**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN VIỄN THÔNG**

****

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG QUẢN LÝ SÁCH**

GVHD : **GS.TS LÊ TIẾN THƯỜNG**

SVTH : **NGUYỄN CÔNG PHƯƠNG**

MSSV : **2010536**

TP. HỒ CHÍ MINH, NGÀY 28 THÁNG 10 NĂM 2024

ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh Phúc

Số: \_\_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Viễn Thông**

**NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP**

1. Họ và tên: **Nguyễn Công Phương** MSSV: **2010536**
2. Ngành: **Điện – Điện tử** Chuyên ngành: **Điện tử - Viễn thông**
3. Đề tài: **HỆ THỐNG IOT ĐIỀU KHIỂN VÀ GIÁM SÁT NHÀ THÔNG MINH**
4. Nhiệm vụ:

* **Thiết kế mạch … thiết kế hệ thống …**
* **Thi công hệ thống …**
* **Triển khai, phân tích và so sánh kết quả …**

1. Ngày giao nhiệm vụ luận văn: **04/09/2024**
2. Ngày hoàn thành nhiệm vụ: **../../2024**
3. Họ và tên người hướng dẫn: **GS.TS LÊ TIẾN THƯỜNG**

*TP.HCM, ngày xx tháng xx năm 2024*

|  |  |
| --- | --- |
| **CHỦ NHIỆM BỘ MÔN** | **NGƯỜI HƯỚNG DẪN CHÍNH** |
|  |  |
| Học hàm. Học vị. Họ tên | Học hàm. Học vị. Họ tên |

**PHẦN DÀNH CHO KHOA, BỘ MÔN:**

Người duyệt (chấm sơ bộ):

Đơn vị:

Ngày bảo vệ :

Điểm tổng kết:

Nơi lưu trữ luận văn:

# LỜI CÁM ƠN

Lời đầu tiên, em xin chân thành gửi lời cảm ơn sâu sắc đến các thầy cô tại trường Đại học Bách Khoa TP.HCM đã tạo điều kiện để em được học tập, rèn luyện, nắm vững kiến thức về ngành điện và hơn hết là chuyên ngành Điện tử.

Em xin gửi lời cảm ơn chân thành đến thầy **GS. Lê Tiến Thường** giảng viên Bộ môn Viễn Thôn - Trường Đại học Bách Khoa, người đã tận tình hướng dẫn, chỉ bảo em trong quá trình làm đồ án.

Do trình độ chuyên môn còn hạn chế nên khó tránh khỏi sai sót không mong muốn. Em rất mong nhận được sự góp ý chân thành đầy quý giá của thầy cô để đồ án em hoàn thiện hơn, giúp em học hỏi được kinh nghiệm, bổ sung được những thiếu sót để vận dụng vào công việc sau này.

Em xin chân thành cảm ơn!

|  |
| --- |
| TP. HCM, ngày xx, tháng xx năm 2024 |
|  |
| Nguyễn Công Phương |

# LỜI CAM ĐOAN

Tôi tên: Nguyễn Công Phương, là sinh viên cao học chuyên ngành Điện tử - Viễn thông, khóa 2020, tại Đại học Quốc gia thành phố Hồ Chí Minh – Trường Đại học Bách Khoa. Tôi xin cam đoan những nội dung sau đều là sự thật: (i) Công trình nghiên cứu này hoàn toàn do chính tôi thực hiện; (ii) Các tài liệu và trích dẫn trong luận văn này được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao; (iii) Các số liệu và kết quả của công trình này được tôi tự thực hiện một cách độc lập và trung thực.

|  |
| --- |
| TP. HCM, ngày xx, tháng xx năm 2024 |
|  |
| Nguyễn Công Phương |

# TÓM TẮT LUẬN VĂN

Trong bối cảnh hiện đại hóa nền giáo dục và sự phát triển mạnh mẽ của công nghệ, việc ứng dụng Internet of Things (IoT) vào quản lý thư viện trở thành xu hướng tất yếu, nhằm nâng cao hiệu quả quản lý, tối ưu hóa quy trình và mang lại trải nghiệm tốt hơn cho người dùng. Luận văn này tập trung vào việc xây dựng một hệ thống quản lý thư viện thông minh, kết hợp với các công nghệ IoT, giúp quản lý tài liệu, sách vở và tài sản thư viện một cách tự động, hiệu quả.

Trong quá trình nghiên cứu, luận văn này đã khảo sát các công nghệ IoT phổ biến và tiềm năng trong việc áp dụng vào hệ thống quản lý thư viện. Từ đó, đề xuất một kiến trúc hệ thống chi tiết, bao gồm các thành phần như: thiết bị IoT để theo dõi sách và tài sản, hệ thống phần mềm quản lý, cơ sở dữ liệu và các ứng dụng trình duyệt để người dùng tương tác. Hệ thống sẽ giúp tự động hóa các nhiệm vụ như theo dõi vị trí sách, phát hiện sách bị thất lạc, cung cấp dịch vụ mượn trả không cần thủ công, và giám sát, thống kê hệ thống tài nguyên và hoạt động mượn trả sách để đưa ra các đánh giá về hoạt động của thư viện.

Ngoài ra, hệ thống còn tích hợp khả năng phân tích dữ liệu, cho phép thống kê, phân tích thói quen mượn sách của người dùng, từ đó giúp thư viện tối ưu hóa việc quản lý tài liệu và định hướng mua sắm sách. Kết quả nghiên cứu cho thấy hệ thống này có tiềm năng lớn trong việc cải thiện hiệu quả quản lý thư viện, tiết kiệm nguồn lực và thời gian, đồng thời nâng cao chất lượng phục vụ cho người dùng, đặc biệt trong các trường học và cơ sở giáo dục.

Luận văn cung cấp cơ sở lý thuyết, mô hình triển khai chi tiết, và đề xuất giải pháp nhằm hiện đại hóa thư viện thông qua việc ứng dụng IoT. Hệ thống này không chỉ là một bước tiến công nghệ, mà còn góp phần cải thiện môi trường học tập, nâng cao trải nghiệm của học sinh và giáo viên trong việc tiếp cận tri thức.

**MỤC LỤC**

[LỜI CÁM ƠN i](#_Toc181000928)

[LỜI CAM ĐOAN ii](#_Toc181000929)

[TÓM TẮT LUẬN VĂN iii](#_Toc181000930)

[DANH SÁCH BẢNG vii](#_Toc181000931)

[DANH SÁCH HÌNH VẼ viii](#_Toc181000932)

[DANH SÁCH TỪ VIẾT TẮT ix](#_Toc181000933)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc181000934)

[1.1 Đặt vấn đề 1](#_Toc181000935)

[1.2 Phạm vi và phương pháp nghiên cứu 2](#_Toc181000936)

[1.2.1 Phạm vi nghiên cứu 2](#_Toc181000937)

[1.2.2 Phương pháp nghiên cứu 3](#_Toc181000938)

[1.3 Các đóng góp của luận văn 4](#_Toc181000939)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7](#_Toc181000940)

[2.1 Giới thiệu về IOT (Internet Of Thing): 7](#_Toc181000941)

[2.1.1 Khái niệm và định nghĩa về IoT 7](#_Toc181000942)

[2.1.2 Cấu trúc và thành phần của hệ thống IOT 7](#_Toc181000943)

[2.1.3 Vai trò của IOT trong quản lý thư viện 8](#_Toc181000944)

[2.2 Raspberry Pi trong ứng dụng IoT: 9](#_Toc181000945)

[2.2.1 Tổng quan về Raspberry Pi 9](#_Toc181000946)

[2.2.2 Cấu hình và vai trò của Raspberry Pi trong hệ thống 9](#_Toc181000947)

[2.2.3 Ưu điểm khi sử dụng Raspberry Pi làm máy chủ 11](#_Toc181000948)

