dựa trên dữ liệu	4
1.3.6. Giao tiếp với người dùng.....	4
1.3.7. Quản lý và bảo trì	5
1.4. Ứng dụng của IoT trong lĩnh vực ghi hình qua internet.....	5
CHƯƠNG 2: CÁC CÔNG NGHỆ LIÊN QUAN ĐẾN XÂY DỰNG HỆ THỐNG GHI HÌNH QUA INTERNET	7
2.1. Công nghệ truyền thông và mạng: TCP/IP, HTTP	7
2.2. Công nghệ cảm biến và thiết bị hình ảnh: Camera IP	7
2.3. Giới thiệu về Framework Python Flask.....	8
2.3.1. Ngôn ngữ lập trình Python	8
2.3.2. Framework Flask	10
2.4. Arduino và module ESP32	12
2.4.1. Arduino	12
2.4.2. Module ESP32.....	15
2.5. Công nghệ đám mây: Firebase	20
2.5.1. Tổng quan về Firebase.....	20
2.5.2. Mối quan hệ giữa các dự án, ứng dụng và sản phẩm Firebase.....	20
2.6. Ngôn ngữ lập trình JavaScript	22

2.6.1. Khái niệm	22
2.6.2. Cách thức hoạt động.....	22
2.6.3. Ưu nhược điểm của JavaScript.....	24
2.6.4. Ứng dụng	25
CHƯƠNG 3: CÁC TIÊU CHUẨN VÀ QUY ĐỊNH LIÊN QUAN	28
3.1. Tiêu chuẩn kỹ thuật về IoT và ghi hình qua internet.....	28
3.2. Quy định pháp lý và an ninh thông tin	29
CHƯƠNG 4: XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG.....	30
4.1. Xây dựng hệ thống ghi hình qua internet	30
4.1.1. Các yêu cầu bài toán.....	31
4.1.2. Các thiết bị yêu cầu	31
4.1.3. Sơ đồ lắp đặt	33
4.1.4. Minh hoạ thực tế.....	34
4.2. Thiết kế chương trình	35
4.2.1. Thiết kế chương trình ESP32-Cam.....	35
4.2.2. Thiết kế ứng dụng web	39
4.3. Thực thi chương trình	43
4.3.1. Giao diện chương trình.....	43
4.3.2. Thực hiện các chức năng	43
4.4. Đánh giá hệ thống.....	45
KẾT LUẬN	46
TÀI LIỆU THAM KHẢO	48

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Kiến thức chung về IoT

Internet of Things (IoT) là một khái niệm mạng lưới mà trong đó các thiết bị và cảm biến được kết nối với internet để thu thập và chia sẻ dữ liệu. Điều này cho phép các thiết bị thông minh giao tiếp với nhau và với con người, tạo ra một môi trường kết nối và điều khiển tự động.

Đặc điểm chính của IoT bao gồm:

- Kết nối: Các thiết bị và cảm biến được kết nối với internet thông qua các giao thức kết nối như Wi-Fi, Bluetooth, Zigbee, và LoRaWAN.
- Thu thập dữ liệu: IoT cho phép thu thập dữ liệu từ môi trường xung quanh thông qua các cảm biến như cảm biến nhiệt độ, độ ẩm, ánh sáng, và các dữ liệu khác.
- Truyền tải và xử lý dữ liệu: Dữ liệu được thu thập từ các thiết bị được truyền tải qua mạng để xử lý và phân tích, tạo ra thông tin hữu ích và hành động.
- Tự động hóa và điều khiển: IoT cho phép tự động hóa các quy trình và điều khiển các thiết bị từ xa thông qua internet.
- Quy mô lớn: Với sự phát triển của công nghệ, số lượng các thiết bị kết nối IoT ngày càng tăng, tạo ra một mạng lưới rộng lớn và phức tạp.

Ứng dụng của IoT bao gồm:

- Nhà thông minh: Điều khiển ánh sáng, nhiệt độ, an ninh từ xa thông qua điện thoại hoặc thiết bị thông minh.
- Y tế thông minh: Theo dõi sức khỏe của bệnh nhân, cảnh báo về các vấn đề y tế, và quản lý thuốc thông qua các thiết bị y tế kết nối.
- Công nghiệp 4.0: Tự động hóa quy trình sản xuất, giám sát và bảo dưỡng thiết bị công nghiệp từ xa.
- Giao thông thông minh: Quản lý luồng giao thông, giảm thiểu ùn tắc và tai nạn giao thông thông qua các hệ thống giao thông kết nối.

1.2. Cấu trúc và thành phần của một hệ thống IoT

Một hệ thống IoT thường bao gồm nhiều thành phần chính là thiết bị cảm biến và thiết bị kết nối, mạng kết nối, nền tảng phần mềm, và quản lý dữ liệu. Dưới đây là mô tả chi tiết về mỗi thành phần:

Thiết bị cảm biến và thiết bị kết nối:

- Thiết bị cảm biến: Là các thiết bị được trang bị các cảm biến để thu thập dữ liệu từ môi trường xung quanh, chẳng hạn như cảm biến nhiệt độ, độ ẩm, ánh sáng, chất lượng không khí, và nhiều loại cảm biến khác.
- Thiết bị kết nối: Là các thiết bị có khả năng kết nối với internet thông qua các giao thức như Wi-Fi, Bluetooth, Zigbee, LoRaWAN, và các công nghệ kết nối khác. Thiết bị kết nối giúp truyền tải dữ liệu từ thiết bị cảm biến đến nền tảng phần mềm và nhận lệnh điều khiển từ nền tảng.

Mạng kết nối:

- Các giao thức kết nối: IoT sử dụng các giao thức như Wi-Fi, Bluetooth, Zigbee, LoRaWAN để kết nối các thiết bị với internet và với nhau.
- Cơ sở hạ tầng mạng: Bao gồm các cơ sở hạ tầng mạng như địa chỉ IP, mạng LAN và WAN, cùng với các cơ sở hạ tầng phần mềm như các máy chủ và dịch vụ đám mây để hỗ trợ việc kết nối và truy cập dữ liệu từ xa.

Nền tảng phần mềm:

- Ứng dụng và giao diện người dùng: Bao gồm các ứng dụng di động, ứng dụng web hoặc các giao diện khác để người dùng tương tác và điều khiển các thiết bị IoT.
- Phần mềm trung tâm: Là phần mềm chạy trên máy chủ hoặc đám mây, có nhiệm vụ quản lý và điều khiển các thiết bị IoT, thu thập và phân tích dữ liệu, cung cấp các dịch vụ thông minh cho người dùng.

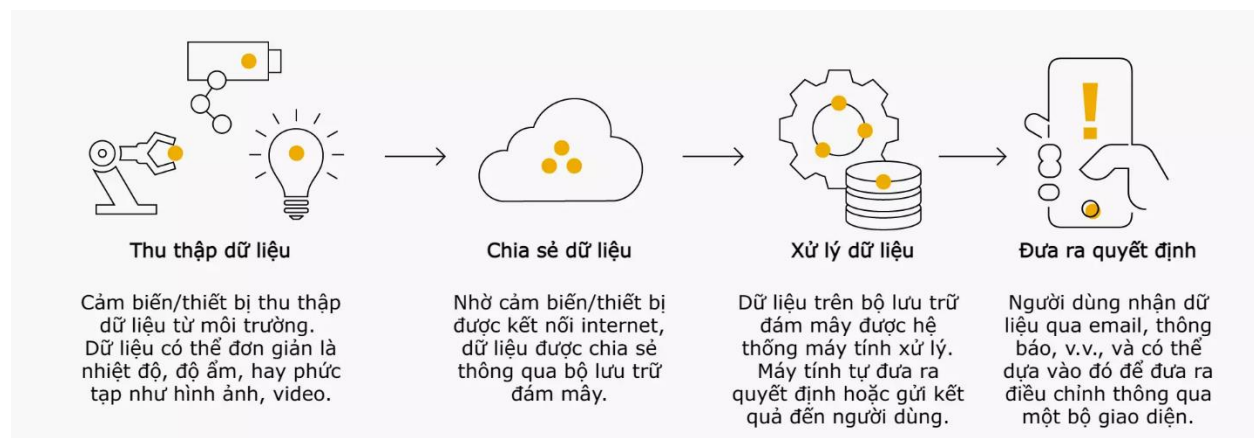
Quản lý dữ liệu:

- Hệ thống lưu trữ: Là nơi lưu trữ dữ liệu thu thập từ các thiết bị IoT, bao gồm cả dữ liệu thô và dữ liệu đã được xử lý.
- Phân tích dữ liệu: Sử dụng các công nghệ phân tích dữ liệu như trí tuệ nhân tạo, học máy để phân tích và rút ra thông tin hữu ích từ dữ liệu IoT, hỗ trợ ra quyết định và cải thiện hiệu suất.

Bằng cách tích hợp các thành phần này, một hệ thống IoT hoạt động hiệu quả, cung cấp thông tin quan trọng và giúp tối ưu hóa quy trình và trải nghiệm người dùng.

1.3. Mô hình hoạt động của hệ thống IoT

Mọi hệ thống IoT hoàn chỉnh đều có 4 bước: Thu thập, chia sẻ, xử lý dữ liệu và đưa ra quyết định. Hình 1.1 và các mục dưới đây sẽ trình bày rõ cách hoạt động của một mô hình IoT.



Hình 1.1: Mô hình hoạt động của hệ thống IoT (nguồn: vietnix.vn)

1.3.1. Thu thập dữ liệu

Các cảm biến IoT được triển khai để thu thập dữ liệu từ môi trường xung quanh hoặc từ các thiết bị khác. Ví dụ, cảm biến nhiệt độ đo nhiệt độ không khí, cảm biến độ ẩm đo độ ẩm, cảm biến ánh sáng đo cường độ ánh sáng, cảm biến chuyển động phát hiện chuyển động. Dữ liệu được thu thập thường là các giá trị số hoặc tín hiệu điện tử.

1.3.2. Truyền tải dữ liệu

Có hai loại kết nối truyền tải dữ liệu: Kết nối không dây: Sử dụng các giao thức như Wi-Fi, Bluetooth, Zigbee, LoRa, NB-IoT, và mạng di động (4G/5G); Kết nối có dây: Sử dụng Ethernet hoặc các mạng công nghiệp như Modbus, Profibus.

Vai trò của Gateway: Gateway là thiết bị trung gian kết nối các cảm biến với mạng internet, chuyển đổi giao thức và đảm bảo an ninh khi truyền tải dữ liệu.

1.3.3. Xử lý dữ liệu ban đầu

Xử lý sơ bộ: Bao gồm lọc nhiễu, nén dữ liệu, và thực hiện các phép tính cơ bản để giảm lượng dữ liệu phải truyền tải lên đám mây.

Phản hồi nhanh: Một số hành động có thể được thực hiện ngay tại thiết bị edge mà không cần chờ dữ liệu được truyền tải và xử lý trên đám mây. Ví dụ, kích hoạt cảnh báo nếu phát hiện khói từ cảm biến khói.

1.3.4. Lưu trữ và phân tích dữ liệu

Lưu trữ dữ liệu: Dữ liệu từ các cảm biến được truyền lên nền tảng đám mây để lưu trữ lâu dài và dễ dàng truy xuất.

Phân tích dữ liệu: Các công cụ và thuật toán phân tích dữ liệu, bao gồm machine learning và AI, được áp dụng để trích xuất thông tin hữu ích, phát hiện xu hướng và dự đoán tương lai.

1.3.5. Hành động dựa trên dữ liệu

Dựa trên kết quả phân tích dữ liệu, các hành động cụ thể được thực hiện bởi bộ truyền động. Có thể là đóng/mở van, bật/tắt thiết bị, điều khiển động cơ, và nhiều hành động khác. Ví dụ, nếu nhiệt độ vượt ngưỡng cho phép, hệ thống sẽ ra lệnh cho máy điều hòa bật lên.

1.3.6. Giao tiếp với người dùng

Ứng dụng di động và web: Người dùng có thể truy cập và tương tác với hệ thống thông qua các ứng dụng di động hoặc giao diện web. Họ có thể theo dõi trạng thái của các thiết bị, nhận thông báo, hoặc điều khiển thiết bị từ xa.

Dashboard: Giao diện trực quan hiển thị dữ liệu thời gian thực và kết quả phân tích, giúp người dùng dễ dàng hiểu và ra quyết định.

1.3.7. Quản lý và bảo trì

1.3.7.a. Quản lý thiết bị

Giám sát thiết bị: Theo dõi tình trạng hoạt động của các thiết bị IoT để đảm bảo chúng hoạt động ổn định.

Cập nhật phần mềm: Thực hiện cập nhật phần mềm từ xa để cải thiện hiệu suất và bảo mật của thiết bị.

1.3.7.b. Bảo mật

Mã hóa dữ liệu: Bảo vệ dữ liệu trong quá trình truyền tải và lưu trữ bằng các phương pháp mã hóa.

Xác thực người dùng: Đảm bảo chỉ có người dùng được ủy quyền mới có thể truy cập và điều khiển hệ thống.

1.4. Ứng dụng của IoT trong lĩnh vực ghi hình qua internet

IoT đã có những ứng dụng đáng kể trong lĩnh vực ghi hình qua internet, cung cấp những giải pháp hiệu quả cho việc giám sát, bảo vệ, và quản lý các vị trí và hoạt động khác nhau. Dưới đây là một số ứng dụng tiêu biểu:

Hệ thống giám sát an ninh:

- Camera kết nối IoT: Các camera an ninh kết nối internet cho phép người dùng giám sát và quản lý nhà ở, văn phòng, cửa hàng, hoặc các vị trí khác từ xa thông qua điện thoại di động hoặc máy tính.
- Cảm biến và báo động: Các cảm biến chuyển động, cửa sổ, cửa ra vào kết hợp với IoT để cung cấp thông báo báo động và bảo vệ an ninh.

Giám sát hạ tầng công cộng:

- Giám sát đường sắt và đường cao tốc: Các thiết bị IoT được sử dụng để giám sát và bảo dưỡng hạ tầng giao thông quan trọng như đường sắt và đường cao tốc, giúp phát hiện và ngăn chặn sự cố.

- Theo dõi môi trường: Các cảm biến IoT được triển khai để giám sát chất lượng không khí, môi trường nước, và các thảm họa tự nhiên khác.

Ghi hình và phân tích dữ liệu:

- Ghi hình thông minh: Hệ thống ghi hình thông minh sử dụng IoT để tự động ghi lại các sự kiện quan trọng, phát hiện và nhận diện đối tượng, giúp giảm thiểu tải công việc của nhân viên giám sát.
- Phân tích dữ liệu video: Công nghệ AI và học máy được tích hợp vào hệ thống để phân tích dữ liệu video, nhận dạng hành vi bất thường, và cung cấp thông tin chi tiết về hoạt động diễn ra.

Giám sát môi trường và năng lượng:

- Cảm biến IoT cho năng lượng và nguồn nước: Các cảm biến được triển khai để giám sát tiêu thụ năng lượng, dự báo nhu cầu và tối ưu hóa sử dụng nguồn nước.
- Theo dõi tài nguyên tự nhiên: Các thiết bị IoT giúp giám sát và bảo vệ tài nguyên tự nhiên như rừng, hồ, và đất đai.

Trong tổng thể, IoT mang lại những ứng dụng đa dạng và tích hợp trong lĩnh vực ghi hình qua internet, tạo ra một môi trường an toàn, thông minh, và hiệu quả cho các hoạt động giám sát và quản lý.

CHƯƠNG 2: CÁC CÔNG NGHỆ LIÊN QUAN ĐẾN XÂY DỰNG HỆ THỐNG GHI HÌNH QUA INTERNET

2.1. Công nghệ truyền thông và mạng: TCP/IP, HTTP

Trong xây dựng hệ thống ghi hình qua Internet, việc sử dụng các công nghệ truyền thông và mạng là vô cùng quan trọng để đảm bảo việc truyền tải dữ liệu giữa các thiết bị một cách tin cậy và hiệu quả. Dưới đây là mô tả chi tiết về các công nghệ này:

TCP/IP (Transmission Control Protocol/Internet Protocol): TCP/IP là một bộ giao thức mạng phổ biến được sử dụng trong Internet và các mạng LAN. Giao thức TCP (Transmission Control Protocol) quản lý việc truyền tải dữ liệu giữa các thiết bị một cách tin cậy bằng cách chia dữ liệu thành các gói nhỏ và xác nhận nhận được dữ liệu. Giao thức IP (Internet Protocol) định địa chỉ cho các thiết bị trên mạng và định tuyến dữ liệu giữa chúng. Trong hệ thống ghi hình qua Internet, TCP/IP đảm bảo việc truyền tải dữ liệu video từ camera IP đến máy chủ hoặc điện thoại di động một cách an toàn và tin cậy.

HTTP (Hypertext Transfer Protocol): HTTP là một giao thức truyền tải dữ liệu dựa trên văn bản, được sử dụng để truy cập và truyền tải thông tin trên World Wide Web. Trong hệ thống ghi hình qua Internet, HTTP thường được sử dụng để điều khiển và truy cập các thiết bị như camera IP thông qua các giao diện người dùng trực tuyến hoặc ứng dụng di động. HTTP cung cấp một cách tiếp cận đơn giản và linh hoạt cho việc quản lý và điều khiển các thiết bị ghi hình từ xa thông qua giao diện web hoặc ứng dụng.