[2.3 ESP32 và các ứng dụng trong IoT: 12](#_Toc181000949)

[2.3.1 Giới thiệu về ESP32 12](#_Toc181000950)

[2.3.2 Khả năng giao tiếp không dây và vai trò trong hệ thống quét sách 14](#_Toc181000951)

[2.3.3 Tích hợp ESP32 với hệ thống quản lý 15](#_Toc181000952)

[2.4 Express.js trong phát triển web: 15](#_Toc181000953)

[2.4.1 Giới thiệu về Express.js 15](#_Toc181000954)

[2.4.2 Lý do lựa chọn Express.js trong phát triển trang web quản lý thư viện 17](#_Toc181000955)

[2.4.3 Các tính năng quan trọng của Express.js 18](#_Toc181000956)

[2.5 MySQL và quản lý cơ sở dữ liệu trong hệ thống: 19](#_Toc181000957)

[2.5.1 Giới thiệu về MySQL 19](#_Toc181000958)

[2.5.2 Mô hình cơ sở dữ liệu trong quản lý thư viện 20](#_Toc181000959)

[2.5.3 Kết nối MySQL với Express.js 22](#_Toc181000960)

[2.6 Tìm hiểu về Web Speech API 23](#_Toc181000961)

[2.6.1 Tổng quan về Web Speech API 23](#_Toc181000962)

[2.6.2 Lợi ích và ứng dụng của Web Speech API 24](#_Toc181000963)

[2.6.3 Cấu trúc và hoạt động của Web Speech API 24](#_Toc181000964)

[2.6.4 Tính tương thích và các trình duyệt hỗ trợ 26](#_Toc181000965)

[2.7 Kết luận chương 26](#_Toc181000966)

[CHƯƠNG 3. THIẾT KẾ HỆ THỐNG THƯ VIỆN 28](#_Toc181000967)

[3.1 Tổng quan về hệ thống thư viện 28](#_Toc181000968)

[3.1.1 Mục tiêu và chức năng của hệ thống 28](#_Toc181000969)

[3.1.2 Phạm vi ứng dụng của hệ thống thư viện kết hợp IoT 29](#_Toc181000970)

[3.2 Kiến trúc hệ thống 30](#_Toc181000971)

[3.2.1 Mô hình tổng thể hệ thống 30](#_Toc181000972)

[3.2.2 Các thành phần phần cứng 31](#_Toc181000973)

[3.2.3 Các thành phần phần mềm 31](#_Toc181000974)

[3.2.4 Mối quan hệ giữa các thành phần trong hệ thống 32](#_Toc181000975)

[3.3 Thiết kế giao diện trang web 33](#_Toc181000976)

[3.3.1 Nguyên tắc thiết kế 34](#_Toc181000977)

[3.3.2 Các thành phần chính của giao diện 34](#_Toc181000978)

[3.3.3 Chức năng tìm kiếm sách bằng giọng nói 35](#_Toc181000979)

[3.3.4 Công cụ và ngôn ngữ sử dụng 35](#_Toc181000980)

[3.3.5 Giao diện mẫu (Wireframe) 36](#_Toc181000981)

[CHƯƠNG 4. KẾT QUẢ VÀ PHÂN TÍCH 37](#_Toc181000982)

[4.1 Phương pháp tiếp cận 37](#_Toc181000983)

[4.2 Kết quả và phân tích 37](#_Toc181000984)

[4.2.1 Khảo sát thông số A (ghi chú: các mục này nên ghi theo câu hỏi nghiên cứu) 37](#_Toc181000985)

[4.2.2 Kết quả mô phỏng thông số B 37](#_Toc181000986)

[4.3 Kết luận chương 37](#_Toc181000987)

[CHƯƠNG 5. KẾT LUẬN 39](#_Toc181000988)

[5.1 Tóm tắt và kết luận chung 39](#_Toc181000989)

[5.2 Hướng phát triển 39](#_Toc181000990)

[PHỤ LỤC A 40](#_Toc181000991)

[A.2 Code chương trình xử lý dữ liệu dùng Matlab 40](#_Toc181000992)

[TÀI LIỆU THAM KHẢO 41](#_Toc181000993)

# DANH SÁCH BẢNG

# DANH SÁCH HÌNH VẼ

[Hình 1 Hệ thống thư viện kết hợp IOT 1](#_Toc181001189)

[Hình 2 Internet of thing 7](#_Toc181001190)

[Hình 3 Raspberry Pi Zero 2 W 9](#_Toc181001191)

[Hình 4 So sánh Raspberry Pi Zero 2 W và Raspberry Pi 3 Model B 11](#_Toc181001192)

[Hình 5 Vi điều khiển ESP32 13](#_Toc181001193)

[Hình 6 Express frame work 16](#_Toc181001194)

[Hình 7 My SQL database development frame work 20](#_Toc181001195)

[Hình 8 Web Speech API flow chart 23](#_Toc181001196)

[Hình 9 Sơ đồ mô hình tổng thể hệ thống thư viện 30](#_Toc181001197)

[Hình 10. Ý tưởng web site quản lý sách 33](#_Toc181001198)

# DANH SÁCH TỪ VIẾT TẮT

|  |  |
| --- | --- |
| SNR | Signal to noise ratio |
| Ghi chú | Ghi các từ viết tắt sử dụng trong luận văn vào bảng này theo thứ tự alphabet |

# GIỚI THIỆU

## Đặt vấn đề



Hình 1 Hệ thống thư viện kết hợp IOT

Trong thời đại công nghệ số, sự phát triển của Internet of Things (IoT) đang tạo ra những bước tiến quan trọng trong nhiều lĩnh vực, từ sản xuất công nghiệp, y tế cho đến giáo dục. Việc ứng dụng IoT không chỉ mang lại những thay đổi về mặt công nghệ mà còn cải thiện hiệu quả quản lý, tiết kiệm nguồn lực và nâng cao trải nghiệm người dùng. Trong bối cảnh đó, thư viện – một bộ phận không thể thiếu của các trường học và cơ sở giáo dục – đang đối diện với những thách thức lớn trong việc quản lý tài liệu, sách vở cũng như cung cấp dịch vụ hiệu quả cho học sinh, giáo viên.

Ở các thư viện truyền thống, việc quản lý sách và tài liệu thường dựa trên các quy trình thủ công như mượn và trả sách thông qua nhân viên, kiểm kê sách theo chu kỳ, và quản lý tài sản thư viện với sự can thiệp của con người. Những quy trình này không chỉ tốn nhiều thời gian và nguồn lực, mà còn dễ xảy ra sai sót trong việc theo dõi, mất mát tài liệu, và thiếu tính linh hoạt trong việc phục vụ người dùng. Thêm vào đó, sự gia tăng về số lượng tài liệu và người dùng tại các trường học ngày nay đòi hỏi một hệ thống quản lý linh hoạt và hiệu quả hơn.

Việc áp dụng IoT trong hệ thống quản lý thư viện trường học không chỉ là một xu hướng hiện đại, mà còn là một giải pháp cần thiết để giải quyết những vấn đề nêu trên. Các thiết bị IoT có thể giúp tự động hóa quy trình mượn trả sách, theo dõi và quản lý tài sản một cách thông minh, tối ưu hóa việc giám sát điều kiện môi trường để bảo vệ sách và tài liệu. Đồng thời, IoT còn cung cấp khả năng phân tích dữ liệu, giúp thư viện dễ dàng quản lý và dự đoán nhu cầu sử dụng của người dùng, từ đó cải thiện trải nghiệm học tập và nghiên cứu của học sinh, giáo viên.