2.2. Công nghệ cảm biến và thiết bị hình ảnh: Camera IP

Trong hệ thống ghi hình qua Internet, việc sử dụng các thiết bị hình ảnh đóng vai trò quan trọng trong việc thu thập dữ liệu hình ảnh từ môi trường xung quanh và truyền tải nó qua mạng Internet.

Camera IP là loại camera có khả năng kết nối trực tiếp với mạng Internet thông qua giao thức IP.

Đặc điểm chính của camera IP bao gồm độ phân giải cao, khả năng ghi hình và truyền tải dữ liệu video một cách linh hoạt và hiệu quả, cũng như khả năng quản lý và kiểm soát từ xa thông qua các ứng dụng di động hoặc giao diện web.

Trong hệ thống ghi hình qua Internet, camera IP được sử dụng để giám sát và ghi lại các sự kiện quan trọng, bảo vệ an ninh và giúp quản lý từ xa các vị trí và hoạt động.

2.3. Giới thiệu về Framework Python Flask

2.3.1. Ngôn ngữ lập trình Python

2.3.1.a. Khái niệm

Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, phát triển phần mềm, khoa học dữ liệu và máy học (Machine Learning). Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Phần mềm Python được tải xuống miễn phí, tích hợp tốt với tất cả các loại hệ thống và tăng tốc độ phát triển.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

2.3.1.b. Ưu điểm

Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo.

2.3.1.c. Lợi ích

- Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
- Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
- Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
- Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
- Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
- Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.

– Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.

2.3.1.d. Một số thư viện của Python

Requests: Thư viện Requests cung cấp nhiều hàm hữu ích cần thiết để phát triển web. Bạn có thể sử dụng thư viện này để gửi các yêu cầu HTTP, bổ sung tiêu đề, thêm tham số URL, bổ sung dữ liệu và thực hiện nhiều tác vụ khác khi giao tiếp với các ứng dụng web.

OpenCV-Python: OpenCV-Python là một thư viện mà các nhà phát triển sử dụng để xử lý hình ảnh cho các ứng dụng thị giác máy tính. Thư viện này cung cấp nhiều hàm cho các tác vụ xử lý hình ảnh như đọc và ghi hình ảnh cùng lúc, xây dựng môi trường 3D từ môi trường 2D cũng như chụp và phân tích hình ảnh từ video.

Firebase-admin: Firebase-admin là thư viện cho phép tương tác với Firebase từ các ứng dụng phía server bằng Python. Thư viện này cung cấp các API để truy cập và quản lý nhiều dịch vụ của Firebase như Authentication, Firestore, Realtime Database, Cloud Storage, và Cloud Messaging.

2.3.1.e. Các framework phổ biến

Django: Django là một trong những framework web Python full-stack được sử dụng rộng rãi nhất để phát triển các ứng dụng web trên quy mô lớn. Framework này cung cấp một số tính năng hữu ích, bao gồm một máy chủ web để phát triển và kiểm thử, một công cụ mẫu để xây dựng giao diện người dùng của trang web và nhiều cơ chế bảo mật khác.

Flask: Flask là một framework micro được sử dụng để phát triển các ứng dụng web nhỏ. Các đặc điểm của framework này bao gồm sự hỗ trợ nhiệt tình từ cộng đồng, tài liệu hữu ích, công cụ mẫu, tính năng kiểm thử đơn vị và máy chủ web được tích hợp sẵn. Nó cũng cung cấp các tiện ích mở rộng để hỗ trợ xác thực, các lớp ánh xạ cơ sở dữ liệu và bảo mật web.

TurboGears: TurboGears là một framework được thiết kế để xây dựng các ứng dụng web một cách nhanh chóng và dễ dàng hơn. Sau đây là một số đặc điểm nổi bật của framework này:

- Cấu trúc bảng cơ sở dữ liệu cụ thể
- Các công cụ tạo và quản lý dự án
- Một công cụ mẫu để xây dựng cơ sở dữ liệu

- Một công cụ mẫu để tạo giao diện người dùng
- Các cơ chế bảo mật web

Apache MXNet: Apache MXNet là một framework học sâu nhanh, linh hoạt và có thể điều chỉnh quy mô được các nhà phát triển sử dụng để xây dựng những nguyên mẫu nghiên cứu và ứng dụng học sâu. Framework này hỗ trợ nhiều ngôn ngữ lập trình, bao gồm Java, C++, R và Perl. Nó cung cấp một bộ công cụ cũng như thư viện phong phú để hỗ trợ phát triển. Ví dụ: bạn có thể tìm thấy một cuốn sách tương tác về máy học (ML), bộ công cụ thị giác máy tính và các mô hình học sâu dành cho Xử lý ngôn ngữ tự nhiên (NLP) để xử lý ngôn ngữ tự nhiên, chẳng hạn như văn bản và lời nói.

PyTorch: PyTorch là một framework cho ML được xây dựng dựa trên thư viện Torch, một thư viện ML nguồn mở khác. Các nhà phát triển sử dụng framework này cho những ứng dụng như NLP, robot và thị giác máy tính, tìm kiếm thông tin có ý nghĩa từ hình ảnh và video. Họ cũng sử dụng PyTorch để chạy những ứng dụng đó bằng CPU và GPU.

2.3.2. Framework Flask

2.3.2.a. Đặc điểm nổi bật

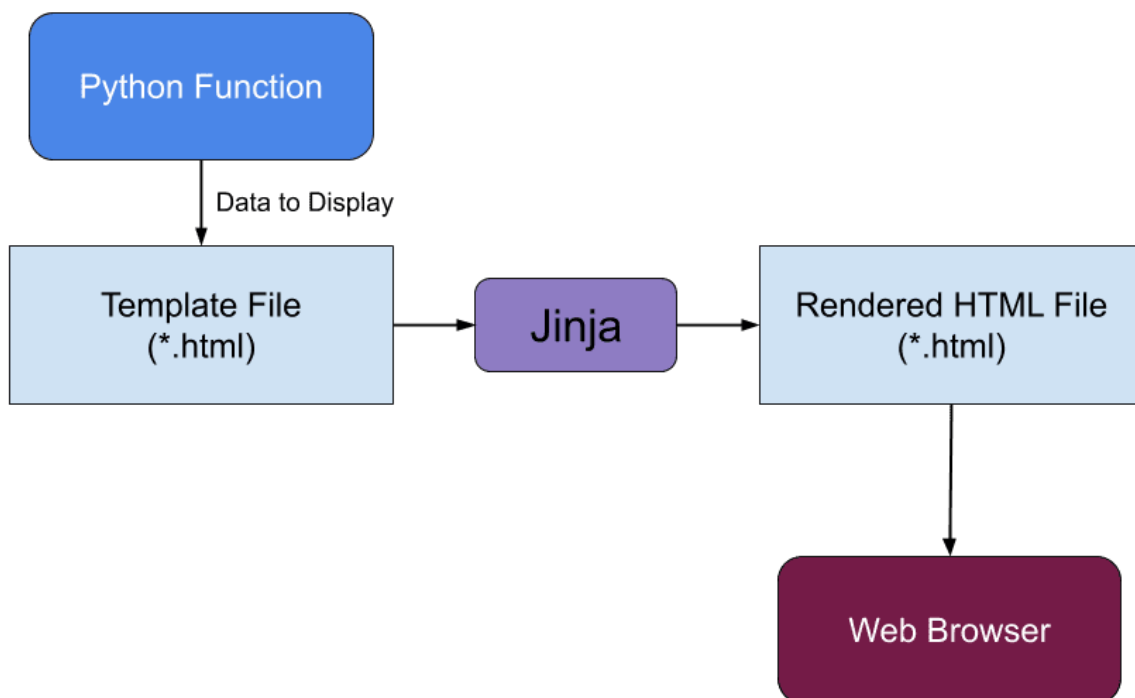
Flask là một framework web siêu nhẹ được viết bằng Python, được sử dụng để phát triển các ứng dụng web đơn giản và nhanh chóng. Flask được phát triển dựa trên nguyên tắc "Keep It Simple, Stupid" (KISS), tập trung vào sự đơn giản và dễ sử dụng. Dưới đây là một số điểm nổi bật của Flask:

- **Linh hoạt**: Flask cung cấp một cách tiếp cận linh hoạt và tùy chỉnh cho việc phát triển ứng dụng web. Bạn có thể tự do chọn lựa các thành phần và thư viện mà bạn muốn sử dụng, không bị ràng buộc bởi cấu trúc cụ thể.
- **Dễ học và sử dụng**: Flask có cú pháp đơn giản và dễ hiểu, giúp cho việc học và sử dụng framework trở nên dễ dàng đối với người mới bắt đầu và người có kinh nghiệm trong lập trình Python.
- **Tiêu chuẩn WSGI**: Flask tuân thủ tiêu chuẩn WSGI (Web Server Gateway Interface), cho phép tích hợp dễ dàng với các máy chủ web khác nhau như Apache, Nginx hoặc Gunicorn.

- Thư viện mở rộng phong phú: Flask đi kèm với một loạt các thư viện mở rộng (extensions) cho các tính năng như xác thực người dùng, xử lý biểu mẫu, và tương tác với cơ sở dữ liệu.
- Tích hợp tốt với các công nghệ khác: Flask dễ dàng tích hợp với các công nghệ khác như SQLAlchemy cho việc làm việc với cơ sở dữ liệu, Jinja2 cho mô-đun hóa HTML, và WTForms cho xử lý biểu mẫu.
- Hỗ trợ mạnh mẽ từ cộng đồng: Flask có một cộng đồng lớn và năng động, cung cấp nhiều tài liệu, ví dụ và hỗ trợ cho những người sử dụng.

Flask là một lựa chọn tuyệt vời cho việc phát triển các ứng dụng web đơn giản, nhưng cũng đủ mạnh mẽ để xây dựng các ứng dụng phức tạp hơn. Điểm mạnh của Flask là tính linh hoạt và sự đơn giản, giúp cho việc phát triển ứng dụng web trở nên nhanh chóng và dễ dàng.

2.3.2.b. Mô hình hoạt động



Hình 2.1: Mô hình hoạt động của Flask (nguồn: topdev.vn)

Hình 2.1 mô tả hoạt động của một ứng dụng web Python sử dụng framework Flask như sau:

- Hàm Python xử lý yêu cầu và chuẩn bị dữ liệu.
- Hàm chuyển dữ liệu này tới mẫu Jinja.
- Jinja kết xuất mẫu với dữ liệu được cung cấp, tạo một tài liệu HTML hoàn chỉnh.
- HTML được hiển thị sẽ được gửi lại cho máy khách (trình duyệt web), tệp này sẽ hiển thị trang web cho người dùng.

2.3.2.c. Ví dụ cơ bản về một hàm Python Flask

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home():
    return 'Hello, Flask!'
if __name__ == '__main__':
    app.run(debug=True)
```

Giải thích:

- `from flask import Flask`: Nhập lớp Flask từ thư viện Flask.
- `app = Flask(name)`: Tạo một đối tượng Flask mới.
- `@app.route('/')`: Định nghĩa một route (đường dẫn) cho trang chủ ('/').
- `def home(): return 'Hello, Flask!'`: Định nghĩa một hàm xử lý khi người dùng truy cập vào trang chủ, trả về chuỗi 'Hello, Flask!'.
- `if name == 'main': app.run(debug=True)`: Chạy ứng dụng Flask với chế độ debug (giúp hiển thị lỗi và tự động reload khi có thay đổi trong mã nguồn).

2.4. Arduino và module ESP32

2.4.1. Arduino

Arduino là một nền tảng mã nguồn mở bao gồm cả phần cứng và phần mềm, được thiết kế để giúp mọi người dễ dàng tạo ra các dự án điện tử. Nó bao gồm các bảng mạch vi điều khiển và một môi trường phát triển tích hợp (IDE) để viết mã và tải lên mạch.

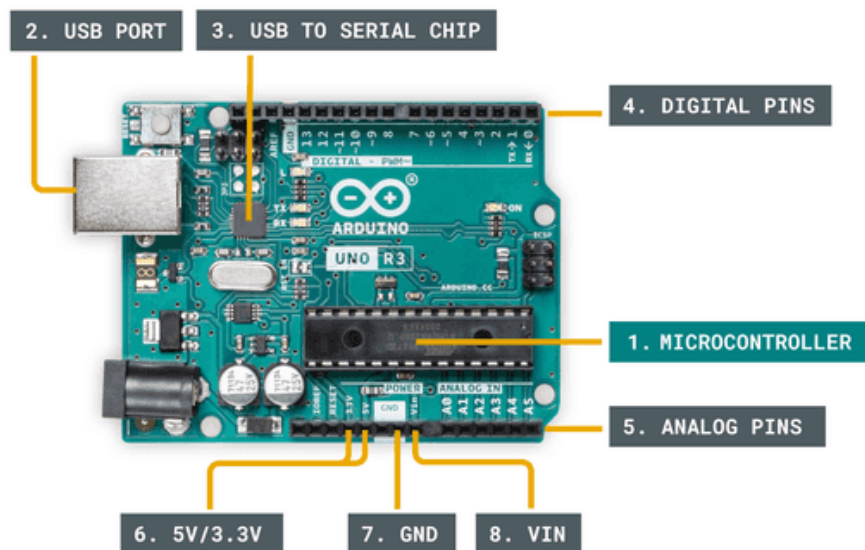
2.4.1.a. Phần cứng Arduino

Phần cứng của Arduino là các bo mạch vi điều khiển với nhiều loại khác nhau, phổ biến nhất là Arduino Uno. Các bo mạch này có nhiều chân I/O (input/output) kỹ thuật số

và analog, cho phép chúng kết nối và điều khiển các linh kiện như cảm biến, đèn LED, động cơ, và nhiều thiết bị khác.

Các thành phần chính của một bo mạch Arduino (được mô tả ở hình 2.2):

- Microcontroller - đây là bộ não của Arduino và là thành phần mà chúng ta tải chương trình vào. Hãy nghĩ về nó như một chiếc máy tính nhỏ, được thiết kế để chỉ thực hiện một số công việc cụ thể.
- USB port - được sử dụng để kết nối bo mạch Arduino của bạn với máy tính.
- USB to Serial chip- USB to Serial là một thành phần quan trọng, vì nó giúp dịch dữ liệu từ máy tính sang bộ vi điều khiển trên bo mạch. Đây là điều giúp bạn có thể lập trình bo mạch Arduino từ máy tính của mình.
- Digital pins - chân sử dụng logic kỹ thuật số (0,1 hoặc THẤP/CAO). Thường được sử dụng cho các công tắc và bật/tắt đèn LED.
- Analog pins - chân có thể đọc các giá trị analog ở độ phân giải 10 bit (0-1023).
- 5V/3.3V pins - những chân này dùng để cấp nguồn cho các linh kiện bên ngoài.
- GND – còn gọi là ground, negative hoặc đơn giản -, được sử dụng để hoàn thành một mạch điện, trong đó mức điện ở mức 0 volt.
- VIN - viết tắt của Volt In, nơi bạn có thể kết nối các nguồn điện bên ngoài.



Hình 2.2: Thành phần chính của bo mạch Arduino (nguồn: docs.arduino.cc)

2.4.1.b. API Arduino

API Arduino, hay còn gọi là "Ngôn ngữ lập trình Arduino", bao gồm một số hàm, biến và cấu trúc dựa trên ngôn ngữ C/C++.

API Arduino có thể được chia thành ba phần chính: functions, variables and structure:

- Functions: để điều khiển bo mạch Arduino và thực hiện tính toán. Ví dụ: để đọc hoặc ghi trạng thái vào chân kỹ thuật số, ánh xạ một giá trị hoặc sử dụng giao tiếp nối tiếp.
- Variables: hằng số, kiểu dữ liệu và chuyển đổi. Ví dụ: int, boolean, array.
- Structure: các thành phần của mã Arduino (C++), chẳng hạn như: sketch (loop(), setup()); cấu trúc điều khiển (if, else, while, for); các toán tử số học (nhân, cộng, trừ); toán tử so sánh (==, !=, <, >)

API Arduino có thể được mô tả là sự đơn giản hóa ngôn ngữ lập trình C++, với nhiều bổ sung để điều khiển phần cứng Arduino.

2.4.1.c. Phần mềm Arduino

Một phần không thể thiếu khác của hệ sinh thái Arduino là các công cụ phần mềm của nó. Để lập trình cho bo mạch của bạn, bạn cần viết một chương trình, biên dịch chương trình đó thành mã máy và cuối cùng: gửi chương trình mới đến bo mạch của bạn.

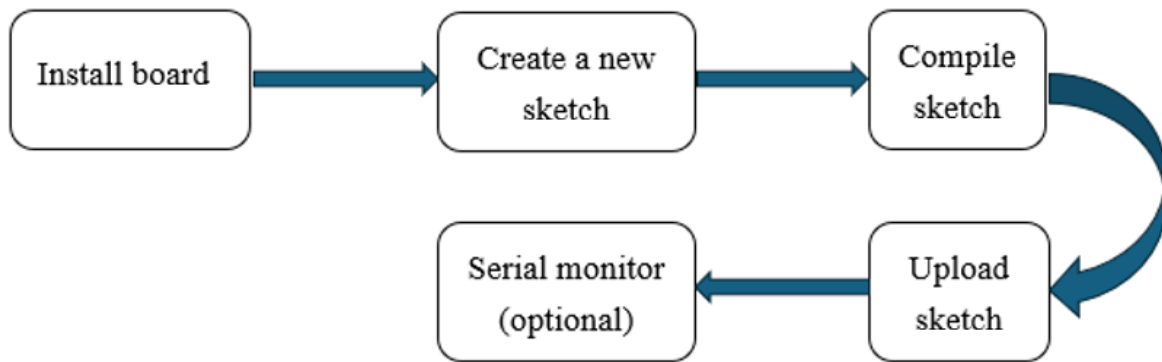
Arduino IDE tạo điều kiện thuận lợi cho tất cả điều này, từ dòng mã đầu tiên được viết, cho đến khi nó được thực thi trên bộ vi điều khiển của bo mạch Arduino. Đó là một chương trình hoặc ứng dụng mà bạn có thể tải xuống (hoặc sử dụng phiên bản trực tuyến) để quản lý tất cả quá trình phát triển mã của mình. Trước đây, đây là một quá trình phức tạp, đòi hỏi phải có kiến thức tốt về điện tử và khoa học máy tính. Giờ đây, bất kỳ ai cũng có thể học cách thực hiện điều đó với sự trợ giúp của Arduino IDE.

Hiện nay, có một số loại Arduino IDE có sẵn: phần mềm Arduino IDE 2, Arduino online web

Quy trình tải mã nguồn vào bo mạch arduino (hình 2.3)

– Lắp đặt bo mạch: điều này có nghĩa là cài đặt đúng "gói" cho bo mạch. Nếu không có gói, chúng ta không thể sử dụng bảng của mình. Việc cài đặt được thực hiện trực tiếp trong IDE và là một thao tác nhanh chóng và dễ dàng.

- Tạo một sketch mới: sketch là tệp chương trình chính. Ở đây cần viết một bộ chương trình mà chúng ta muốn thực hiện trên bộ vi điều khiển.
- Biên dịch sketch: là mã sketch đã chuyển đổi thành tệp nhị phân (1 và 0). Nếu có lỗi xảy ra, chúng ta sẽ nhận được lỗi này trong bảng điều khiển lỗi.
- Upload sketch: sau khi quá trình biên dịch thành công, mã có thể được tải lên bảng mạch. Trong bước này, cần kết nối bo mạch với máy tính một cách vật lý và chọn cổng nối tiếp phù hợp.
- Màn hình nối tiếp (tùy chọn): đối với hầu hết các dự án Arduino, điều quan trọng là phải biết điều gì đang diễn ra trên bo mạch. Công cụ Serial Monitor có sẵn trong tất cả các IDE cho phép gửi dữ liệu từ bo mạch đến máy tính.



Hình 2.3: Quy trình làm việc để tải mã nguồn vào bo mạch arduino

2.4.2. Module ESP32

2.4.2.a. Tổng quan

Bên cạnh sự xuất hiện phổ biến của module wifi esp8266, module esp32 xuất hiện trở thành lựa chọn tiếp theo dành cho người dùng. ESP32 là một vi điều khiển tích hợp khả năng kết nối Wi-Fi và Bluetooth, được phát triển bởi Espressif Systems. Nó là một giải pháp mạnh mẽ và linh hoạt cho các ứng dụng IoT (Internet of Things) và các thiết bị nhúng, nổi bật với hiệu suất cao và chi phí thấp. Dưới đây là cấu hình của ESP32

CPU

CPU: Xtensa Dual-Core LX6 microprocessor.

- Chạy hệ 32 bit
- Tốc độ xử lý 160MHZ up to 240 MHz

- Tốc độ xung nhịp đọc flash chip 40mhz --> 80mhz (tùy chỉnh khi lập trình)
- RAM: 520 KByte SRAM liền chip – (trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

Hỗ trợ 2 giao tiếp không dây

- Wi-Fi: 802.11 b/g/n/e/i
- Bluetooth: v4.2 BR/EDR and BLE

Hỗ trợ tất cả các loại giao tiếp

- 8-bit DACs(digital to analog) 2 cổng
- Analog(ADC) 12-bit 16 cổng.
- I²C – 2 cổng
- UART – 3 cổng
- SPI – 3 cổng (1 cổng cho chip FLASH)
- I²S – 2 cổng
- SD card /SDIO/MMC host
- Slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN bus 2.0
- IR (TX/RX)
- Băm xung PWM (tất cả các chân)
- Ultra low power analog pre-amplifier'

Cảm biến tích hợp trên chip esp32

- 1 cảm biến Hall (cảm biến từ trường)
- 1 cảm biến đo nhiệt độ
- Cảm biến chạm (điện dung) với 10 đầu vào khác nhau.

Bảo mật

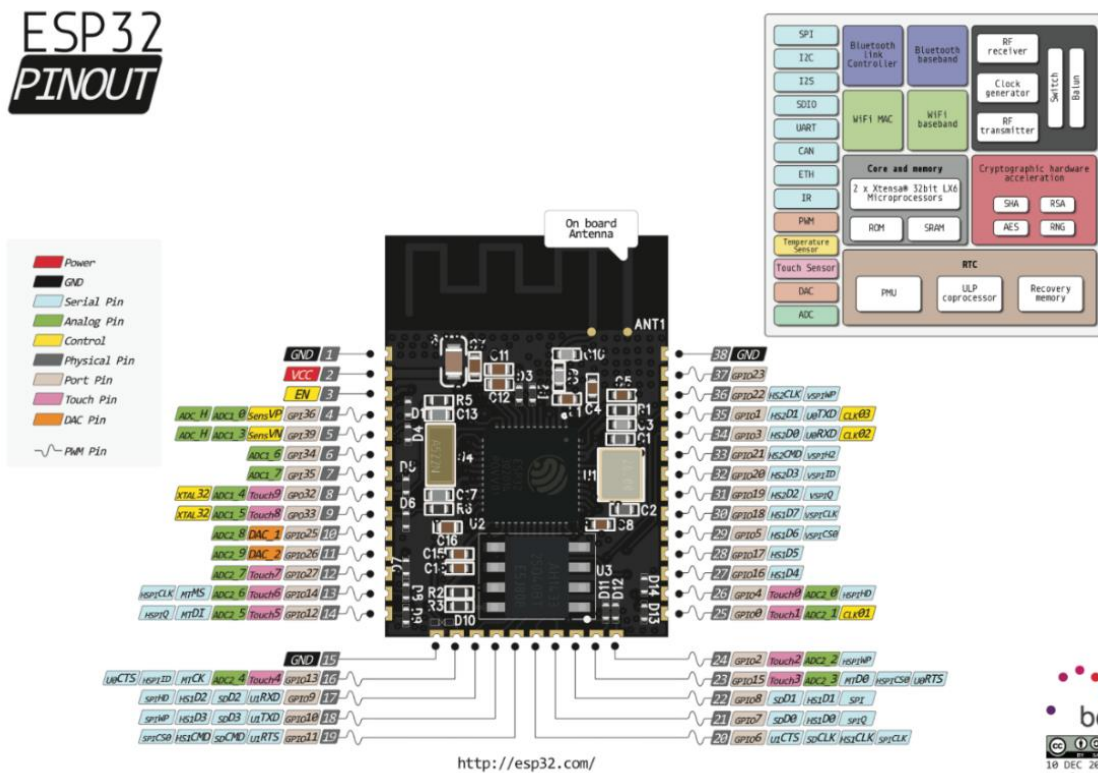
- IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI

- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

Nguồn điện hoạt động

- Nhiệt độ hoạt động $-40 + 85^{\circ}\text{C}$
- Điện áp hoạt động: 2.2-3.6V
- Số cổng GPIOs : 34

Với esp8266, cùng với wifi, esp32 hỗ trợ thêm truyền nhận Bluetooth, RAM nhiều hơn, Tốc độ xử lý nhanh hơn, số chân GPIO nhiều hơn, nhiều cổng giao tiếp hơn, nhiều chân PWM hơn, nhiều chân ADC hơn, tích hợp cả 3 loại cảm biến (nhiệt độ, hall, touch sensor)... Tất cả ưu điểm đó cũng đủ khiến người dùng của ESP hài lòng ngay từ cái nhìn đầu tiên.

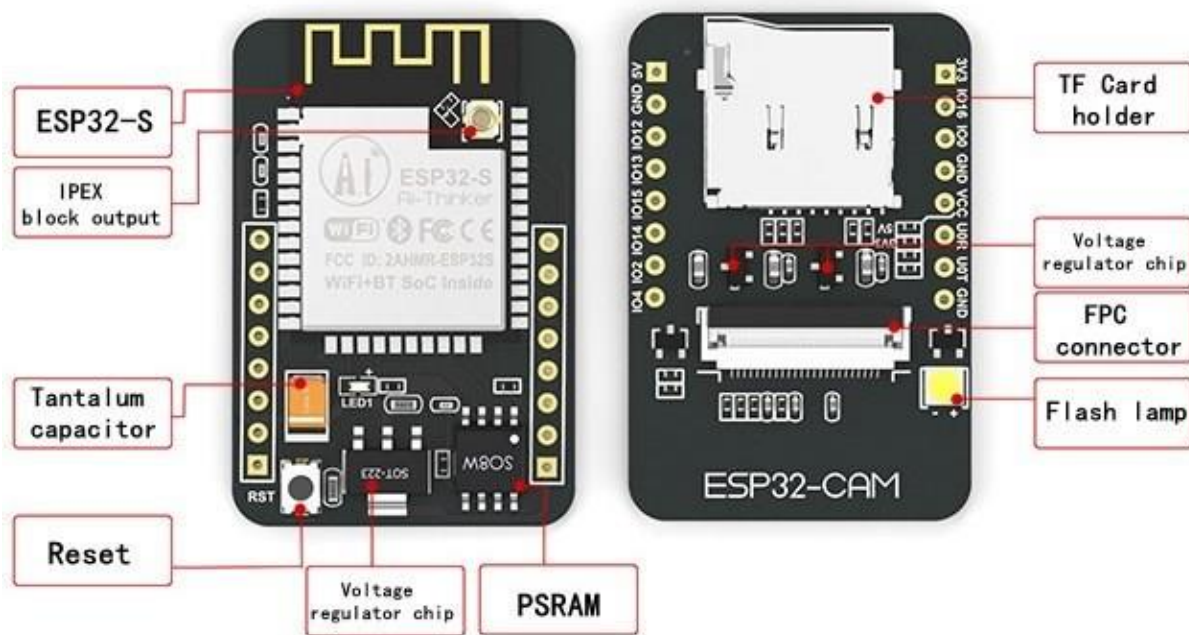


Hình 2.4: Mapping cho một module chip esp32 (nguồn: arduino.vn)

2.4.2.b. ESP32-CAMERA

ESP32-CAM có một camera kích thước nhỏ, rất cạnh tranh trong ngành, giống như mô-đun chính, mô-đun này có thể được xử lý công việc độc lập, module có kích thước nhỏ gọn chỉ 40 x 27 x12 mm, dòng nghỉ chỉ 6mA.

ESP32-CAM có thể được sử dụng rộng rãi trong các ứng dụng IoT khác nhau, thích hợp cho thiết bị thông minh gia đình, điều khiển không dây công nghiệp, giám sát không dây kiểm soát, nhận dạng không dây QR, tín hiệu hệ thống định vị không dây... Nó là một giải pháp lý tưởng cho các ứng dụng IoT



Hình 2.5: ESP32-CAM (nguồn: mecsu.vn)

Mạch thu phát Wifi BLE ESP32 này (hình 2.5) là mạch chính hãng AI – Thinker có chất lượng độ ổn định và độ bền rất cao, sử dụng camera OV2640 chất lượng cao hình ảnh sắc nét, không nhiễu sọc, không xảy ra tình trạng treo khi hoạt động do sử dụng ic cấp nguồn chất lượng cao.

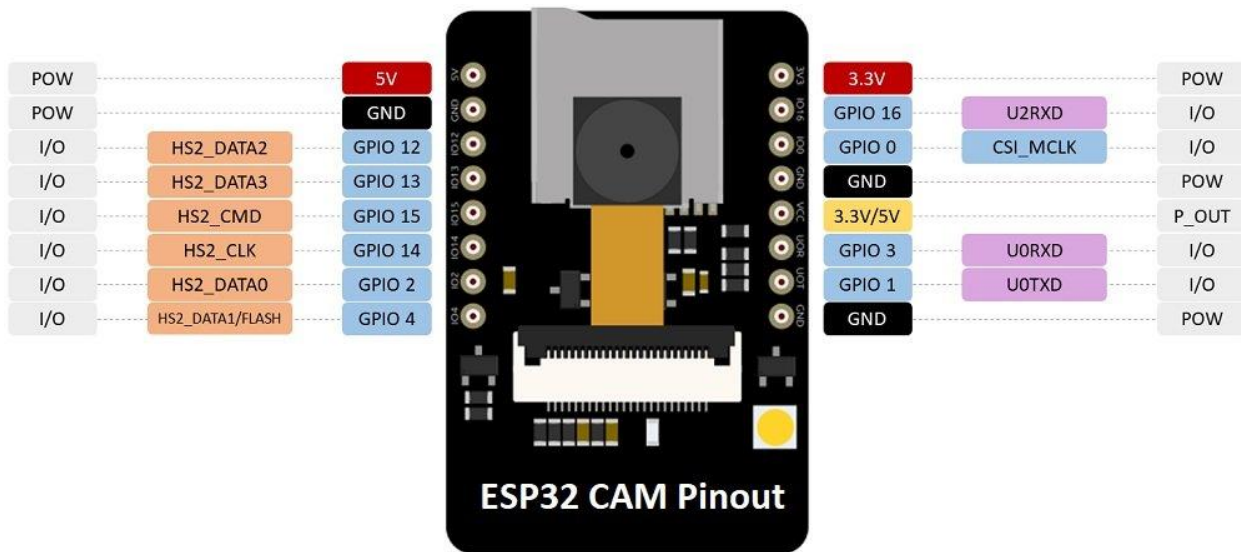
Mạch thu phát Wifi BLE ESP32-CAM Ai-Thinker có thể sử dụng Arduino IDE để biên dịch và viết code, được hỗ trợ mạnh mẽ từ cộng đồng.

Thông số cơ bản của Esp32-Cam:

- IC chính: ESP32-S (AI-Thinker)
- Mô-đun Wi-Fi BT SoC 802.11 b/g/n/e/i

- CPU 32-bit công suất thấp, cũng có thể phục vụ bộ xử lý ứng dụng
- Tốc độ đồng hồ lên đến 160MHz, sức mạnh tính toán lên đến 600 DMIPS
- Tích hợp 520 KB SRAM, 4MPSRAM bên ngoài
- Dải tần số: 1421 ~ 2484 Mhz
- Bluetooth: 4.2 BR/EDR BLE
- Hỗ trợ UART / SPI / I2C / PWM / ADC / DAC
- Hỗ trợ máy ảnh OV2640 và OV7670, đèn flash tích hợp
- Hỗ trợ tải lên WiFi hình ảnh
- Hỗ trợ thẻ TF
- Hỗ trợ nhiều chế độ ngủ
- Nhúng Lwip và FreeRTOS
- Hỗ trợ chế độ hoạt động STA / AP / STA + AP
- Hỗ trợ cấu hình thông minh / công nghệ AirKiss
- Hỗ trợ nâng cấp cục bộ và từ xa cho công nối tiếp (FOTA)

Hình ảnh sau đây mô tả sơ đồ chân của thiết bị ESP32-CAM:



Hình 2.6: Sơ đồ chân ESP32-CAM (nguồn: mecsu.vn)

2.5. Công nghệ đám mây: Firebase

2.5.1. Tổng quan về Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Nó cung cấp một loạt các dịch vụ đám mây, từ việc lưu trữ và cơ sở dữ liệu đến xác thực người dùng và phân tích, giúp cho việc phát triển và triển khai các ứng dụng trở nên dễ dàng và hiệu quả. Dưới đây là một số điểm nổi bật của Firebase:

- *Realtime database*: Firebase cung cấp một cơ sở dữ liệu thời gian thực cho phép đồng bộ dữ liệu giữa các thiết bị một cách tức thì. Dữ liệu được cập nhật ngay lập tức khi có thay đổi, giúp cho việc phát triển các ứng dụng đa nền tảng trở nên dễ dàng hơn.

- *Authentication*: Firebase Authentication cung cấp các phương thức xác thực người dùng an toàn và dễ dàng. Người dùng có thể đăng nhập bằng email, số điện thoại, hoặc tài khoản mạng xã hội như Google, Facebook, và Twitter.

- *Cloud firestore*: Firestore là một cơ sở dữ liệu linh hoạt cho phép lưu trữ và truy vấn dữ liệu từ bất kỳ thiết bị nào. Nó hỗ trợ cấu trúc dữ liệu linh hoạt và cho phép truy vấn dữ liệu một cách hiệu quả.

- *Cloud functions*: Firebase Cloud Functions cho phép bạn viết và triển khai các hàm backend mà không cần quản lý các máy chủ. Điều này giúp cho việc xử lý logic phía server trở nên linh hoạt và dễ dàng hơn.