Tuy nhiên, việc triển khai một hệ thống quản lý thư viện kết hợp IoT không phải là một nhiệm vụ đơn giản. Nó đòi hỏi phải có một kiến thức chuyên sâu về cả công nghệ IoT lẫn các quy trình quản lý thư viện. Ngoài ra, tính khả thi của việc ứng dụng công nghệ mới vào môi trường giáo dục cần được xem xét kỹ lưỡng để đảm bảo hệ thống có thể đáp ứng tốt yêu cầu thực tiễn mà vẫn phù hợp với nguồn lực của nhà trường.

Xuất phát từ thực tiễn và nhu cầu cấp bách đó, đề tài “Xây dựng hệ thống quản lý thư viện kết hợp IoT cho trường học” được thực hiện với mục tiêu nghiên cứu và triển khai một giải pháp hiện đại, giúp nâng cao hiệu quả quản lý thư viện trong môi trường giáo dục. Đề tài sẽ tập trung vào việc tích hợp các thiết bị IoT để theo dõi, quản lý tài sản thư viện, cung cấp các dịch vụ mượn trả sách tự động, và phân tích dữ liệu để tối ưu hóa hoạt động quản lý. Đây là một hướng đi mới nhằm thúc đẩy sự phát triển của thư viện trường học, đồng thời mang lại những lợi ích thiết thực cho cả nhà trường lẫn người học.

## Phạm vi và phương pháp nghiên cứu

### Phạm vi nghiên cứu

Đề tài này tập trung nghiên cứu và triển khai hệ thống quản lý thư viện trường học kết hợp công nghệ IoT. Hệ thống được xây dựng với các chức năng tự động hóa quy trình mượn trả sách, theo dõi tình trạng tài liệu và quản lý tài sản thư viện. Các thành phần chính của hệ thống bao gồm:

* **Trang web quản lý thư viện**: Được xây dựng bằng framework Express.js, trang web này cho phép người dùng (nhân viên thư viện, học sinh) mượn trả sách, kiểm tra thông tin sách, và quản lý tài liệu từ xa. Hệ thống web sẽ được triển khai và chạy trên Raspberry Pi, đóng vai trò như một máy chủ xử lý dữ liệu và kết nối với các thiết bị IoT.
* **Thiết bị IoT để quét sách**: Một module ESP32 sẽ được tích hợp với cảm biến (chẳng hạn như máy quét mã vạch hoặc RFID) để quét sách khi người dùng mượn hoặc trả. Thiết bị ESP32 sẽ giao tiếp với hệ thống web thông qua mạng không dây, gửi dữ liệu về sách đã quét để cập nhật thông tin mượn trả trong cơ sở dữ liệu.
* **Tích hợp IoT vào hệ thống quản lý**: Việc theo dõi và quản lý tài sản thư viện như sách, tài liệu sẽ được thực hiện thông qua cảm biến RFID hoặc mã vạch, với dữ liệu được thu thập và quản lý tập trung trên trang web. Hệ thống này giúp tối ưu hóa quy trình mượn trả sách, ngăn chặn thất thoát tài liệu, và cung cấp các báo cáo chi tiết về việc sử dụng sách trong thư viện.

Phạm vi nghiên cứu bao gồm việc thiết kế, lập trình và triển khai hệ thống, từ việc xây dựng giao diện web, cơ sở dữ liệu cho đến kết nối với các thiết bị IoT thông qua Raspberry Pi và ESP32. Đề tài này tập trung vào ứng dụng cho các thư viện trong trường học, với mục tiêu cung cấp một giải pháp quản lý hiện đại, dễ sử dụng và có khả năng mở rộng trong tương lai.

### Phương pháp nghiên cứu

1. **Nghiên cứu lý thuyết và tài liệu liên quan**: Trước khi bắt tay vào xây dựng hệ thống, nghiên cứu các công nghệ liên quan như IoT, Raspberry Pi, ESP32, Express.js và cơ sở dữ liệu MySQL. Đồng thời, tìm hiểu về các hệ thống quản lý thư viện hiện có và cách chúng vận hành để xác định các yêu cầu thực tiễn.
2. **Thiết kế hệ thống**:
   * **Kiến trúc hệ thống**: Phân chia hệ thống thành các thành phần chính, bao gồm module quét sách (ESP32), máy chủ web quản lý (Raspberry Pi) và cơ sở dữ liệu. Thiết kế kiến trúc tổng quan của hệ thống, xác định cách các thành phần giao tiếp và trao đổi dữ liệu với nhau.
   * **Thiết kế giao diện người dùng**: Giao diện của trang web quản lý được thiết kế thân thiện, dễ sử dụng với người dùng cuối như nhân viên thư viện và học sinh, sinh viên.
3. **Phát triển hệ thống**:
   * **Phát triển module quét sách**: Lập trình cho ESP32 để thực hiện quét mã vạch hoặc RFID, gửi dữ liệu sách về máy chủ Raspberry Pi.
   * **Phát triển máy chủ web với Express.js**: Xây dựng trang web quản lý thư viện, kết nối với cơ sở dữ liệu và xử lý các yêu cầu của người dùng như mượn, trả sách và quản lý tài sản thư viện.
   * **Tích hợp IoT và truyền dữ liệu**: Kết nối ESP32 với hệ thống qua mạng không dây, đảm bảo việc giao tiếp và truyền dữ liệu giữa các thành phần diễn ra một cách ổn định và hiệu quả.
4. **Kiểm thử và đánh giá**: Sau khi phát triển, tiến hành kiểm thử hệ thống trên môi trường thực tế tại thư viện của trường học. Kiểm tra các chức năng như quét sách, quản lý mượn trả và giao tiếp giữa các thiết bị. Đồng thời, thu thập phản hồi từ người dùng để đánh giá hiệu quả và đề xuất cải tiến nếu cần.
5. **Tổng kết và đánh giá**: Đánh giá tổng quan về hệ thống, so sánh với các hệ thống truyền thống, từ đó rút ra những ưu điểm và hạn chế trong quá trình triển khai ứng dụng IoT vào quản lý thư viện.

## Các đóng góp của luận văn

Luận văn “Xây dựng hệ thống quản lý thư viện kết hợp IoT cho trường học” mang lại nhiều đóng góp quan trọng, cả về lý thuyết lẫn thực tiễn. Cụ thể, những đóng góp của luận văn có thể được chia thành các khía cạnh sau:

1. **Ứng dụng thực tiễn của IoT trong quản lý thư viện**:
   * Luận văn đã giới thiệu và phát triển một giải pháp quản lý thư viện thông minh dựa trên công nghệ IoT, giúp tự động hóa quy trình mượn, trả sách, theo dõi và quản lý tài sản. Việc ứng dụng các thiết bị IoT như ESP32 để quét sách và Raspberry Pi làm máy chủ quản lý đã giúp tối ưu hóa quy trình vận hành thư viện, từ đó nâng cao năng suất và giảm thiểu công việc thủ công cho nhân viên.
   * Hệ thống này có thể áp dụng rộng rãi cho các trường học với quy mô nhỏ và trung bình, đồng thời có khả năng mở rộng để phù hợp với các thư viện có quy mô lớn hơn.
2. **Giải pháp tối ưu hóa hoạt động quản lý thư viện**:
   * Thông qua việc triển khai hệ thống quản lý thư viện thông minh, luận văn đã góp phần giải quyết những vấn đề cố hữu trong các thư viện truyền thống, chẳng hạn như thất thoát sách, việc kiểm kê tài sản tốn thời gian, và quy trình mượn trả phức tạp.
   * Hệ thống giúp cung cấp thông tin chi tiết về tình trạng sách, tạo ra các báo cáo thống kê nhằm phân tích thói quen mượn sách của học sinh, từ đó hỗ trợ ban quản lý thư viện trong việc đưa ra các quyết định liên quan đến việc mua sắm, lưu trữ tài liệu.
3. **Phát triển một hệ thống quản lý thư viện hiện đại và dễ triển khai**:
   * Hệ thống được phát triển dựa trên các công nghệ hiện đại như Express.js, Raspberry Pi và ESP32, với cách triển khai đơn giản và tiết kiệm chi phí. Việc tích hợp giữa phần mềm quản lý và phần cứng IoT giúp tạo ra một giải pháp toàn diện, dễ dàng triển khai cho các thư viện với nguồn lực hạn chế.
   * Giao diện quản lý trực quan và thân thiện giúp người dùng dễ dàng làm quen và sử dụng hệ thống mà không cần nhiều đào tạo chuyên sâu.
4. **Đóng góp về mặt nghiên cứu và công nghệ**:
   * Luận văn đã cung cấp một hướng đi mới trong việc ứng dụng IoT vào các lĩnh vực giáo dục, cụ thể là quản lý thư viện, mở ra tiềm năng phát triển các hệ thống quản lý tương tự cho các cơ sở giáo dục khác.
   * Đề tài này còn là một tài liệu tham khảo hữu ích cho các nghiên cứu về việc ứng dụng công nghệ IoT vào các hệ thống quản lý khác, không chỉ giới hạn trong thư viện mà còn trong các lĩnh vực khác như quản lý tài sản, kho hàng, và giáo dục.
5. **Khả năng mở rộng và nâng cấp hệ thống**:
   * Hệ thống được thiết kế với tính linh hoạt và khả năng mở rộng cao. Trong tương lai, hệ thống có thể được bổ sung thêm các tính năng mới như tích hợp máy quét nhận diện khuôn mặt, cảnh báo tự động khi sách được mượn quá hạn, hay giám sát điều kiện môi trường để bảo quản tài liệu. Điều này giúp hệ thống không chỉ dừng lại ở quy mô nhỏ mà còn có thể được nâng cấp để đáp ứng nhu cầu của các thư viện lớn hơn.

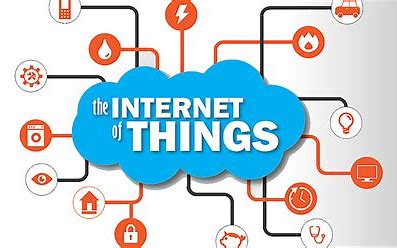
Tóm lại, luận văn không chỉ đóng góp vào việc nghiên cứu ứng dụng IoT trong quản lý thư viện mà còn mang lại giải pháp thiết thực, có khả năng cải thiện hoạt động quản lý tại các thư viện trường học.

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về IOT (Internet Of Thing):

### Khái niệm và định nghĩa về IoT

Internet of Things (IoT) là một mạng lưới các thiết bị vật lý (things) được kết nối với nhau qua internet, cho phép chúng thu thập, chia sẻ và trao đổi dữ liệu mà không cần sự can thiệp của con người. Các thiết bị này có thể bao gồm từ các cảm biến đơn giản, máy móc công nghiệp, đến các thiết bị gia đình thông minh như đèn, điều hòa, và thậm chí cả xe cộ. IoT mở ra khả năng tích hợp thế giới vật lý với hệ thống kỹ thuật số, giúp tối ưu hóa việc quản lý, giám sát và điều khiển từ xa các thiết bị này.



Hình 2 Internet of thing

### Cấu trúc và thành phần của hệ thống IOT

Một hệ thống IoT thường bao gồm ba thành phần chính:

**Thiết bị (Things):** Đây là các đối tượng vật lý được trang bị các cảm biến, bộ truyền tín hiệu, và kết nối mạng để thu thập và truyền tải dữ liệu. Ví dụ: các cảm biến nhiệt độ, máy quét mã vạch, đèn chiếu sáng, thiết bị đeo thông minh.

**Kết nối mạng:** Hệ thống IoT yêu cầu một mạng lưới để kết nối các thiết bị với nhau và với các máy chủ, thường là qua Wi-Fi, Bluetooth, hoặc các mạng di động. Dữ liệu từ thiết bị sẽ được gửi đến các máy chủ hoặc dịch vụ đám mây để xử lý và lưu trữ.

**Máy chủ và nền tảng xử lý dữ liệu:** Đây là nơi dữ liệu từ các thiết bị được thu thập, lưu trữ và phân tích. Các máy chủ có thể là các hệ thống cục bộ hoặc trên đám mây, đóng vai trò điều phối các hoạt động của thiết bị và đưa ra các quyết định điều khiển.

### Vai trò của IOT trong quản lý thư viện

Trong lĩnh vực quản lý thư viện, IoT mang lại nhiều lợi ích quan trọng, đặc biệt là trong việc tự động hóa các quy trình và nâng cao hiệu quả quản lý tài sản. Một số vai trò chính của IoT trong quản lý thư viện bao gồm:

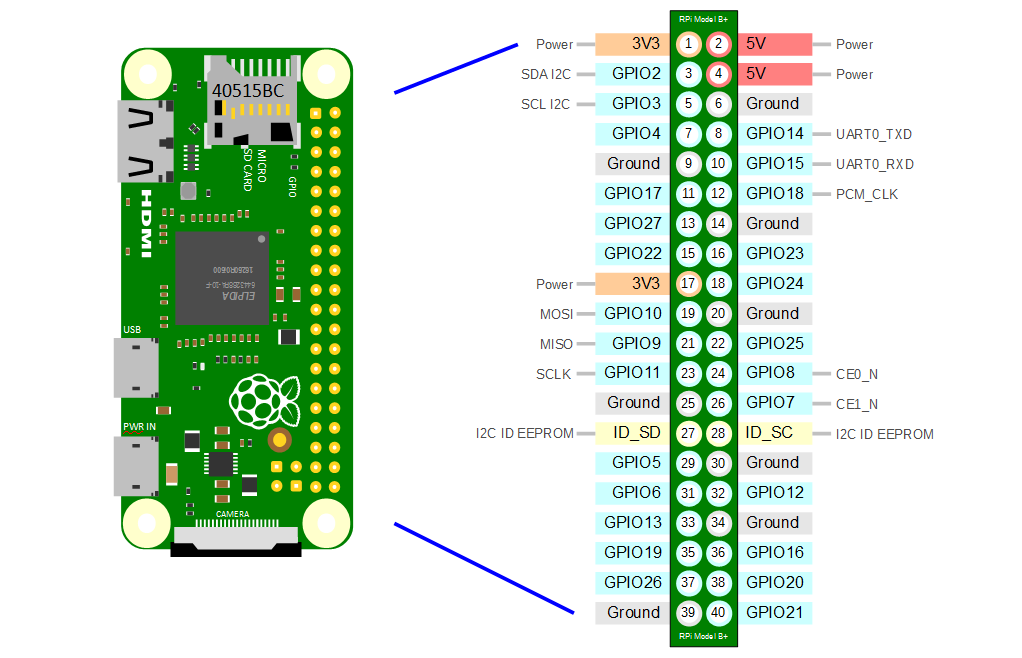
1. **Tự động hóa quy trình mượn và trả sách**: Sử dụng các thiết bị IoT như máy quét mã vạch hoặc cảm biến RFID, hệ thống có thể nhận biết và ghi nhận các thao tác mượn và trả sách mà không cần sự can thiệp của nhân viên thư viện. Điều này giúp tiết kiệm thời gian và giảm thiểu sai sót trong việc quản lý số lượng sách.
2. **Giám sát tài sản thư viện**: Các thiết bị IoT có thể được sử dụng để theo dõi vị trí, tình trạng và số lượng của sách và các tài liệu khác trong thư viện. Ví dụ, việc gắn thẻ barcode cho từng cuốn sách giúp dễ dàng kiểm kê, tìm kiếm và đảm bảo không có cuốn sách nào bị thất lạc.
3. **Tối ưu hóa quản lý dữ liệu và báo cáo**: Dữ liệu thu thập từ các thiết bị IoT có thể được phân tích để tạo ra các báo cáo chi tiết về số lượng sách mượn, sách trả, tần suất sử dụng tài liệu theo thời gian. Từ đó, thư viện có thể đưa ra các quyết định về việc mua sắm, bổ sung tài liệu một cách hợp lý.