- *Hosting*: Firebase Hosting cung cấp một cách dễ dàng và nhanh chóng để triển khai các ứng dụng web và tĩnh. Bạn có thể triển khai các trang web chỉ trong vài phút mà không cần phải quản lý cơ sở hạ tầng.

- *Analytics*: Firebase Analytics cung cấp các công cụ phân tích mạnh mẽ giúp bạn hiểu rõ hơn về cách người dùng tương tác với ứng dụng của bạn. Bạn có thể theo dõi các sự kiện, người dùng và thu thập dữ liệu phân tích để cải thiện trải nghiệm của người dùng.

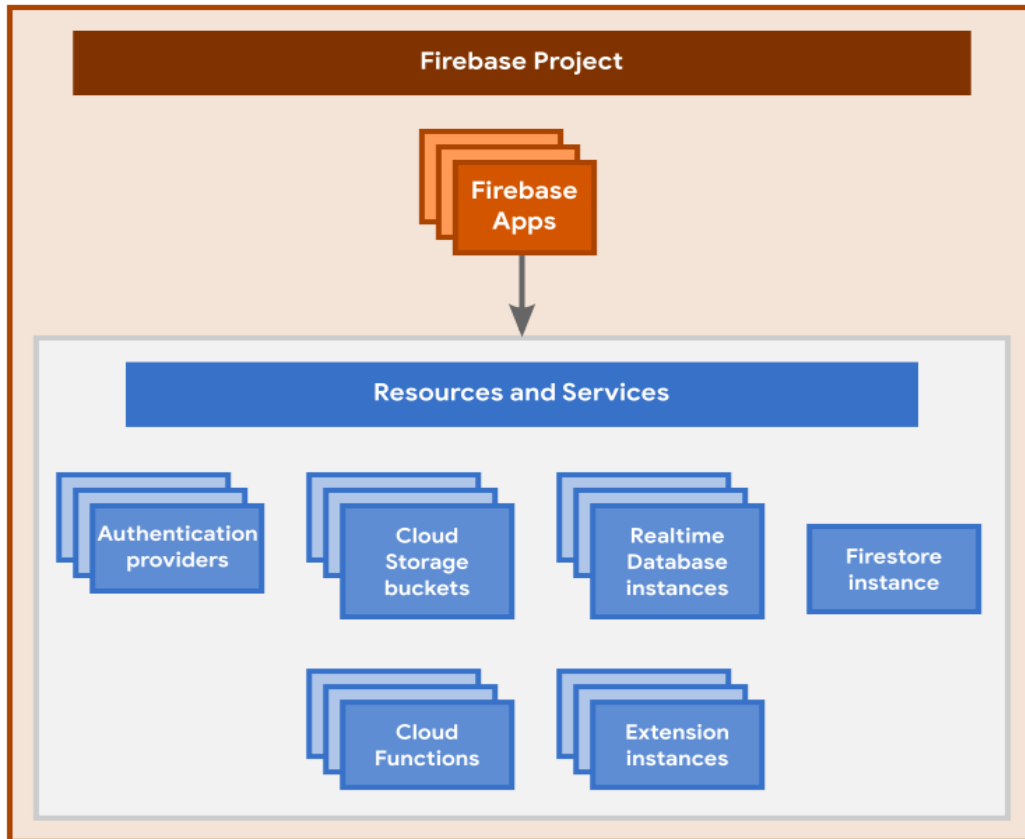
Firebase là một nền tảng đáng tin cậy và mạnh mẽ cho việc phát triển ứng dụng di động và web. Nó cung cấp một loạt các dịch vụ cần thiết để xây dựng, triển khai và quản lý các ứng dụng của bạn một cách dễ dàng và hiệu quả.

2.5.2. Mối quan hệ giữa các dự án, ứng dụng và sản phẩm Firebase

Dự án Firebase là thực thể cấp cao nhất của Firebase. Trong một dự án, bạn có thể đăng ký các ứng dụng Apple, Android hoặc web của mình. Sau khi đăng ký ứng dụng của

mình với Firebase, bạn có thể thêm SDK Firebase cho bất kỳ số lượng sản phẩm Firebase nào, như Analytics, Cloud Firestore, Giám sát hiệu suất hoặc Cấu hình từ xa.

Hệ thống phân cấp của các dự án Firebase:



Hình 2.7: Mối quan hệ giữa các dự án, ứng dụng và sản phẩm Firebase
(nguồn: firebase.google.com)

Sơ đồ này hiển thị hệ thống phân cấp cơ bản của dự án Firebase. Dưới đây là các mối quan hệ chính:

- Dự án Firebase giống như một nơi chứa tất cả ứng dụng của bạn cũng như mọi tài nguyên và dịch vụ được cung cấp cho dự án.
- Một dự án Firebase có thể có một hoặc nhiều ứng dụng Firebase được đăng ký với dự án đó (ví dụ: cả phiên bản iOS và Android của một ứng dụng hoặc cả phiên bản miễn phí và trả phí của một ứng dụng).

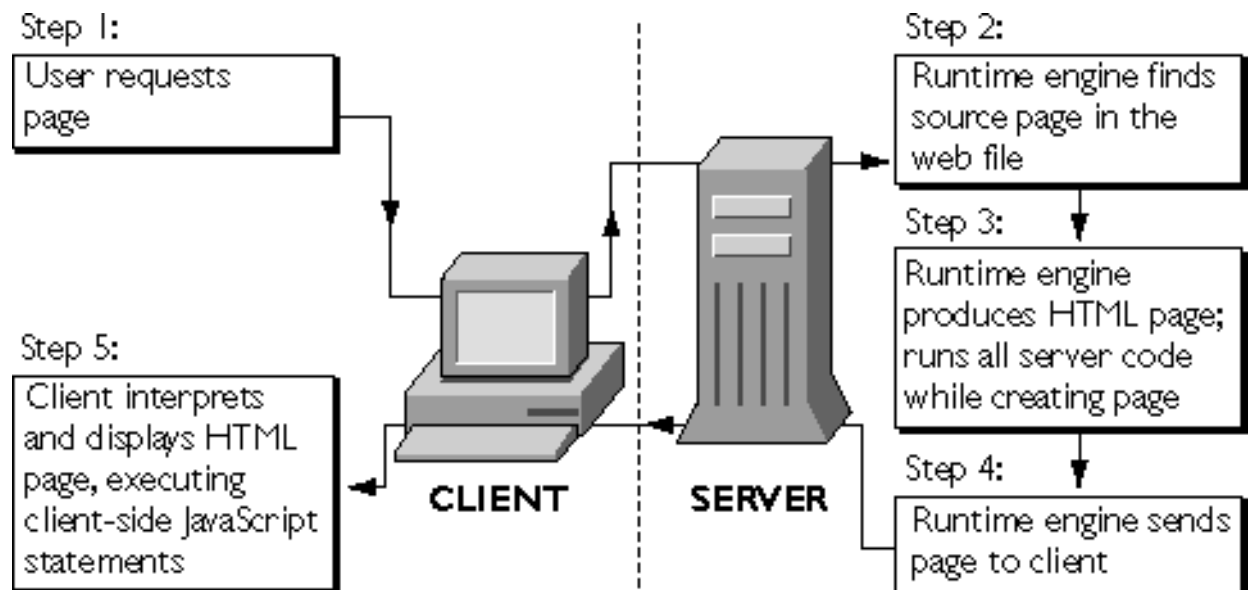
- Tất cả ứng dụng Firebase đã đăng ký cùng chia sẻ dự án Firebase và có quyền truy cập vào tất cả các tài nguyên cũng như dịch vụ giống nhau được cung cấp cho dự án.

2.6. Ngôn ngữ lập trình JavaScript

2.6.1. Khái niệm

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Từ làm mới bảng tin trên trang mạng xã hội đến hiển thị hình ảnh động và bản đồ tương tác, các chức năng của JavaScript có thể cải thiện trải nghiệm người dùng của trang web. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web. Ví dụ: khi duyệt internet, bất cứ khi nào bạn thấy quảng cáo quay vòng dạng hình ảnh, menu thả xuống nhấp để hiển thị hoặc màu sắc phần tử thay đổi động trên trang web cũng chính là lúc bạn thấy các hiệu ứng của JavaScript.

2.6.2. Cách thức hoạt động



Hình 2.8: Xử lý yêu cầu trang JavaScript (nguồn: *Server-Side JavaScript Guide Chapter 12*, url: <https://s.pro.vn/jZUQ>)

JavaScript phía máy khách: JavaScript phía máy khách đề cập đến cách thức JavaScript hoạt động trong trình duyệt của bạn. Trong trường hợp này, công cụ JavaScript nằm trong mã trình duyệt. Tất cả các trình duyệt web phổ biến đều có các công cụ JavaScript tích hợp riêng. Các nhà phát triển ứng dụng web viết mã JavaScript với các hàm

khác nhau được liên kết với nhiều loại sự kiện, chẳng hạn như nhấp chuột hoặc khi di chuột đến. Các hàm này thực hiện các thay đổi đối với HTML và CSS.

Sau đây là tổng quan về cách thức hoạt động của JavaScript phía máy khách:

- Trình duyệt tải một trang web khi bạn truy cập trang đó.
- Trong khi tải, trình duyệt chuyển đổi trang và tất cả các phần tử của trang, chẳng hạn như các nút, nhãn và hộp thả xuống, thành một cấu trúc dữ liệu được gọi là Mô hình đối tượng tài liệu (DOM).
- Công cụ JavaScript của trình duyệt chuyển đổi mã JavaScript thành mã bytecode. Mã này đóng vai trò là trung gian giữa cú pháp JavaScript và máy.
- Các sự kiện khác nhau, chẳng hạn như nhấp vào nút, sẽ kích hoạt việc thực thi khối mã JavaScript được liên kết. Sau đó, công cụ này sẽ diễn giải bytecode và thực hiện các thay đổi đối với DOM.
- Trình duyệt hiển thị DOM mới.

JavaScript phía máy chủ: JavaScript phía máy chủ đề cập đến việc sử dụng ngôn ngữ viết mã trong logic máy chủ back-end. Trong trường hợp này, công cụ JavaScript nằm trực tiếp trên máy chủ. Hàm JavaScript phía máy chủ có thể truy cập cơ sở dữ liệu, thực hiện các phép toán logic khác nhau và phản hồi nhiều loại sự kiện do hệ điều hành của máy chủ kích hoạt. Ưu điểm chính của viết tập lệnh phía máy chủ là bạn có khả năng tùy chỉnh linh hoạt phản hồi của trang web dựa trên yêu cầu và quyền truy cập của mình cũng như theo yêu cầu thông tin từ trang web.

So sánh giữa phía máy khách và phía máy chủ: Từ linh hoạt đúng với cả JavaScript phía máy khách và phía máy chủ. Hành vi linh hoạt là khả năng cập nhật phần hiển thị của trang web để tạo nội dung mới theo yêu cầu. Sự khác biệt giữa JavaScript phía máy khách và phía máy chủ nằm ở cách chúng tạo ra nội dung mới. Mã phía máy chủ tạo nội dung mới một cách linh hoạt bằng cách sử dụng logic ứng dụng và sửa đổi dữ liệu từ cơ sở dữ liệu. Mặt khác, JavaScript phía máy khách tạo nội dung mới một cách linh hoạt bên trong trình duyệt thông qua sử dụng logic giao diện người dùng và sửa đổi nội dung trang web đã có trên máy khách. Ý nghĩa hơi khác biệt ở hai ngữ cảnh nhưng có sự liên quan đến nhau và cả hai cách tiếp cận này đều cùng nhau nâng cao trải nghiệm người dùng.

Ngoài việc triển khai các tính năng linh hoạt, một điểm khác biệt nữa giữa hai cách sử dụng JavaScript là ở tài nguyên mà mã có thể truy cập. Về phía máy khách, trình duyệt

kiểm soát môi trường thời gian chạy của JavaScript. Mã chỉ có thể truy cập những tài nguyên mà trình duyệt cho phép truy cập. Ví dụ: mã không thể ghi nội dung vào ổ cứng của bạn trừ khi bạn nhấp vào nút tải xuống. Mặt khác, các hàm phía máy chủ có thể truy cập tất cả các tài nguyên của máy chủ khi cần thiết.

2.6.3. Ưu nhược điểm của JavaScript

2.6.3.a. Ưu điểm

- *Dễ học và sử dụng*: Javascript rất phù hợp với các bạn mới học lập trình. Bởi chúng thật sự dễ học và sử dụng ngay nếu bạn chú tâm. Trong đó, cấu trúc cú pháp của Javascript đơn giản và cấu trúc mã nguồn dễ hiểu.

- *Đa năng và linh hoạt cao*: Javascript có thể được sử dụng cho nhiều mục đích khác nhau. Từ việc tạo ra các tính năng tương tác trên trang web cho đến phát triển các ứng dụng web động.

- *Hoạt động trên nhiều nền tảng*: Javascript hoạt động trên nhiều nền tảng khác nhau, bao gồm máy tính để bàn, điện thoại thông minh, máy tính bảng,...

- *Khả năng phát hiện lỗi nhanh chóng*: Các lỗi của Javascript rất dễ phát hiện, điều này giúp bạn sửa các lỗi nhanh và dễ dàng hơn.

- *Khả năng tương tác với HTML và CSS*: Javascript có thể tương tác với các phần tử HTML và CSS trên trang web. Các trình duyệt web có thể dịch Javascript bằng HTML mà không cần một compiler.

- *Hệ thống giao diện phong phú*: Drag and Drop, Slider để cung cấp một Rich Interface.

2.6.3.b. Nhược điểm

- *Dễ bị tấn công bảo mật*: Javascript có thể bị tấn công bảo mật trên trang web từ các hacker và scammer. Chẳng hạn như việc xâm nhập vào máy chủ hoặc đánh cắp dữ liệu người dùng. Mã nguồn có thể dễ dàng bị xem và sao chép bởi người dùng.

- *Tốc độ xử lý chậm trên các thiết bị cũ*: Javascript có thể chạy chậm trên các thiết bị cũ hoặc trình duyệt không được cập nhật mới nhất. Từ đó, khiến trải nghiệm người dùng trở nên chậm chạp.

- *Khả năng tương thích và chuẩn hóa*: JavaScript có sự khác biệt trong cách mà các trình duyệt khác nhau thực hiện JavaScript.

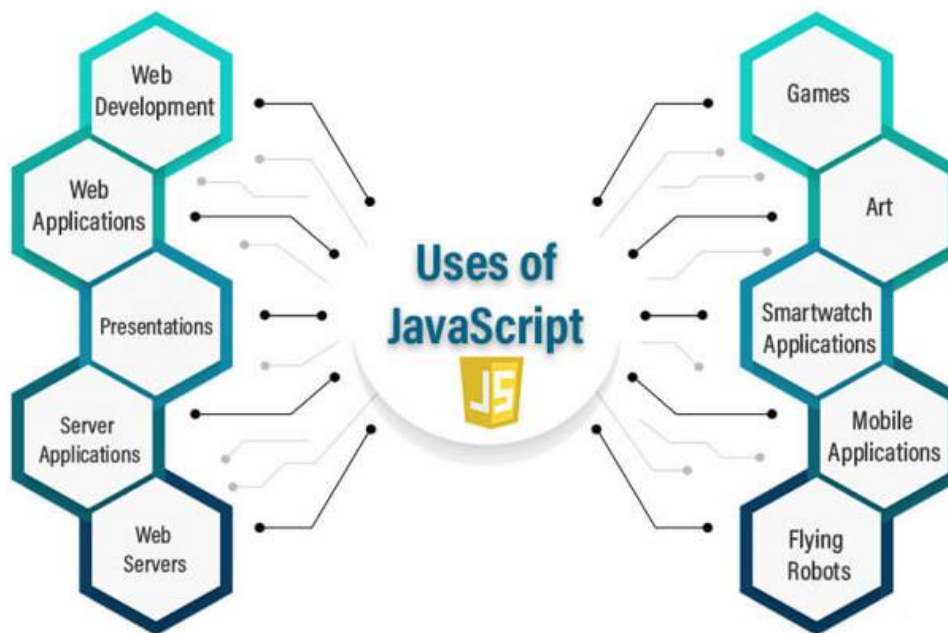
– *Thiếu một số tính năng nâng cao*: So với các ngôn ngữ lập trình khác, JavaScript thiếu một số tính năng nâng cao như xử lý đa luồng (multi-threading) hiệu quả.

– *Thiếu kiểu dữ liệu mạnh mẽ*: JavaScript có một hệ thống kiểu dữ liệu lỏng lẻo, dễ gây ra các lỗi runtime do lỗi kiểu dữ liệu.

Client-side Javascript không cho phép đọc và ghi các file do các vấn đề về bảo mật.

2.6.4. Ứng dụng

JavaScript là một ngôn ngữ lập trình đa năng với nhiều ứng dụng trong các lĩnh vực khác nhau. Dưới đây là thông tin chi tiết về một số sử dụng chính của JavaScript:



Hình 2.9: Các ứng dụng của JavaScript (nguồn: educba.com)

2.6.4.a. Web Development

JavaScript là ngôn ngữ cốt lõi cho phát triển web frontend, cung cấp khả năng tương tác động cho các trang web. Với JavaScript, các nhà phát triển có thể thay đổi nội dung và cấu trúc của HTML trực tiếp từ mã, quản lý các sự kiện người dùng như nhấp chuột, di chuột, và nhập liệu, cũng như sử dụng AJAX để tải dữ liệu từ máy chủ mà không cần tải lại toàn bộ trang. Điều này tạo ra trải nghiệm người dùng mượt mà và linh hoạt hơn, giúp tăng cường sự hấp dẫn và chức năng của các trang web hiện đại.

2.6.4.b. Web Applications

JavaScript là nền tảng của nhiều ứng dụng web, đặc biệt là các ứng dụng một trang (SPA) và ứng dụng web tiên bộ (PWA). Các framework như React, Angular, và Vue.js cho phép các nhà phát triển xây dựng các ứng dụng web phức tạp, có khả năng phản hồi nhanh và giao diện người dùng mượt mà. PWAs mang lại trải nghiệm người dùng gần giống như ứng dụng gốc, với khả năng hoạt động offline và tải nhanh hơn, nhờ vào các công nghệ như Service Workers và Web App Manifests.

2.6.4.c. Presentations

JavaScript có thể được sử dụng để tạo các bài thuyết trình tương tác và động, thay thế cho các công cụ truyền thống như PowerPoint. Các thư viện như Reveal.js và Impress.js cho phép tạo ra các bài thuyết trình web với các hiệu ứng đẹp mắt và điều hướng sáng tạo, sử dụng HTML, CSS và JavaScript. Điều này mang lại sự linh hoạt và sáng tạo trong việc trình bày thông tin, phù hợp cho các buổi họp, hội thảo và các sự kiện giáo dục.

2.6.4.d. Server Applications

JavaScript không chỉ giới hạn ở phía client mà còn mạnh mẽ ở phía server nhờ vào Node.js. Với Node.js, các nhà phát triển có thể viết mã server-side, tạo ra các ứng dụng web backend hiệu quả và có khả năng mở rộng cao. Express.js, một framework web tối giản cho Node.js, giúp xây dựng các API và ứng dụng web một cách nhanh chóng và dễ dàng, từ đó tạo ra các dịch vụ web linh hoạt và mạnh mẽ.

2.6.4.e. Web Servers

JavaScript, thông qua Node.js, có thể được sử dụng để xây dựng các máy chủ web hiệu quả. Các nhà phát triển có thể tạo ra các máy chủ HTTP đơn giản hoặc phức tạp, cũng như các ứng dụng thời gian thực sử dụng WebSockets và thư viện như Socket.io. Điều này cho phép xây dựng các ứng dụng chat, trò chơi trực tuyến và các dịch vụ tương tác thời gian thực một cách dễ dàng và hiệu quả.

2.6.4.f. Games

JavaScript là một công cụ mạnh mẽ để phát triển game 2D và 3D, cả trên trình duyệt và các nền tảng khác. Các framework như Phaser cho phép tạo ra các game 2D hấp dẫn, trong khi Three.js và Babylon.js cung cấp các công cụ để phát triển các game 3D và ứng dụng đồ họa phức tạp. JavaScript mang lại khả năng tạo ra các trò chơi trực tuyến phong phú và tương tác, dễ dàng tiếp cận người chơi trên nhiều thiết bị.

2.6.4.g. *Art*

JavaScript có thể được sử dụng để tạo ra các tác phẩm nghệ thuật kỹ thuật số tương tác, mở ra những khả năng sáng tạo mới. Các thư viện như p5.js và Processing.js cho phép các nghệ sĩ và nhà lập trình tạo ra các tác phẩm nghệ thuật và hình ảnh động phong phú. Những công cụ này hỗ trợ các dự án nghệ thuật thị giác và các ứng dụng sáng tạo, kết hợp lập trình với nghệ thuật để tạo ra những trải nghiệm độc đáo và thú vị.

2.6.4.h. *Smartwatch Applications*

JavaScript có thể được sử dụng để phát triển ứng dụng cho các thiết bị đeo tay như smartwatch. Các công cụ như Pebble.js cho phép viết mã JavaScript để phát triển ứng dụng cho đồng hồ thông minh Pebble, trong khi hệ điều hành Tizen của Samsung hỗ trợ phát triển ứng dụng bằng JavaScript cho các thiết bị đeo tay. Điều này mở rộng khả năng của các nhà phát triển để tạo ra các ứng dụng tiện ích và giải trí trên các thiết bị đeo tay hiện đại.

2.6.4.i. *Mobile Applications*

JavaScript có thể được sử dụng để phát triển các ứng dụng di động đa nền tảng, giúp tiết kiệm thời gian và nguồn lực cho các nhà phát triển. React Native là một framework phổ biến cho phép phát triển ứng dụng di động cho cả iOS và Android bằng JavaScript, mang lại hiệu suất gần giống như ứng dụng gốc. Apache Cordova (PhoneGap) cũng cho phép sử dụng HTML, CSS và JavaScript để phát triển các ứng dụng di động, giúp các nhà phát triển tận dụng kỹ năng web của họ để tạo ra các ứng dụng di động mạnh mẽ.

2.6.4.k. *Flying Robots*

JavaScript có thể được sử dụng để lập trình và điều khiển các thiết bị bay như drone. Các thư viện như NodeCopter.js cho phép sử dụng JavaScript để điều khiển drone AR.Drone, trong khi Johnny-Five cung cấp các công cụ để điều khiển nhiều loại phần cứng khác nhau, bao gồm cả drone và robot. Điều này mở ra những khả năng mới cho việc phát triển các ứng dụng điều khiển từ xa, khám phá và tự động hóa trong nhiều lĩnh vực.

CHƯƠNG 3: CÁC TIÊU CHUẨN VÀ QUY ĐỊNH LIÊN QUAN

3.1. Tiêu chuẩn kỹ thuật về IoT và ghi hình qua internet

Trong lĩnh vực IoT và ghi hình qua internet, việc thiết lập và tuân thủ các tiêu chuẩn kỹ thuật là vô cùng quan trọng để đảm bảo tính tương thích, an toàn và hiệu suất của các hệ thống. Dưới đây là một số điểm cần được trình bày rõ:

– *Giao thức truyền tải dữ liệu*: Các hệ thống IoT và ghi hình qua internet thường sử dụng các giao thức truyền tải dữ liệu như MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protocol), và HTTP (Hypertext Transfer Protocol). Việc tuân thủ các tiêu chuẩn về giao thức giúp đảm bảo tính tương thích và hiệu suất của hệ thống.

– *Định dạng dữ liệu*: Để truyền tải và xử lý dữ liệu một cách hiệu quả, các hệ thống IoT và ghi hình cần tuân thủ các tiêu chuẩn về định dạng dữ liệu. Ví dụ, JSON (JavaScript Object Notation) thường được sử dụng làm định dạng dữ liệu cho việc truyền tải thông tin giữa các thiết bị và máy chủ.

– *Giao diện người dùng*: Giao diện người dùng của các ứng dụng IoT và ghi hình qua internet cũng cần tuân thủ các tiêu chuẩn về trải nghiệm người dùng và tương tác. Các tiêu chuẩn về giao diện người dùng giúp đảm bảo tính thân thiện và dễ sử dụng của hệ thống.

– *Bảo mật và quyền riêng tư*: Các tiêu chuẩn kỹ thuật cũng cần tập trung vào bảo mật và quyền riêng tư của dữ liệu. Việc áp dụng các phương pháp bảo mật như mã hóa dữ liệu, xác thực người dùng và kiểm soát truy cập giúp đảm bảo rằng thông tin của người dùng được bảo vệ một cách an toàn.

– *Tích hợp và mở rộng*: Cuối cùng, các tiêu chuẩn kỹ thuật cần tạo điều kiện cho tính linh hoạt và tính mở rộng của hệ thống. Điều này đảm bảo rằng các hệ thống có thể tích hợp dễ dàng với các công nghệ và thiết bị khác, cũng như mở rộng để đáp ứng nhu cầu người dùng trong tương lai.

Việc tuân thủ các tiêu chuẩn kỹ thuật đảm bảo rằng các hệ thống IoT và ghi hình qua internet được triển khai và sử dụng một cách an toàn, hiệu quả và tương thích.

3.2. Quy định pháp lý và an ninh thông tin

Trong lĩnh vực sử dụng IoT và ghi hình qua internet, quy định pháp lý và an ninh thông tin đóng vai trò quan trọng trong việc đảm bảo tính bảo mật, quyền riêng tư và tuân thủ pháp luật. Dưới đây là một số điểm cần được trình bày rõ:

- *Bảo vệ dữ liệu cá nhân*: Các hệ thống IoT và ghi hình qua internet cần tuân thủ các quy định về bảo vệ dữ liệu cá nhân, như GDPR (General Data Protection Regulation) ở châu Âu hoặc các quy định tương tự ở các quốc gia khác. Điều này đảm bảo rằng thông tin cá nhân của người dùng được bảo vệ và xử lý một cách đúng đắn.

- *Quy định về bảo mật mạng*: Các tổ chức cần tuân thủ các quy định về bảo mật mạng để đảm bảo rằng hệ thống của họ được bảo vệ khỏi các mối đe dọa mạng như tấn công từ chối dịch vụ (DDoS), tấn công tin nhắn giả mạo (Phishing), và các cuộc tấn công khác.

- *Quy định về sử dụng và lưu trữ dữ liệu*: Các quy định về sử dụng và lưu trữ dữ liệu đặt ra các nguyên tắc và hướng dẫn cụ thể về việc thu thập, lưu trữ, xử lý và chia sẻ thông tin. Điều này đảm bảo rằng dữ liệu được sử dụng một cách hợp pháp và minh bạch.

- *Tuân thủ pháp luật về an ninh thông tin*: Các tổ chức cần tuân thủ các quy định pháp lý và tiêu chuẩn về an ninh thông tin như ISO 27001 để đảm bảo rằng hệ thống của họ được bảo vệ khỏi các mối đe dọa và rủi ro liên quan đến an ninh thông tin.

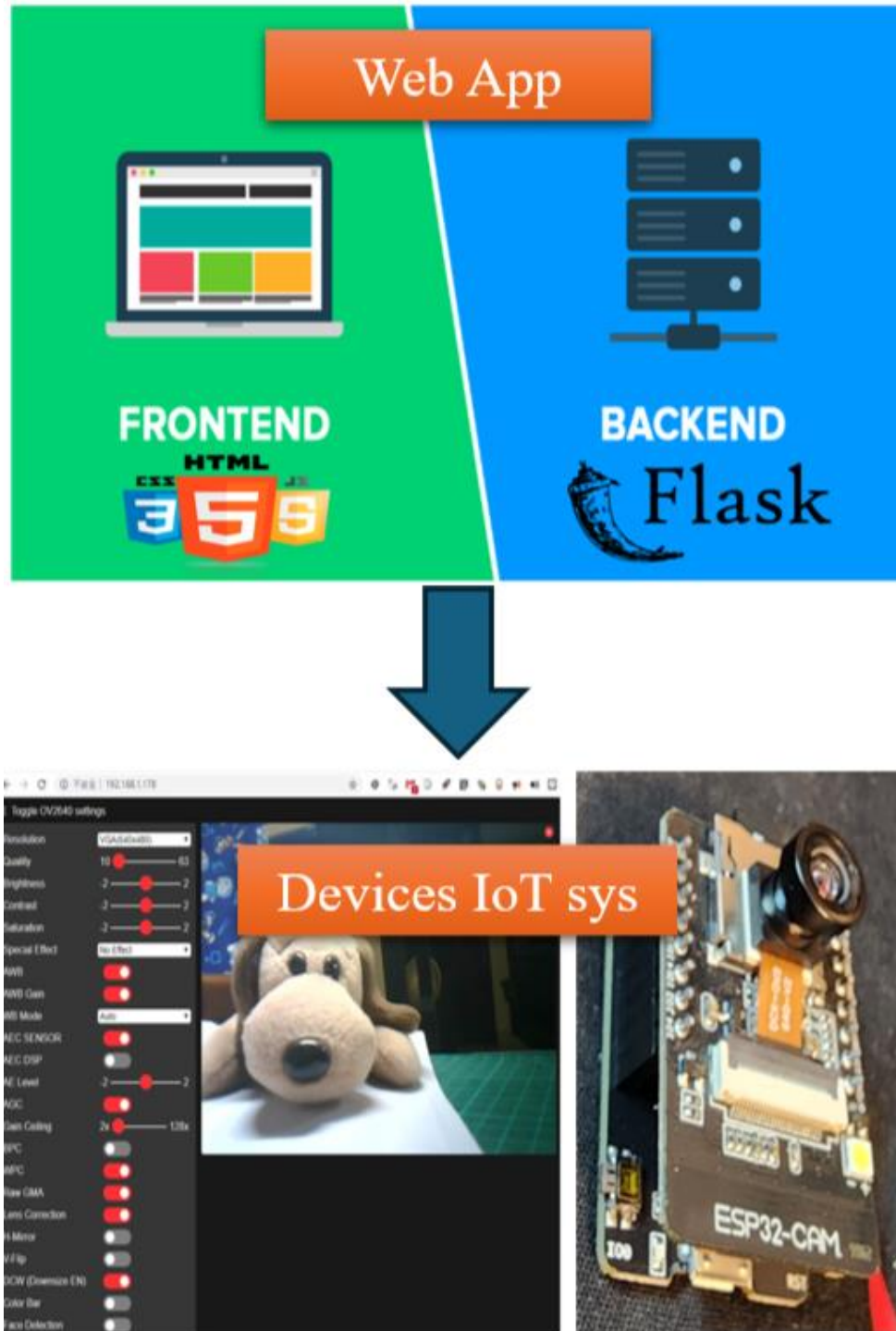
- *Tuân thủ quy định về trách nhiệm pháp lý*: Cuối cùng, các tổ chức cần tuân thủ các quy định về trách nhiệm pháp lý, bao gồm việc cung cấp thông tin và bảo vệ quyền lợi của người dùng. Điều này bao gồm việc cung cấp thông tin rõ ràng về việc thu thập và sử dụng dữ liệu, cũng như giải quyết các khiếu nại từ phía người dùng một cách công bằng và hiệu quả.

Tuân thủ các quy định pháp lý và an ninh thông tin giúp đảm bảo rằng các hệ thống IoT và ghi hình qua internet được triển khai và sử dụng một cách an toàn, bảo mật và hợp pháp.

CHƯƠNG 4: XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG

4.1. Xây dựng hệ thống ghi hình qua internet

Kiến trúc tổng quan hệ thống ghi hình qua Internet được trình bày theo ảnh sau:



Hình 4.1: Kiến trúc tổng quan của hệ thống ghi hình qua Internet

4.1.1. Các yêu cầu bài toán

Mục đích của đề tài là xây dựng một hệ thống ghi hình qua Internet, cho phép nhân viên an ninh trong khách sạn, chung cư có thể theo dõi được tình hình ra vào của khách hàng, nhân viên và nhiều đối tượng khác để tăng cường tình trạng an ninh của những nơi này tốt hơn. Tất nhiên là hệ thống có thể sử dụng trong một số nhiệm vụ khác hay một số nhu cầu khác theo người sử dụng, vì vậy tôi xin trình bày một số yêu cầu kỹ thuật cơ bản ở bên dưới đây.

Hệ thống ghi hình qua internet có những yêu cầu sau:

- Thực hiện hiển thị được luồng stream thông qua ESP32-Cam.
- Có chức năng bật/tắt ghi hình để có thể ghi lại video theo thời gian thực.
- Có chức năng chụp ảnh màn hình stream.
- Hiển thị danh sách các file được lưu trên firebase.
- Lưu video (hoặc ảnh chụp màn hình) đã quay vào local, có thể upload lên firebase và có thể download được video lưu trên firebase về local.

4.1.2. Các thiết bị yêu cầu

Module ESP32-Camera OV2640: Kit RF Thu Phát Wifi ESP32 Camera có kích thước nhỏ gọn với bộ xử lý chính là module ESP32 + Camera OV2640 được sử dụng trong các ứng dụng truyền hình ảnh, xử lý ảnh qua Wifi, Bluetooth hoặc các ứng dụng IoT, mạch có chất lượng gia công tốt, độ bền cao. Chi phí sử dụng cho thiết bị này là: 130.000 VNĐ

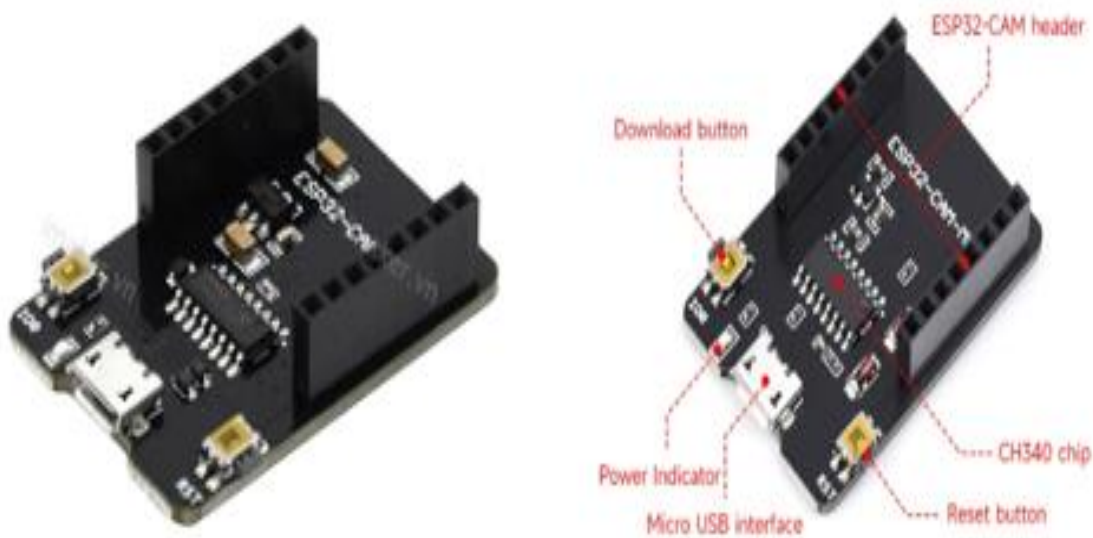


Hình 4.2: Module ESP32-CAM OV2640 (nguồn: imaker.vn)

Để nạp chương trình ESP32-CAM USB programming adapter: Để nạp chương trình ESP32-CAM USB Programming Adapter giúp mạch ESP32-CAM nạp chương trình và giao tiếp truyền dữ liệu với máy tính qua giao IC chuyển USB-UART CH340 một cách dễ dàng, để còn tích hợp hai nút nhấn cho chân RST và IO0 của ESP32-CAM để sử dụng trong quá trình nạp chương trình. Chi phí sử dụng cho thiết bị này là: 40.000 VNĐ

Thông số kỹ thuật:

- Model: ESP32-CAM-MB
- Sử dụng để nạp chương trình và giao tiếp truyền dữ liệu với máy tính cho mạch ESP32-CAM.
- Điện áp sử dụng: 5VDC từ cổng MicroUSB
- IC chuyển giao tiếp USB-UART: CH340
- Tích hợp hai nút nhấn RST và IO0 sử dụng trong quá trình nạp chương trình.
- Kích thước: 27x40mm



Hình 4.3: Module ESP32-CAM-MB (nguồn: grobotronics.com)

Dây cáp nạp ESP32: Sử dụng để kết nối, nạp chương trình từ máy tính cho mạch ESP32. Chi phí sử dụng cho thiết bị này là: 10.000 VNĐ

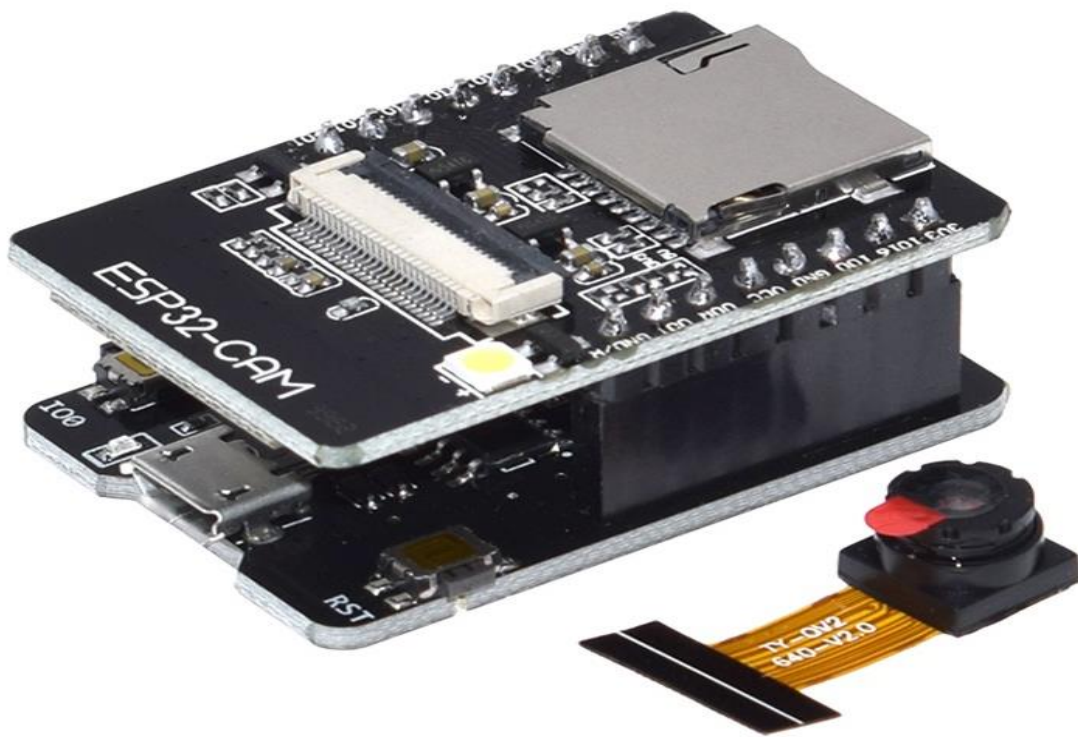
Thông số cơ bản:

- Trọng lượng: 20g
- Chiều dài: 50cm
- Chất liệu: Nhựa bọc bạc chống nhiễu
- Màu sắc: Xanh
- Điện áp: 5V/2A



Hình 4.4: Dây cáp Micro USB (nguồn: shopee.vn)

4.1.3. Sơ đồ lắp đặt

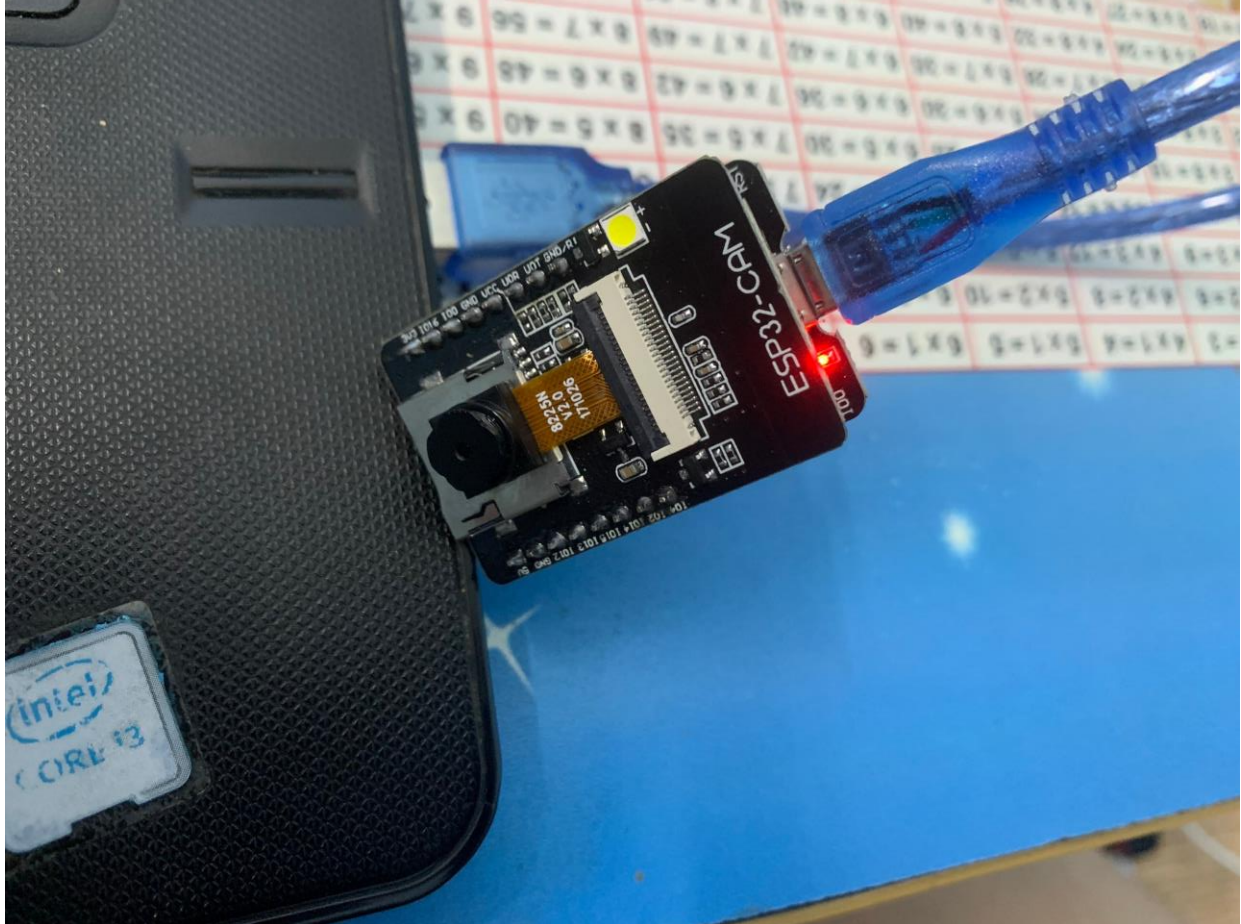


Hình 4.5: Sơ đồ lắp đặt (nguồn: shopee.vn)

Hình 4.5 thể hiện cách lắp đặt ESP32-CAM Ai-Thinker kết nối với để nạp chương trình ESP32-CAM USB Programming Adapter

4.1.4. Minh hoạ thực tế

Dưới đây là hình ảnh minh hoạ thực tế cho việc lắp đặt bo mạch ESP32-Cam kết nối với máy tính.

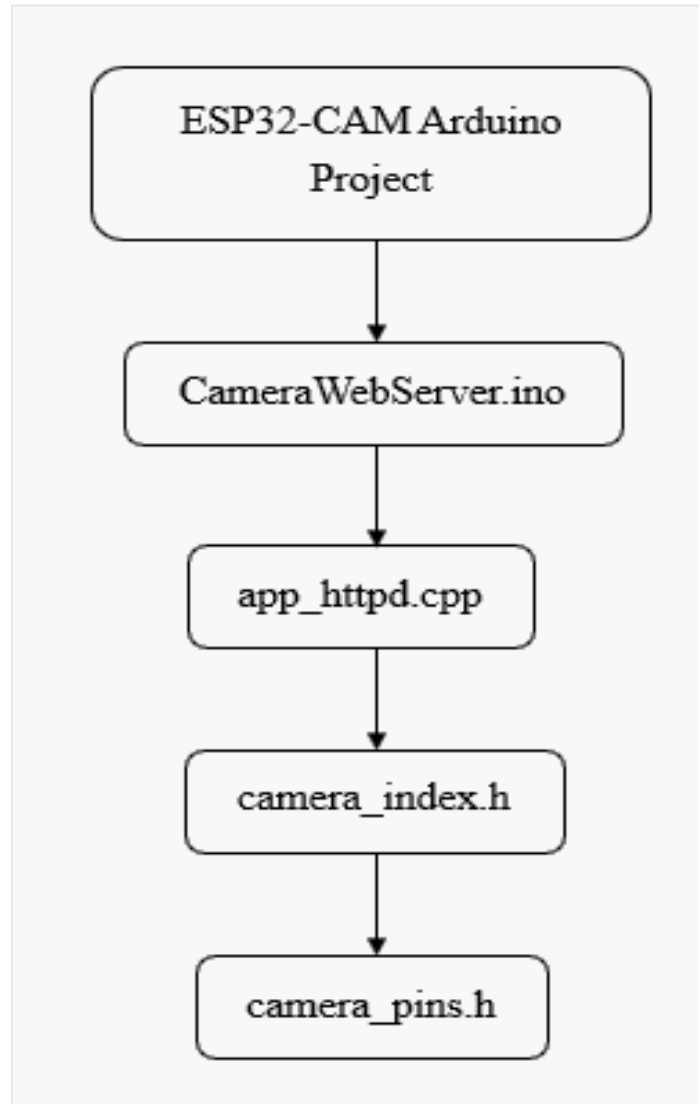


Hình 4.6: Lắp đặt thực tế

4.2. Thiết kế chương trình

4.2.1. Thiết kế chương trình ESP32-Cam

4.2.1.a. Sơ đồ dự án



Hình 4.7: Sơ đồ dự án

4.2.1.b. Chức năng chính của từng thành phần

– *Tập camera_pins.h*: Chứa các định nghĩa về cấu hình chân GPIO (General Purpose Input Output) cho các model camera khác nhau được sử dụng với ESP32. Các định nghĩa này xác định các chân cụ thể của vi điều khiển ESP32 sẽ kết nối với các chân của module camera. Mã nguồn này là một phần quan trọng trong việc cấu hình phần cứng để đảm bảo rằng vi điều khiển có thể giao tiếp đúng cách với camera. Tập này cho phép mã nguồn

chính (CameraWebServer.ino) để dàng chọn và cấu hình các chân GPIO tương ứng cho model camera cụ thể mà bạn đang sử dụng. Điều này làm cho mã linh hoạt và dễ dàng điều chỉnh cho các phần cứng khác nhau mà không cần thay đổi mã nguồn chính. Chỉ cần xác định model camera cụ thể trong tệp “CameraWebServer.ino”. Mã nguồn ở hình 4.8 dưới đây.