Như vậy, IoT không chỉ giúp tự động hóa và tối ưu hóa các hoạt động hàng ngày tại thư viện, mà còn cung cấp các công cụ mạnh mẽ để giám sát, quản lý và phân tích tài liệu trong thư viện một cách hiệu quả hơn. Việc ứng dụng IoT vào quản lý thư viện trường học sẽ giúp nâng cao trải nghiệm của người dùng và giảm tải công việc cho các nhân viên thư viện.

## Raspberry Pi trong ứng dụng IoT:

### Tổng quan về Raspberry Pi

Raspberry Pi Zero 2 W là một trong những phiên bản nhỏ gọn và giá rẻ của dòng máy tính Raspberry Pi, được thiết kế với khả năng xử lý mạnh mẽ trong một kích thước siêu nhỏ. Được trang bị vi xử lý quad-core ARM Cortex-A53 64-bit và RAM 512MB, Raspberry Pi Zero 2 W cung cấp sức mạnh xử lý đủ cho các ứng dụng IoT cơ bản và trung bình. Với kích thước nhỏ gọn và khả năng kết nối không dây tích hợp (Wi-Fi và Bluetooth), Raspberry Pi Zero 2 W là một lựa chọn lý tưởng cho các dự án IoT, đặc biệt là trong môi trường có không gian hạn chế như các hệ thống nhúng.



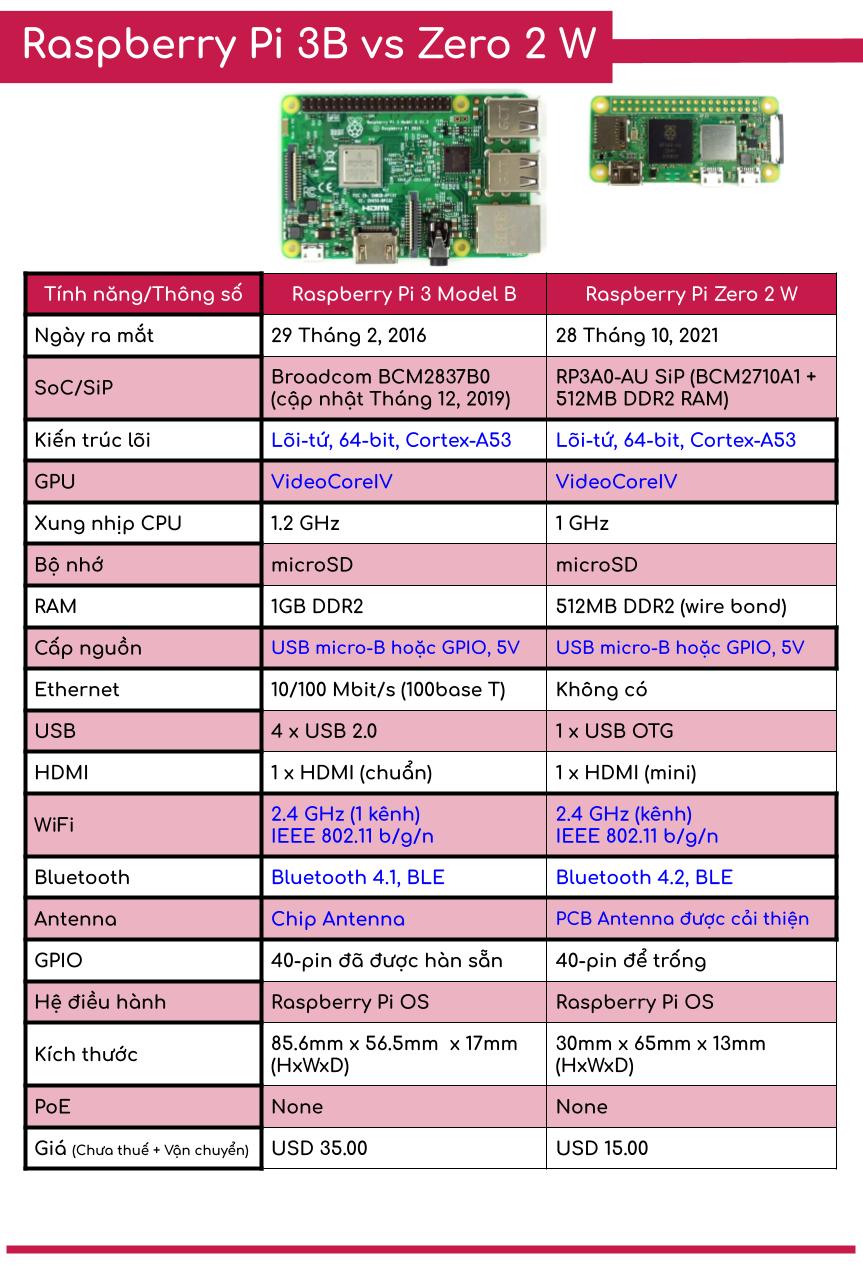
Hình 3 Raspberry Pi Zero 2 W

### Cấu hình và vai trò của Raspberry Pi trong hệ thống

Trong hệ thống quản lý thư viện kết hợp IoT, Raspberry Pi Zero 2 W sẽ đóng vai trò là máy chủ trung tâm, điều khiển và quản lý việc thu thập dữ liệu từ các thiết bị ngoại vi, chẳng hạn như máy quét sách hoặc cảm biến. Các ứng dụng cụ thể của Raspberry Pi Zero 2 W trong hệ thống bao gồm:

1. **Máy chủ Web**: Raspberry Pi Zero 2 W có thể được sử dụng để chạy một trang web quản lý thư viện dựa trên Express.js, giúp người dùng có thể truy cập thông tin về tình trạng sách, quản lý mượn trả, và xem các báo cáo từ xa thông qua trình duyệt web.
2. **Giao tiếp với thiết bị IoT**: Với khả năng kết nối Wi-Fi và Bluetooth tích hợp, Raspberry Pi Zero 2 W có thể giao tiếp với các thiết bị IoT như module ESP32 để thu thập dữ liệu từ các cảm biến, máy quét mã vạch, hoặc các thẻ RFID được gắn trên sách. Dữ liệu từ các thiết bị này sẽ được xử lý và lưu trữ trong cơ sở dữ liệu MySQL được chạy trên Raspberry Pi.
3. **Quản lý cơ sở dữ liệu**: Raspberry Pi có thể đóng vai trò là một máy chủ cơ sở dữ liệu MySQL, nơi lưu trữ toàn bộ thông tin liên quan đến sách, học sinh, và các hoạt động mượn trả. Khả năng xử lý của Raspberry Pi Zero 2 W đủ mạnh để đảm bảo các hoạt động truy vấn và cập nhật cơ sở dữ liệu diễn ra mượt mà, đồng thời dễ dàng triển khai các bản sao lưu dữ liệu định kỳ.
4. **Tích hợp với các thiết bị ngoại vi**: Raspberry Pi Zero 2 W có các cổng GPIO, cho phép kết nối với các thiết bị ngoại vi như đèn LED, chuông báo, hoặc các cảm biến khác. Điều này giúp hệ thống có thể phát ra các tín hiệu cảnh báo hoặc điều khiển các thiết bị vật lý khi cần thiết, ví dụ như khi phát hiện sách bị thất thoát hoặc khi hệ thống cần xác nhận trạng thái mượn trả của người dùng.