```

C camera_pins.h > ≡ RESET_GPIO_NUM
139  #elif defined(CAMERA_MODEL_AI_THINKER)
140  #define PWDN_GPIO_NUM    32
141  #define RESET_GPIO_NUM   -1
142  #define XCLK_GPIO_NUM     0
143  #define SIOD_GPIO_NUM     26
144  #define SIOC_GPIO_NUM     27
145
146  #define Y9_GPIO_NUM       35
147  #define Y8_GPIO_NUM       34
148  #define Y7_GPIO_NUM       39
149  #define Y6_GPIO_NUM       36
150  #define Y5_GPIO_NUM       21
151  #define Y4_GPIO_NUM       19
152  #define Y3_GPIO_NUM       18
153  #define Y2_GPIO_NUM        5
154  #define VSYNC_GPIO_NUM    25
155  #define HREF_GPIO_NUM     23
156  #define PCLK_GPIO_NUM     22
157
158  // 4 for flash led or 33 for normal led
159  #define LED_GPIO_NUM      4
160
161  #elif defined(CAMERA_MODEL_TTGO_T_JOURNAL)
162  #define PWDN_GPIO_NUM     0
163  #define RESET_GPIO_NUM    15
164  #define XCLK_GPIO_NUM     27
165  #define SIOD_GPIO_NUM     25
166  #define SIOC_GPIO_NUM     23

```

Hình 4.8: Mã nguồn camera_pins.h

– *Tệp camera_index.h*: Định nghĩa các trang HTML (hoặc các phần của trang HTML) mà máy ảnh ESP32-CAM sẽ phản hồi cho các yêu cầu từ trình duyệt web. Nó tạo ra giao diện người dùng cho việc kiểm soát máy ảnh và xem video trực tiếp. Mã nguồn ở hình 4.9 dưới đây.