### Ưu điểm khi sử dụng Raspberry Pi làm máy chủ



Hình 4 So sánh Raspberry Pi Zero 2 W và Raspberry Pi 3 Model B

Việc sử dụng Raspberry Pi Zero 2 W trong hệ thống quản lý thư viện mang lại nhiều lợi ích đáng kể:

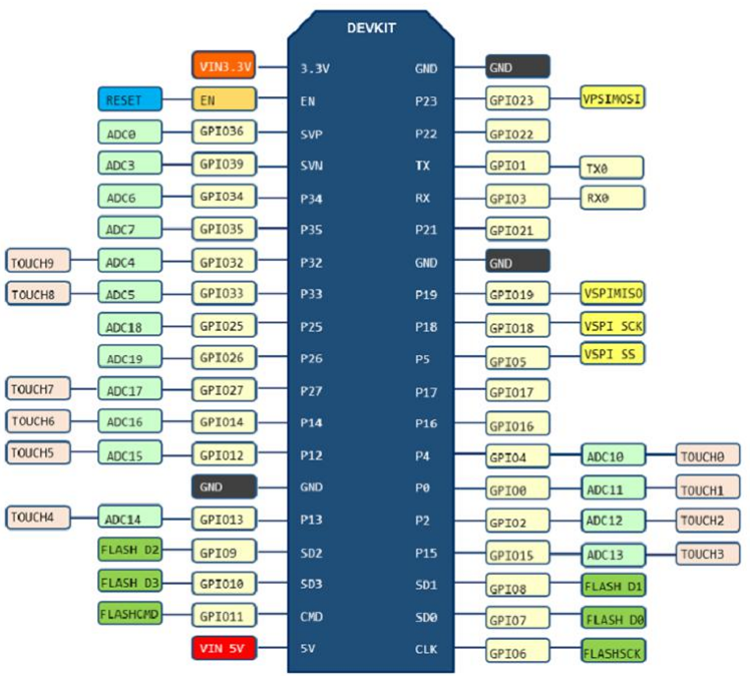
1. **Kích thước nhỏ gọn và tiết kiệm năng lượng**: Với thiết kế nhỏ gọn, Raspberry Pi Zero 2 W có thể được cài đặt ở bất kỳ vị trí nào trong thư viện mà không chiếm nhiều không gian. Bên cạnh đó, nó tiêu thụ rất ít điện năng, giúp tiết kiệm chi phí vận hành.
2. **Khả năng xử lý mạnh mẽ trong các ứng dụng IoT**: Dù có kích thước nhỏ, Raspberry Pi Zero 2 W vẫn đủ khả năng xử lý các tác vụ quan trọng như quản lý cơ sở dữ liệu, điều khiển các thiết bị IoT, và vận hành máy chủ web. Điều này giúp nó trở thành lựa chọn lý tưởng cho các hệ thống nhúng có quy mô nhỏ hoặc trung bình.
3. **Kết nối không dây linh hoạt**: Raspberry Pi Zero 2 W có khả năng kết nối không dây, giúp dễ dàng thiết lập hệ thống mà không cần đến dây cáp mạng phức tạp. Điều này đặc biệt hữu ích trong môi trường thư viện, nơi các thiết bị có thể được phân bổ ở nhiều vị trí khác nhau.
4. **Chi phí thấp và dễ dàng mở rộng**: Raspberry Pi Zero 2 W có giá thành rất hợp lý, giúp giảm chi phí đầu tư ban đầu cho hệ thống. Đồng thời, nhờ vào cộng đồng người dùng rộng lớn và nguồn tài liệu phong phú, việc mở rộng hoặc nâng cấp hệ thống trong tương lai cũng trở nên dễ dàng hơn.

#### Ứng dụng của Raspberry Pi Zero 2 W trong hệ thống IoT quản lý thư viện:

Raspberry Pi Zero 2 W sẽ là bộ não của hệ thống quản lý thư viện IoT, điều khiển toàn bộ quy trình mượn trả sách, giám sát thiết bị IoT và cung cấp giao diện quản lý trực quan cho người dùng thông qua ứng dụng web. Với khả năng tích hợp linh hoạt, nó sẽ giúp tự động hóa hoạt động của thư viện, mang lại hiệu quả cao hơn trong quản lý và giảm thiểu thời gian cho các công việc thủ công.

## ESP32 và các ứng dụng trong IoT:

### Giới thiệu về ESP32



Hình 5 Vi điều khiển ESP32

ESP32 là một vi điều khiển với khả năng kết nối không dây mạnh mẽ, được phát triển bởi Espressif Systems. Với việc tích hợp cả Wi-Fi và Bluetooth, ESP32 được sử dụng rộng rãi trong các ứng dụng IoT nhờ vào hiệu suất cao, tiết kiệm năng lượng và tính linh hoạt. ESP32 sử dụng vi xử lý dual-core Xtensa® 32-bit LX6, cho phép xử lý đa nhiệm với tốc độ lên đến 240 MHz, giúp thực hiện các tác vụ tính toán phức tạp và xử lý nhanh chóng.

Các đặc điểm nổi bật của ESP32:

* **Wi-Fi và Bluetooth tích hợp**: ESP32 hỗ trợ kết nối Wi-Fi (802.11 b/g/n) và Bluetooth (Classic và BLE), giúp nó trở thành nền tảng lý tưởng cho các ứng dụng IoT.
* **Khả năng xử lý mạnh mẽ**: Vi xử lý dual-core với tốc độ lên tới 240MHz, giúp thực hiện các tác vụ nhanh chóng.
* **Giao tiếp linh hoạt**: ESP32 hỗ trợ nhiều giao diện như GPIO, ADC, DAC, UART, SPI, I2C, PWM, giúp kết nối và điều khiển các thiết bị ngoại vi như cảm biến, màn hình, mạch điều khiển.
* **Tiết kiệm năng lượng**: ESP32 có nhiều chế độ tiết kiệm năng lượng, đặc biệt hữu ích trong các ứng dụng IoT cần hoạt động liên tục với thời gian dài.

Với những đặc điểm này, ESP32 trở thành một lựa chọn tối ưu cho các ứng dụng IoT, bao gồm cả việc xây dựng các hệ thống thông minh như hệ thống quản lý thư viện IoT mà chúng ta đang phát triển.

### Khả năng giao tiếp không dây và vai trò trong hệ thống quét sách

Một trong những yếu tố quan trọng giúp ESP32 trở thành vi điều khiển lý tưởng cho các ứng dụng IoT chính là khả năng giao tiếp không dây mạnh mẽ qua Wi-Fi và Bluetooth. Đặc biệt, khi tích hợp vào các hệ thống quét sách trong thư viện, ESP32 giúp giảm bớt các bước thủ công trong việc quản lý sách và cải thiện tính chính xác của các quy trình.

#### **Khả năng giao tiếp không dây của ESP32:**

* **Wi-Fi**: ESP32 có khả năng kết nối với các mạng Wi-Fi, giúp truyền tải dữ liệu từ các thiết bị ngoại vi (như máy quét mã vạch hoặc cảm biến RFID) đến hệ thống máy chủ (Raspberry Pi trong trường hợp của bạn) một cách nhanh chóng và ổn định.
* **Bluetooth**: Hỗ trợ giao tiếp Bluetooth cho phép ESP32 kết nối với các thiết bị cầm tay hoặc các thiết bị không dây khác trong phạm vi gần.

Trong hệ thống quản lý thư viện IoT, ESP32 có thể được sử dụng để:

* **Quét mã vạch sách**: ESP32 có thể kết nối với các thiết bị quét mã vạch hoặc cảm biến RFID để nhận diện sách tự động khi người dùng mượn hoặc trả sách.
* **Truyền tải dữ liệu về máy chủ**: Sau khi nhận diện sách, ESP32 có thể truyền tải thông tin về sách (ID, tên sách, trạng thái mượn/trả) qua mạng Wi-Fi đến Raspberry Pi hoặc cơ sở dữ liệu trung tâm để xử lý.
* **Tích hợp các cảm biến thông minh**: ESP32 có thể kết nối với các cảm biến thông minh khác như cảm biến nhiệt độ, độ ẩm trong thư viện để giám sát môi trường, đảm bảo điều kiện tốt nhất cho sách.