```
C camera_index.h > [0] index_ov2640_html_gz
1  #define index_ov2640_html_gz_len 6787
2  const uint8_t index_ov2640_html_gz[] = {
3      0x1F, 0x8B, 0x08, 0x23, 0xFC, 0x69, 0x5E, 0x00, 0x03, 0x69, 0x6E, 0x64, 0x65, 0x78, 0x5F,
4      0x6F, 0x76, 0x32, 0x36, 0x34, 0x30, 0x2E, 0x68, 0x74, 0x6D, 0x6C, 0x00, 0xED, 0x3D, 0x6B, 0x73,
5      0xDB, 0x46, 0x92, 0xDF, 0xFD, 0x2B, 0x60, 0x24, 0x6B, 0x92, 0x25, 0x92, 0x22, 0x29, 0x4A, 0x96,
6      0x15, 0x89, 0x3E, 0x5B, 0x96, 0x1F, 0xB5, 0x76, 0xE2, 0xB5, 0x12, 0xC7, 0x5B, 0xA9, 0x2D, 0x07,
7      0x04, 0x86, 0x24, 0x62, 0x10, 0xE0, 0x02, 0xA0, 0x48, 0x26, 0xA5, 0xDF, 0x71, 0x3F, 0xE8, 0xFE,
8      0xD8, 0x75, 0xCF, 0x03, 0x18, 0x00, 0x83, 0x07, 0x49, 0x80, 0xF4, 0xFA, 0x8E, 0x4E, 0x45, 0x78,
9      0x4C, 0xF7, 0xF4, 0xBB, 0x7B, 0x66, 0x30, 0xC0, 0xF9, 0x43, 0xCB, 0x33, 0xC3, 0xD5, 0x8C, 0x68,
10     0x93, 0x70, 0xEA, 0x0C, 0x1E, 0x9C, 0xB3, 0x3F, 0x1A, 0xFC, 0xCE, 0x27, 0xC4, 0xB0, 0xD8, 0x21,
11     0x3D, 0x9D, 0x92, 0xD0, 0xD0, 0xCC, 0x89, 0xE1, 0x07, 0x24, 0xBC, 0xD0, 0xE7, 0xE1, 0xA8, 0x75,
12     0xAA, 0xA7, 0x6F, 0xBB, 0xC6, 0x94, 0x5C, 0xE8, 0x37, 0x36, 0x59, 0xCC, 0x3C, 0x3F, 0xD4, 0x35,
13     0xD3, 0x73, 0x43, 0xE2, 0x42, 0xF3, 0x85, 0x6D, 0x85, 0x93, 0x0B, 0x8B, 0xDC, 0xD8, 0x26, 0x69,
14     0xD1, 0x93, 0xA6, 0xED, 0xDA, 0xA1, 0x6D, 0x38, 0xAD, 0xC0, 0x34, 0x1C, 0x72, 0xD1, 0x95, 0x71,
15     0x85, 0x76, 0xE8, 0x90, 0xC1, 0xD5, 0xF5, 0xFB, 0xA3, 0x9E, 0xF6, 0xD3, 0xC7, 0x5E, 0xFF, 0xA4,
16     0x73, 0x7E, 0xC8, 0xAE, 0xC5, 0x6D, 0x82, 0x70, 0x25, 0x9F, 0xE3, 0x6F, 0xE8, 0x59, 0x2B, 0xED,
17     0xAF, 0xC4, 0x25, 0xFC, 0x8D, 0x80, 0x88, 0xD6, 0xC8, 0x98, 0xDA, 0xCE, 0xEA, 0x4C, 0x7B, 0xE6,
18     0x43, 0x9F, 0xCD, 0xD7, 0xC4, 0xB9, 0x21, 0xA1, 0x6D, 0x1A, 0xCD, 0xC0, 0x70, 0x83, 0x56, 0x40,
19     0x7C, 0x7B, 0xF4, 0x43, 0x06, 0x70, 0x68, 0x98, 0x5F, 0xC6, 0xBE, 0x37, 0x77, 0xAD, 0x33, 0xED,
20     0xBB, 0xEE, 0x29, 0xFE, 0xCB, 0x36, 0x32, 0x3D, 0xC7, 0xF3, 0xE1, 0xFE, 0xD5, 0x4B, 0xFC, 0x97,
21     0xBD, 0x4F, 0x7B, 0x0F, 0xEC, 0x3F, 0xC9, 0x99, 0xD6, 0x3D, 0x99, 0x2D, 0x13, 0xF7, 0x6F, 0x1F,
22     0x24, 0x4E, 0x27, 0xBD, 0x3C, 0xEA, 0x39, 0xFC, 0x69, 0x31, 0x7C, 0x40, 0xCC, 0xD0, 0xF6, 0xDC,
23     0xF6, 0xD4, 0xB0, 0x5D, 0x05, 0x26, 0xCB, 0x0E, 0x66, 0x8E, 0x01, 0x32, 0x18, 0x39, 0xA4, 0x10,
24     0xCF, 0x77, 0x53, 0xE2, 0xCE, 0x9B, 0x25, 0xD8, 0x10, 0x49, 0xCB, 0xB2, 0x7D, 0xD6, 0xEA, 0x0C,
25     0xE5, 0x30, 0x9F, 0xBA, 0xA5, 0x68, 0xB8, 0xE8, 0x72, 0x3D, 0x97, 0x28, 0x04, 0x88, 0x1D, 0x2D,
26     0x7C, 0x63, 0x86, 0x0D, 0xF0, 0x6F, 0xB6, 0xC9, 0xD4, 0x76, 0x99, 0x51, 0x9D, 0x69, 0x47, 0xFD,
27     0xCE, 0x6C, 0x59, 0xA2, 0xCA, 0xA3, 0x13, 0xFC, 0x97, 0x6D, 0x34, 0x33, 0x2C, 0xCB, 0x76, 0xC7,
28     0x67, 0xDA, 0xA9, 0x12, 0xB5, 0xE7, 0x5B, 0xC4, 0x6F, 0xF9, 0xB6, 0x65, 0xCF, 0xB3, 0x33, 0xAD,
29     0xAF, 0x6A, 0x33, 0x35, 0xFC, 0x31, 0xD0, 0x12, 0x7A, 0x40, 0x6C, 0xAB, 0xAB, 0xA4, 0x84, 0x37,
```

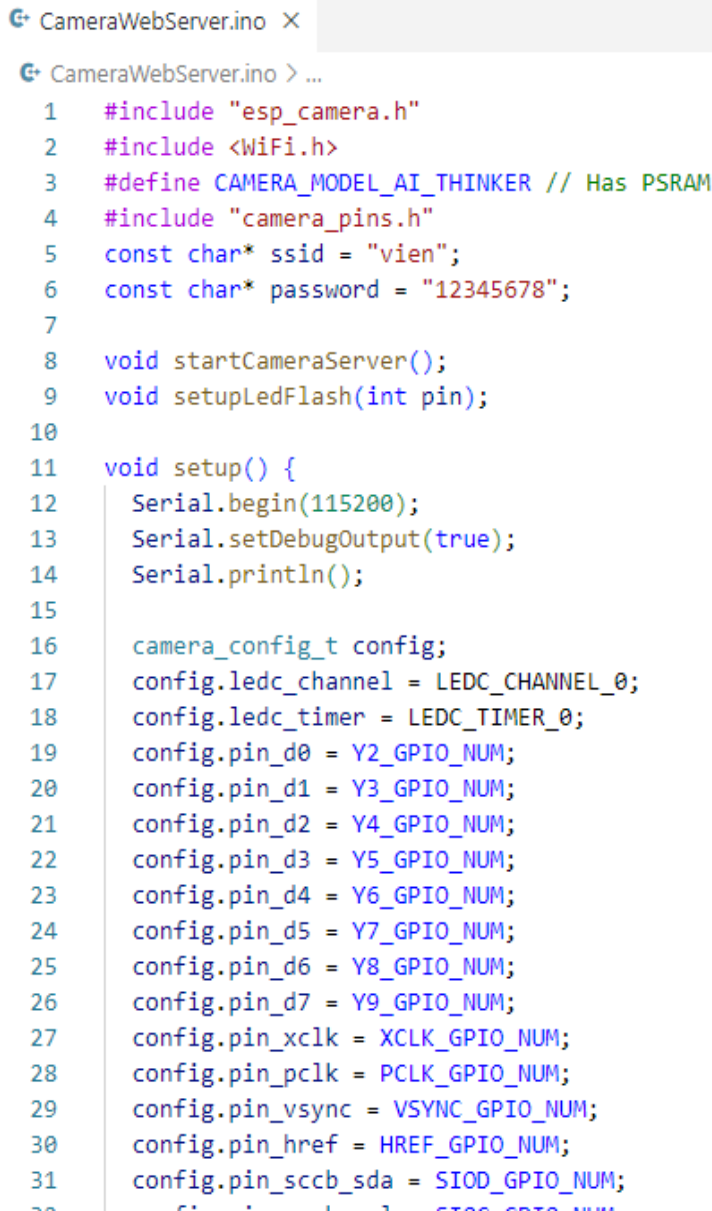
Hình 4.9: Mã nguồn camera_index.h

– *Tệp app_httpd.cpp*: Chứa mã để xử lý các yêu cầu HTTP từ các thiết bị khác thông qua HTTP server trên ESP32-CAM. Mã nguồn ở hình 4.10 dưới đây.

```
app_httpd.cpp > ...
1  #include "esp_http_server.h"
2  #include "esp_timer.h"
3  #include "esp_camera.h"
4  #include "img_converters.h"
5  #include "fb_gfx.h"
6  #include "esp32-hal-ledc.h"
7  #include "sdkconfig.h"
8  #include "camera_index.h"
9
10 #if defined(ARDUINO_ARCH_ESP32) && defined(CONFIG_ARDUHAL_ESP_LOG)
11 #include "esp32-hal-log.h"
12 #endif
13
14 // Face Detection will not work on boards without (or with disabled) PSRAM
15 #ifndef BOARD_HAS_PSRAM
16 #define CONFIG_ESP_FACE_DETECT_ENABLED 1
17 // Face Recognition takes upward from 15 seconds per frame on chips other than ESP32S3
18 // Makes no sense to have it enabled for them
19 #if CONFIG_IDF_TARGET_ESP32S3
20 #define CONFIG_ESP_FACE_RECOGNITION_ENABLED 1
21 #else
22 #define CONFIG_ESP_FACE_RECOGNITION_ENABLED 0
23 #endif
24 #else
25 #define CONFIG_ESP_FACE_DETECT_ENABLED 0
26 #define CONFIG_ESP_FACE_RECOGNITION_ENABLED 0
27 #endif
28
29 #if CONFIG_ESP_FACE_DETECT_ENABLED
30
31 #include <vector>
32 #include "human_face_detect_msr01.hpp"
33 #include "human_face_detect_mnp01.hpp"
```

Hình 4.10: Mã nguồn app_httpd.cpp

– *Tập CameraWebServer.ino*: Mã nguồn chính của dự án, nơi viết các chức năng điều khiển của ESP32-Cam đồng thời bao gồm việc thiết lập kết nối mạng, khởi tạo máy ảnh, cấu hình HTTP server, và xử lý các yêu cầu từ trình duyệt web thông qua giao diện người dùng được cung cấp trong các tệp khác như “camera_index.h”. Mã nguồn ở hình 4.11 dưới đây.



```

1  #include "esp_camera.h"
2  #include <WiFi.h>
3  #define CAMERA_MODEL_AI_THINKER // Has PSRAM
4  #include "camera_pins.h"
5  const char* ssid = "vien";
6  const char* password = "12345678";
7
8  void startCameraServer();
9  void setupLedFlash(int pin);
10
11 void setup() {
12     Serial.begin(115200);
13     Serial.setDebugOutput(true);
14     Serial.println();
15
16     camera_config_t config;
17     config.ledc_channel = LEDC_CHANNEL_0;
18     config.ledc_timer = LEDC_TIMER_0;
19     config.pin_d0 = Y2_GPIO_NUM;
20     config.pin_d1 = Y3_GPIO_NUM;
21     config.pin_d2 = Y4_GPIO_NUM;
22     config.pin_d3 = Y5_GPIO_NUM;
23     config.pin_d4 = Y6_GPIO_NUM;
24     config.pin_d5 = Y7_GPIO_NUM;
25     config.pin_d6 = Y8_GPIO_NUM;
26     config.pin_d7 = Y9_GPIO_NUM;
27     config.pin_xclk = XCLK_GPIO_NUM;
28     config.pin_pclk = PCLK_GPIO_NUM;
29     config.pin_vsync = VSYNC_GPIO_NUM;
30     config.pin_href = HREF_GPIO_NUM;
31     config.pin_sccb_sda = SIOD_GPIO_NUM;
32     config.pin_sccb_scl = SIOC_GPIO_NUM;

```

Hình 4.11: Mã nguồn CameraWebServer.ino

4.2.2. Thiết kế ứng dụng web

4.2.2.a. Python Flask

Flask là framework Python mạnh mẽ chúng ta sử dụng để xây dựng backend của ứng dụng. Flask xử lý các yêu cầu từ phía frontend, xác định logic ứng dụng, và gửi lại các phản hồi tương ứng. Sau đây tôi sẽ trình bày việc xây dựng backend cho dự án bằng flask.

Xây dựng luồng frame để hiển thị video

```
FlaskDemo.py > ...
51 # Tạo luồng frame
52 def generate_frames():
53     global capture_screenshot
54     response = requests.get(url32esp + '/control?var=framesize&val=8')
55     response = requests.get(url32esp + '/control?var=vflip&val=1')
56     frame_size = response.text.strip()
57     if frame_size == "13":
58         width, height = 160, 120
59     elif frame_size == "0": ...
60     else: ...
61
62
63 cap = cv2.VideoCapture(url32esp + ':81/stream')
64 cap.set(cv2.CAP_PROP_FRAME_WIDTH, width)
65 cap.set(cv2.CAP_PROP_FRAME_HEIGHT, height)
66
67
68 while True:
69     success, frame = cap.read()
70     if not success:
71         break
72     else:
73         draw_timeline(frame)
74         if recording:
75             frame_queue.put((datetime.now(), frame))
76         # Chụp màn hình nếu cần
77         if capture_screenshot:
78             take_screenshot(frame)
79             capture_screenshot = False
80         ret, jpeg = cv2.imencode('.jpg', frame)
81         frame = jpeg.tobytes()
82         yield (b'--frame\r\n'
83               b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')
84
```

Hình 4.12: Hàm tạo luồng stream

Xây dựng hàm tạo video khi người dùng muốn ghi hình

```
def create_video_from_frames():
    global start_time
    if os.listdir(temp_dir):
        fourcc = cv2.VideoWriter_fourcc(*'mp4v')
        current_time = datetime.now()
        start_time = current_time
        video_name = current_time.strftime("%Y%m%d_%H%M%S") + ".mp4"
        video_path = os.path.join('VideoRecording', video_name)
        out = cv2.VideoWriter(video_path, fourcc, 18.0, (640, 480))
        frame_files = [f for f in os.listdir(temp_dir) if f.endswith('.jpg')]
        frame_files.sort()
        for filename in frame_files:
            frame = cv2.imread(os.path.join(temp_dir, filename))
            out.write(frame)
        out.release()
        #upload_video_to_firebase(video_path, "RecordingLibs/" + video_name)
        shutil.rmtree(temp_dir)
```

Hình 4.13: Hàm tạo tập video mp4

Xây dựng hàm chụp màn hình stream

```

3 def take_screenshot(frame):
4     if not os.path.exists("screenshots"):
5         os.makedirs("screenshots")
6
7     screenshot_path = os.path.join("screenshots", f"screenshot_{datetime.now().strftime('%Y%m%d_%H%M%S')}.jpg")
8     cv2.imwrite(screenshot_path, frame)
9     print(f"Saved screenshot: {screenshot_path}")

```

Hình 4.14: Hàm chụp ảnh màn hình

Tạo một tệp json lưu cấu hình kết nối với firebase

```

{} credentials.json > ...
1 {
2     "type": "service_account",
3     "project_id": "py-recording",
4     "private_key_id": "35ee9b17e00b6c93bb0be9decec4108b1d453b5b",
5     "private_key": "-----BEGIN PRIVATE KEY-----\nMIIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKggSwAgEAAoIBAQDtcS04JUe813mj
6     "client_email": "firebase-adminsdk-c3xiy@py-recording.iam.gserviceaccount.com",
7     "client_id": "116728243550035748135",
8     "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9     "token_uri": "https://oauth2.googleapis.com/token",
10    "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11    "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-c3xiy%40py-reco
12    "universe_domain": "googleapis.com"
13 }
14

```

Hình 4.15: Cấu hình kết nối với firebase

Sử dụng module `firebase_admin` để thực hiện khởi tạo firebase và xây dựng hàm tải lên (tải xuống) tệp từ firebase storage

```

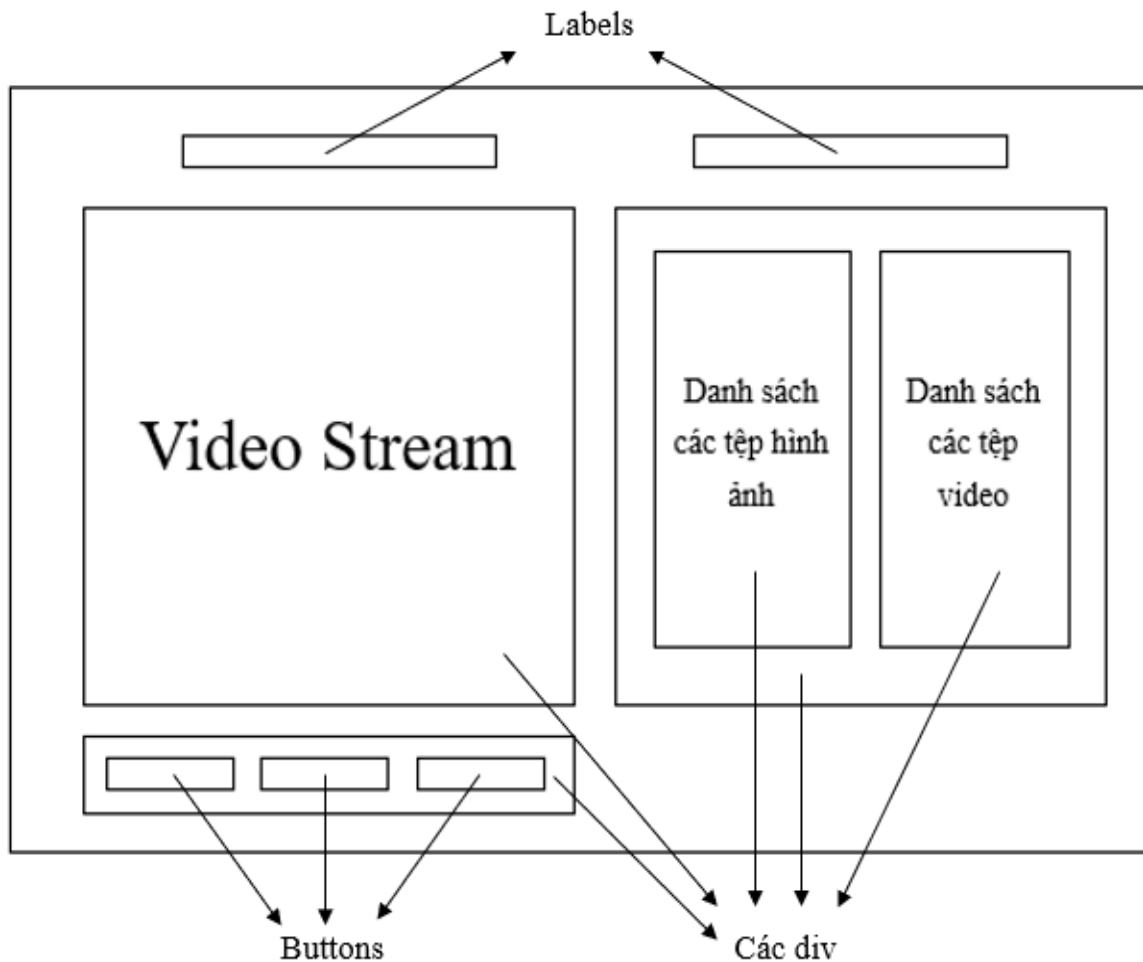
161 import firebase_admin
162 from firebase_admin import credentials
163 from firebase_admin import storage
164
165 # Khởi tạo Firebase Admin SDK
166 cred = credentials.Certificate("credentials.json")
167 firebase_admin.initialize_app(cred, {
168     'storageBucket': 'py-recording.appspot.com'
169 })
170
171 # Hàm upload video lên Firebase Cloud Storage
172 def upload_to_firebase(file_path, destination_path):
173     bucket = storage.bucket()
174     blob = bucket.blob(destination_path)
175     blob.upload_from_filename(file_path)
176
177 # Hàm download video từ Firebase Cloud Storage về local
178 def download_file_from_firebase(file_from_storage, local_path):
179     bucket = storage.bucket()
180     blob = bucket.blob(file_from_storage)
181     blob.download_to_filename(local_path)
182

```

Hình 4.16: Khởi tạo firebase và xây dựng hàm upload, download

4.2.2.b. Giao diện web

Thiết kế giao diện



Hình 4.17: Thiết kế giao diện web

Trang web được chia thành hai phần chính:

- Video stream: Hiển thị luồng stream từ ESP32-Cam.
- Danh sách các tệp (ảnh, video): Hiển thị các ảnh và video trong local và trên firebase storage. Tại đây có hiển thị thêm các nút upload (cho phép người dùng tải tệp lên firebase storage) và nút download (cho phép người dùng tải tệp về máy từ firebase storage) bên dưới tên các file.

Các tag được sử dụng bao gồm: div, label, ul, li...

Mỗi sự kiện cho button đều được cài đặt ở phần script.

Hộp thoại alert sẽ hiển thị sau một vài tác vụ nhất định.

Mã nguồn

```

templates > <> index.html
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>ESP32-CAM Stream</title>
7      <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
8    <style> ...
38  </style>
39  </head>
40  <body>
41    <div id="all-element">
42      <div id="Frame-streaming">
43        <h1>ESP32-CAM Stream</h1>
44        <div class="container">
45          
50        </div>
51      <div class="buttons"> ...
58    </div>
59  </div>
60  <div class="All-files-capture"> ...
74  </div>
75  <div id="All-files-video"> ...
89  </div>
90  </div>
91

```

Hình 4.18: Nội dung của index chính

```

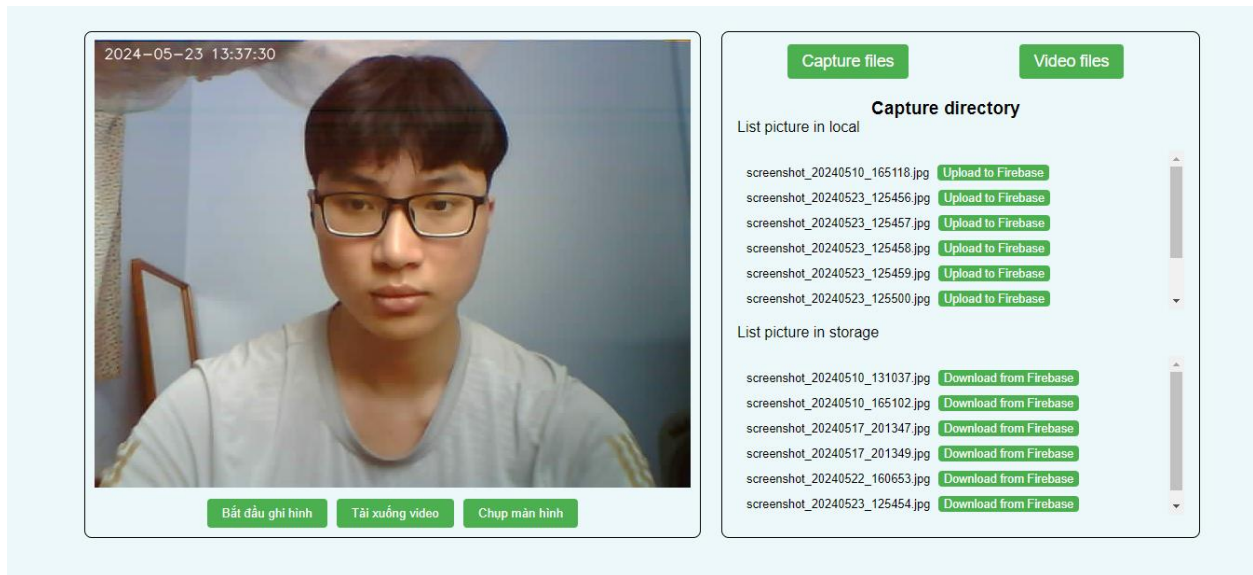
92  <script>
93  > $(document).ready(function () { ...
120  });
121  // Hàm gọi API để upload capture lên Firebase
122  > function uploadCapture(captureName) { ...
138  }
139  // Hàm gọi API để upload video lên Firebase
140  > function uploadToFirebase(videoName) { ...
156  }
157  // Cập nhật danh sách ảnh chụp màn hình
158  > function updateCaptureList() { ...
181  }
182  // Sau khi cập nhật danh sách video, kiểm tra từng video
183  > function updateVideoList() { ...
205  }
206  // Hàm kiểm tra xem capture có tồn tại trong mảng files hay không
207  > function checkCaptureInFirebase(captureName, callback) { ...
213  }
214  // Hàm kiểm tra xem videoName có tồn tại trong mảng files hay không
215  > function checkVideoInFirebase(videoName, callback) { ...
221  }
222  // Danh sách file trong storage
223  > function updateStorageCaptureList(dirName) { ...
241  }
242  > function updateStorageFileList(dirName) { ...
260  }
261  // Hàm gọi API để tải xuống video từ Firebase
262  > function downloadFromFirebase(file_name, dirName) { ...
277  }
278  </script>

```

Hình 4.19: Các hàm xây dựng trong script

4.3. Thực thi chương trình

4.3.1. Giao diện chương trình



Hình 4.20: Giao diện chương trình

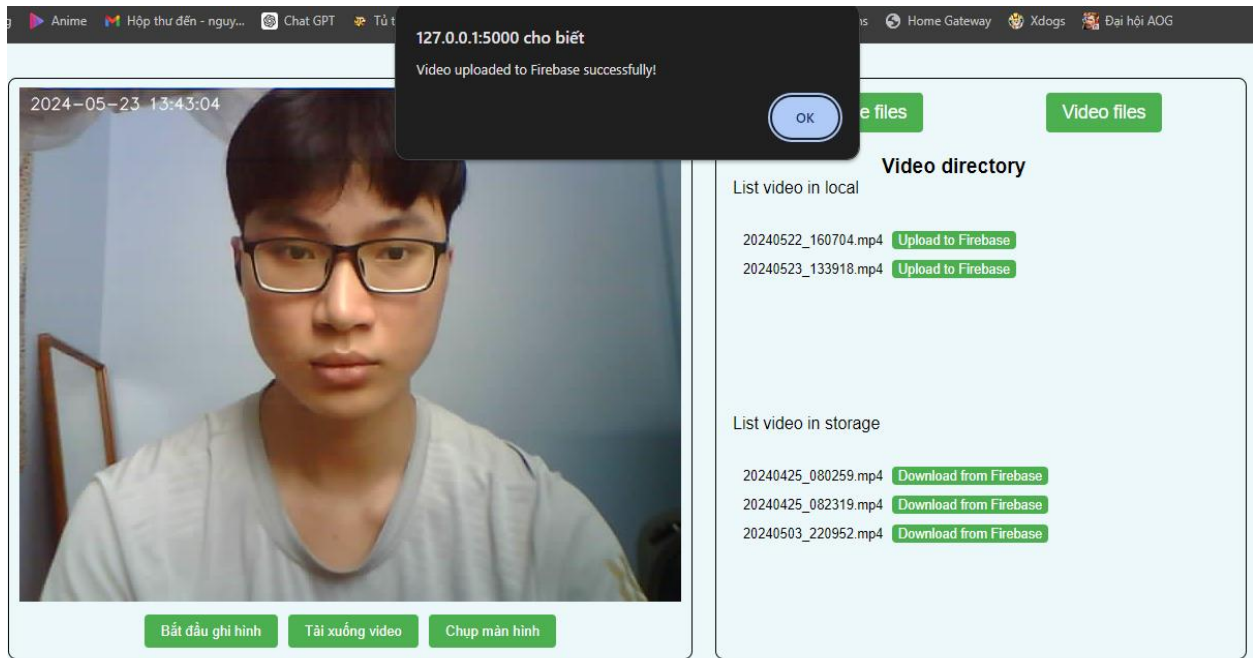
4.3.2. Thực hiện các chức năng

Chức năng ghi hình



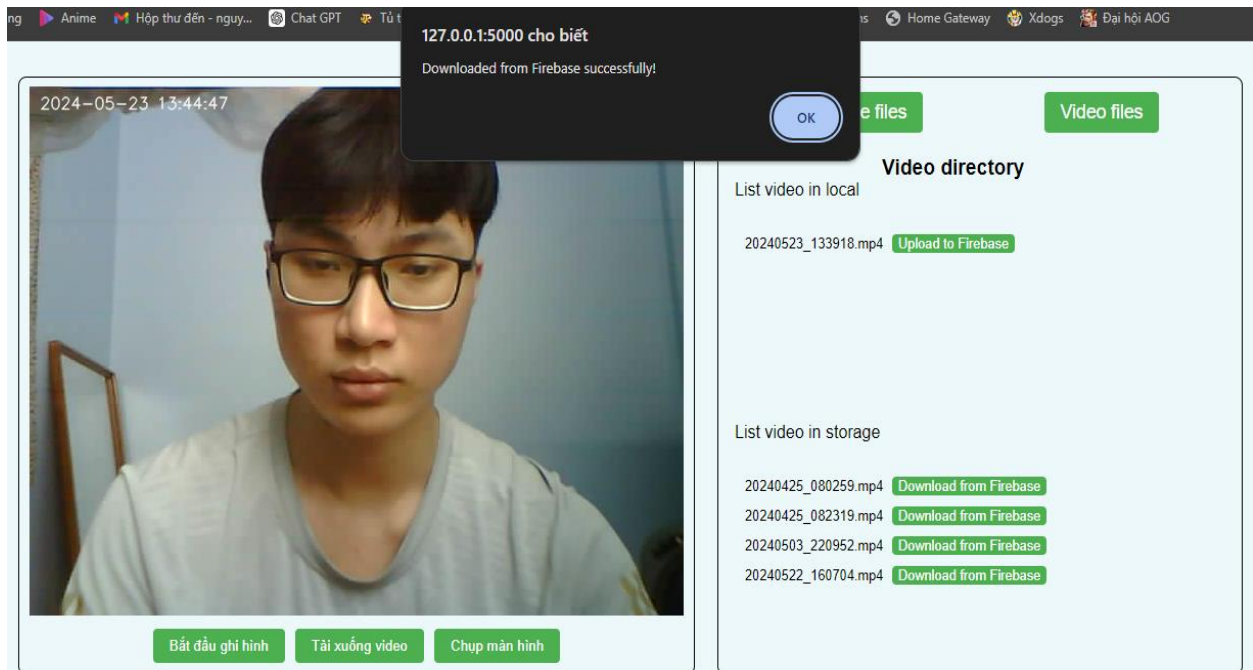
Hình 4.21: Thực hiện ghi hình

Chức năng upload file lên firebase



Hình 4.22: Chức năng upload

Chức năng download file từ firebase



Hình 4.23: Chức năng download

4.4. Đánh giá hệ thống

Đánh giá chung

- Hệ thống đã hoạt động chính xác, hoàn thiện đầy đủ các yêu cầu của dự án.
- Hệ thống dễ sử dụng.
- Giao diện hệ thống đơn giản, dễ nhìn.
- Hệ thống có logic rõ ràng.
- Hệ thống yêu cầu thiết bị không phức tạp, có thể chuẩn bị dễ dàng.
- Tính năng lưu trữ linh hoạt.

Ưu Điểm

- Linh hoạt và tiện lợi: Có thể giám sát từ xa, dễ dàng quản lý.
- Chất lượng hình ảnh cao: Đáp ứng nhu cầu giám sát chi tiết.
- Dễ dàng mở rộng: Thêm mới camera một cách đơn giản.
- Tiết kiệm chi phí: Giảm thiểu chi phí lắp đặt và bảo trì so với các hệ thống giám sát truyền thống.

Nhược Điểm

- Phụ thuộc vào internet: Hiệu suất và độ tin cậy bị ảnh hưởng bởi chất lượng kết nối internet.
- Bảo mật: Rủi ro bảo mật nếu hệ thống không được bảo vệ tốt.
- Chi phí ban đầu: Đầu tư ban đầu cho thiết bị và hệ thống có thể cao.
- Chất lượng video, hình ảnh phụ thuộc vào thiết bị camera.

KẾT LUẬN

Dự án của tôi đã thành công trong việc xây dựng một ứng dụng web đơn giản sử dụng Flask và HTML. Qua quá trình phát triển, tôi đã triển khai các chức năng chính của ứng dụng, bao gồm:

- *Xây dựng giao diện người dùng*: Sử dụng HTML và CSS để tạo ra một giao diện người dùng thân thiện và dễ sử dụng..
- *Phát triển backend bằng Flask*: Sử dụng Flask để xử lý các yêu cầu HTTP và cung cấp các phản hồi tương ứng. Tích hợp các chức năng cơ bản như xử lý dữ liệu, xác thực người dùng.
- *Kết nối và xử lý dữ liệu*: Sử dụng các công cụ và thư viện của Python để kết nối và xử lý dữ liệu hình ảnh, làm việc với tập tin. Đảm bảo rằng dữ liệu được quản lý và xử lý một cách hiệu quả và bảo mật.

Những điểm nổi bật

- *Đơn giản và dễ dàng mở rộng*: Flask, với sự đơn giản và linh hoạt của nó, cho phép tôi dễ dàng mở rộng và tùy chỉnh ứng dụng theo nhu cầu.
- *Tích hợp với Firebase*: Việc tích hợp Firebase giúp tôi dễ dàng quản lý và lưu trữ dữ liệu, đặc biệt là trong việc xử lý tệp và xác thực người dùng.

Hướng phát triển tương lai: Mặc dù dự án hiện tại đã đáp ứng được các mục tiêu ban đầu, vẫn còn nhiều hướng phát triển và cải tiến có thể thực hiện trong tương lai:

- *Mở rộng chức năng*: Thêm các chức năng mới như quản lý người dùng, chức năng tìm kiếm. Phát triển thêm các module để cải thiện trải nghiệm người dùng và tăng tính tương tác.
- *Tối ưu hóa hiệu suất*: Tối ưu hóa mã nguồn và cơ sở dữ liệu để cải thiện tốc độ và hiệu suất của ứng dụng. Áp dụng các kỹ thuật cache để giảm tải cho server và tăng tốc độ phản hồi.
- *Bảo mật*: Tăng cường các biện pháp bảo mật để bảo vệ dữ liệu người dùng và ngăn chặn các tấn công từ bên ngoài. Thực hiện các quy trình kiểm thử bảo mật thường xuyên để đảm bảo an toàn cho ứng dụng.
- *Triển khai trên môi trường thực tế*: Nâng cấp hạ tầng để triển khai ứng dụng trên các nền tảng như AWS, Heroku, hoặc Google Cloud. Đảm bảo rằng ứng dụng có thể hoạt động ổn định và hiệu quả trong môi trường sản xuất.

- *Tích hợp tính năng nhận diện khuôn mặt*: Thêm các chức năng nhận diện, phân tích khuôn mặt và thông báo thống kê (ví dụ như hôm nay xuất hiện bao nhiêu khuôn mặt) để tăng cường khả năng giám sát của hệ thống.
- *Tích hợp tự động hoá*: Các module cần được tích hợp việc tự động hoá như: tự động chụp ảnh khoảnh khắc khi xuất hiện người trong khung hình, tự động lưu trữ khi ghi hình, tự động xoá những tệp video đã cũ để giảm thiểu kích thước bộ nhớ lưu trữ.
- *Tăng thêm số lượng camera giám sát*: Mở rộng số lượng, phạm vi camera giám sát sẽ giúp tăng cường được khả năng giám sát an ninh cho hệ thống.

TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1] Phạm Quang Huy, Nguyễn Tất Bảo Thiện (2018). *Lập Trình IOT Với ARDUINO, ESP8266 Và XBEE*. NXB Thanh Niên.
- [2] Võ Duy Tuấn (2017). *Python cơ bản*. NXB Đại học Quốc Gia Tp. HCM.

Tiếng Anh:

- [1] Bahga, A., & Madisetti, V. (2014). *Internet of Things: A Hands-On Approach*. VPT.
- [2] Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., & Boyle, D. (2014). *From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press.
- [3] McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.)*. O'Reilly Media.
- [4] Banzai, M., & Shiloh, M. (2014). *Getting Started with Arduino (3rd ed.)*. O'Reilly Media.

Danh mục các trang web tham khảo:

- [1] Google. *Tài liệu dành cho nhà phát triển Firebase*. Truy cập ngày 10/05/2024 tại: <https://firebase.google.com/docs?hl=vi>
- [2] Karl Söderby (2024), *Getting Started with Arduino*. Truy cập ngày 15/05/2024 tại: <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/>
- [3] Lucas Saavedra Vaz (2024), *Dự án CameraWebServer github*. Truy cập ngày 01/05/2024 tại: <https://github.com/lucasssvaz/arduino-esp32/tree/master/libraries/ESP32/examples/Camera/CameraWebServer>
- [4] Dylan Field and Evan Wallace (2016), *Figma*. Truy cập ngày 16/05/2024 tại: <https://www.figma.com/>
- [5] Amazon Web Services (2006), *Trang web chính thức của Amazon Web Services*. Truy cập ngày 16/05/2024 tại: <https://aws.amazon.com/vi/>
- [6] Refsnes Data (1998), *W3School*. Truy cập ngày 17/05/2024 tại: <https://www.w3schools.com/>