ESP32 giúp hệ thống quét sách trở nên linh hoạt và tiết kiệm thời gian, đồng thời giảm thiểu lỗi do con người gây ra trong quá trình quản lý.

### Tích hợp ESP32 với hệ thống quản lý

ESP32 có thể kết nối với Raspberry Pi Zero 2 W qua Wi-Fi để truyền tải dữ liệu về trạng thái của các sách trong thư viện. Raspberry Pi sẽ đóng vai trò như một máy chủ xử lý, quản lý dữ liệu từ ESP32 và cung cấp giao diện web cho người dùng để thực hiện các thao tác như mượn sách, trả sách, hoặc kiểm tra tình trạng sách.

#### **Tự động hóa quy trình:**

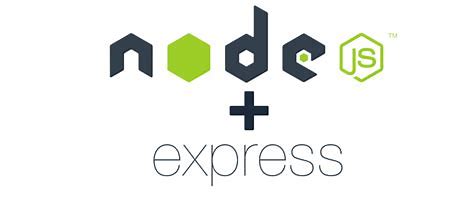
Việc tích hợp ESP32 vào hệ thống quản lý giúp tự động hóa các quy trình như:

* **Mượn sách**: Khi người dùng quét mã vạch hoặc RFID của sách, ESP32 sẽ gửi dữ liệu về Raspberry Pi để cập nhật trạng thái sách trong cơ sở dữ liệu.
* **Trả sách**: Khi người dùng trả sách, ESP32 sẽ nhận diện và cập nhật thông tin về việc trả sách vào hệ thống, giúp quản lý chính xác tình trạng sách.
* **Quản lý sách mất/mò**: Với sự tích hợp của ESP32 và các cảm biến thông minh, hệ thống có thể cảnh báo khi có sách bị mất hoặc bị lỗi.

## Express.js trong phát triển web:

### Giới thiệu về Express.js

Express.js là một framework phát triển web nhẹ, nhanh và đơn giản được xây dựng trên nền Node.js, cung cấp các tính năng cơ bản để xây dựng các ứng dụng web và API. Được ra mắt lần đầu tiên vào năm 2010, Express.js đã nhanh chóng trở thành một trong những framework phổ biến nhất cho Node.js nhờ vào tính linh hoạt, dễ sử dụng và khả năng mở rộng mạnh mẽ.



Hình 6 Express frame work

Express.js cung cấp các tính năng quan trọng như:

* **Routing**: Cho phép quản lý các route (đường dẫn) và xử lý các yêu cầu HTTP từ client (GET, POST, PUT, DELETE, v.v.).
* **Middleware**: Hỗ trợ các đoạn mã trung gian (middleware) để xử lý các yêu cầu, như kiểm tra xác thực, logging, xử lý lỗi, v.v.
* **Template engines**: Hỗ trợ các engines như EJS, Pug, Handlebars, giúp tạo các giao diện động.
* **Đơn giản hóa quản lý HTTP requests và responses**: Express cung cấp một API dễ hiểu để xử lý các yêu cầu và trả về phản hồi một cách hiệu quả.
* **Tích hợp cơ sở dữ liệu**: Có thể dễ dàng kết nối với các hệ quản trị cơ sở dữ liệu như MongoDB, MySQL, PostgreSQL thông qua các thư viện hoặc ORM như Mongoose, Sequelize.

Với các đặc điểm này, Express.js trở thành một công cụ tuyệt vời trong việc xây dựng các ứng dụng web, đặc biệt là khi cần xây dựng các hệ thống có khả năng mở rộng và tối ưu cho các ứng dụng real-time.

### Lý do lựa chọn Express.js trong phát triển trang web quản lý thư viện

Việc lựa chọn Express.js cho việc phát triển trang web quản lý thư viện IoT của chúng ta có nhiều lý do thuyết phục. Dưới đây là các lý do quan trọng:

* **Đơn giản và dễ học**: Express.js dễ dàng tiếp cận và có cú pháp rất đơn giản, giúp lập trình viên có thể xây dựng các API hoặc trang web nhanh chóng mà không cần phải học quá nhiều về cấu trúc phức tạp. Với nền tảng Node.js, Express giúp việc xử lý các yêu cầu HTTP trở nên nhanh chóng và hiệu quả.
* **Khả năng mở rộng và linh hoạt**: Express.js hỗ trợ một cách dễ dàng việc mở rộng hệ thống, cho phép thêm nhiều tính năng mà không gặp phải các vấn đề phức tạp về cấu hình hay cấu trúc. Điều này rất quan trọng khi xây dựng hệ thống quản lý thư viện, nơi cần kết nối với nhiều dịch vụ như cơ sở dữ liệu, các thiết bị IoT (ESP32, Raspberry Pi), và các giao diện người dùng khác nhau.
* **Tích hợp dễ dàng với cơ sở dữ liệu**: Express.js hỗ trợ rất tốt việc kết nối với các cơ sở dữ liệu như MySQL, MongoDB, PostgreSQL thông qua các thư viện như mysql, mongoose và pg. Điều này giúp xây dựng hệ thống quản lý thư viện dễ dàng với tính năng quản lý sách, mượn trả sách, và lưu trữ thông tin người dùng.
* **Tương thích với REST API**: Express.js là framework lý tưởng để xây dựng các API RESTful, cho phép giao tiếp giữa các thiết bị như ESP32 hoặc Raspberry Pi với web server. Điều này rất cần thiết trong hệ thống quản lý thư viện IoT của bạn, nơi các dữ liệu như thông tin sách, trạng thái mượn sách sẽ được truyền tải giữa các thiết bị và cơ sở dữ liệu.
* **Tích hợp với các công nghệ khác**: Express.js dễ dàng tích hợp với các công nghệ như WebSocket, giúp xây dựng các ứng dụng có khả năng tương tác theo thời gian thực (real-time). Điều này có thể ứng dụng trong việc theo dõi tình trạng mượn sách hay kiểm soát việc trả sách trong hệ thống quản lý thư viện.

### Các tính năng quan trọng của Express.js

Trong việc phát triển hệ thống thư viện online thông qua website, Express.js cung cấp nhiều tính năng quan trọng để hỗ trợ phát triển ứng dụng. Các tính năng quan trọng bao gồm:

1. **Middleware và xử lý yêu cầu**:
   * Express.js hỗ trợ các đoạn mã middleware, cho phép xử lý các yêu cầu từ người dùng trước khi đưa ra phản hồi. Ví dụ, bạn có thể thêm các middleware để kiểm tra quyền truy cập của người dùng khi mượn sách hoặc kiểm tra trạng thái của thiết bị trước khi quét mã vạch.
   * Middleware còn có thể được sử dụng để xử lý các lỗi hệ thống, đảm bảo rằng hệ thống thư viện hoạt động ổn định và an toàn.
2. **Quản lý tuyến đường (Routing)**:
   * Express.js cho phép bạn dễ dàng định nghĩa các tuyến đường cho các yêu cầu từ người dùng như trang chính, trang tìm kiếm sách, trang đăng nhập và các API cho hệ thống quản lý sách. Bạn có thể tạo các route như /books, /borrow, /return, và /search, giúp việc tương tác với hệ thống trở nên mượt mà.
   * Các route này có thể hỗ trợ các yêu cầu HTTP như GET, POST, PUT và DELETE, cho phép thực hiện các chức năng như thêm sách vào thư viện, mượn sách, và trả sách.
3. **Hỗ trợ Template Engines**:
   * Express.js hỗ trợ các template engine như EJS, Pug, hoặc Handlebars, giúp bạn tạo ra các giao diện người dùng động cho hệ thống quản lý thư viện. Các giao diện này có thể hiển thị danh sách sách, các thông tin về sách và lịch sử mượn trả của người dùng, tất cả được cập nhật tự động từ cơ sở dữ liệu.
4. **Tích hợp cơ sở dữ liệu**:
   * Express.js dễ dàng tích hợp với các cơ sở dữ liệu quan hệ như MySQL hoặc các cơ sở dữ liệu NoSQL như MongoDB. Điều này giúp quản lý thông tin sách, thông tin người dùng, mượn trả sách và lịch sử dễ dàng hơn. Với các ORM như Sequelize (cho MySQL) hoặc Mongoose (cho MongoDB), việc kết nối và thao tác với cơ sở dữ liệu trở nên đơn giản và hiệu quả.
5. **Kết nối với các thiết bị IoT**:
   * Một điểm mạnh của Express.js là khả năng tích hợp với các thiết bị IoT như ESP32 và Raspberry Pi. Dữ liệu từ các cảm biến hoặc thiết bị quét mã vạch có thể được gửi từ ESP32 về máy chủ Raspberry Pi sử dụng Express.js qua API RESTful, giúp hệ thống quản lý thư viện hoạt động trơn tru và tự động.
6. **Hỗ trợ API RESTful**:
   * Express.js hỗ trợ xây dựng các API RESTful để giao tiếp giữa client và server. Điều này rất quan trọng trong việc quản lý thư viện online, vì bạn sẽ cần xây dựng các API để thao tác với cơ sở dữ liệu, thực hiện các phép toán liên quan đến sách, mượn sách, trả sách và cập nhật tình trạng của sách.

Với các tính năng mạnh mẽ như vậy, Express.js thực sự là một công cụ lý tưởng để phát triển hệ thống thư viện IoT của bạn, đảm bảo khả năng mở rộng, hiệu suất và tính linh hoạt khi triển khai các tính năng quản lý sách online.

## MySQL và quản lý cơ sở dữ liệu trong hệ thống:

### Giới thiệu về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến, mã nguồn mở và miễn phí, được phát triển bởi Oracle. MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để truy vấn và quản lý dữ liệu. Nó là một trong những hệ quản trị cơ sở dữ liệu được sử dụng rộng rãi trong các ứng dụng web, với khả năng xử lý các yêu cầu cao và dễ dàng tích hợp vào các hệ thống khác nhau.



Hình 7 My SQL database development frame work

Các tính năng nổi bật của MySQL:

* **Quản lý dữ liệu quan hệ**: MySQL hỗ trợ các bảng dữ liệu với các mối quan hệ giữa chúng, có thể sử dụng các khóa ngoại và các ràng buộc dữ liệu để duy trì tính toàn vẹn của dữ liệu.
* **Hiệu suất cao**: MySQL được tối ưu hóa để xử lý nhanh các truy vấn với số lượng lớn dữ liệu.
* **Khả năng mở rộng**: MySQL hỗ trợ các tính năng như phân mảnh cơ sở dữ liệu và replication để mở rộng quy mô hệ thống.
* **Độ tin cậy và bảo mật**: MySQL cung cấp các tính năng bảo mật mạnh mẽ, như mã hóa kết nối và kiểm soát quyền truy cập người dùng.
* **Hỗ trợ nhiều nền tảng**: MySQL có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.

Với các tính năng này, MySQL là một sự lựa chọn lý tưởng cho các hệ thống quản lý thư viện, nơi có nhu cầu lưu trữ và truy xuất thông tin sách, người dùng, lịch sử mượn sách, và các tính năng liên quan.

### Mô hình cơ sở dữ liệu trong quản lý thư viện

Trong hệ thống quản lý thư viện, cơ sở dữ liệu có vai trò quan trọng trong việc lưu trữ và quản lý thông tin về sách, người dùng, mượn trả sách và các giao dịch khác. Mô hình cơ sở dữ liệu thường được thiết kế theo dạng các bảng, với các mối quan hệ giữa chúng.

Một mô hình cơ sở dữ liệu đơn giản cho hệ thống quản lý thư viện có thể bao gồm các bảng sau:

1. **Bảng Sách (Books)**:
   * **Item\_id** (INT): Mã sách (khóa chính).
   * **Title** (VARCHAR): Tên sách.
   * **Author** (VARCHAR): Tác giả.
   * **Genre** (VARCHAR): Thể loại sách.
   * **Year\_published** (YEAR): Năm xuất bản.
   * **Available\_copies** (INT): Số lượng sách có sẵn.
2. **Bảng Người dùng (Users)**:
   * **User\_id** (INT): Mã người dùng (khóa chính).
   * **User\_name** (VARCHAR): Tên người dùng.
   * **Email** (VARCHAR): Địa chỉ email.
   * **Password** (VARCHAR): Mật khẩu.
   * **Role** (VARCHAR): Quyền hạn (ví dụ: sinh viên, giảng viên).
3. **Bảng Mượn sách (Borrowing)**:
   * **Borrow\_id** (INT): Mã mượn sách (khóa chính).
   * **User\_id** (INT): Mã người dùng (khóa ngoại liên kết với bảng Users).
   * **Item\_id** (INT): Mã sách (khóa ngoại liên kết với bảng Books).
   * **Borrow\_date** (DATE): Ngày mượn.
   * **Return\_date** (DATE): Ngày trả.
4. **Bảng Lịch sử mượn sách (Borrowing\_history)**:
   * **History\_id** (INT): Mã lịch sử mượn (khóa chính).
   * **User\_id** (INT): Mã người dùng.
   * **Item\_id** (INT): Mã sách.
   * **Borrow\_date** (DATE): Ngày mượn.
   * **Return\_date** (DATE): Ngày trả.

Mối quan hệ giữa các bảng:

* Mối quan hệ giữa **Users** và **Borrowing** là một-nhiều (một người dùng có thể mượn nhiều sách).
* Mối quan hệ giữa **Books** và **Borrowing** là một-nhiều (một cuốn sách có thể được mượn nhiều lần).

Ngoài ra, chúng ta có thể mở rộng mô hình này với các bảng bổ sung như **Categories** (danh mục sách), **Reviews** (đánh giá sách), hoặc **Reservations** (đặt sách).

### Kết nối MySQL với Express.js

Kết nối MySQL với Express.js là một phần quan trọng trong việc xây dựng hệ thống quản lý thư viện, giúp hệ thống có thể tương tác với cơ sở dữ liệu để thực hiện các thao tác như truy vấn, thêm, sửa, và xóa dữ liệu. Để kết nối MySQL với Express.js, bạn có thể sử dụng thư viện mysql hoặc mysql2.

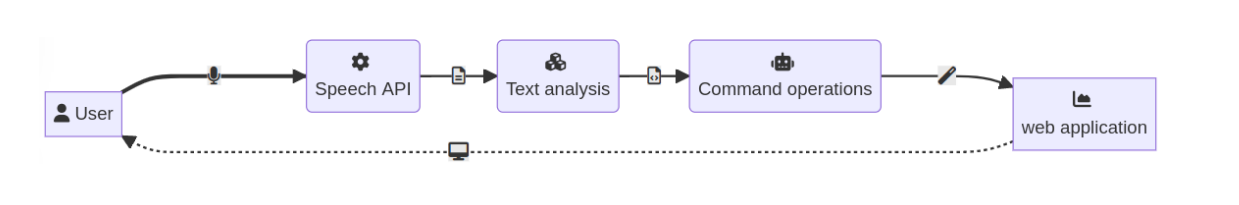
Chúng ta có thể dễ dàng kết nối đến database Mysql thông qua code java script trong đoạn code ví dụ mẫu sau

|  |
| --- |
| const mysql = require('mysql');  // create Mysql database object  const db = mysql.createConnection({  host: 'localhost',  user: 'root',  password: 'yourpassword',  database: 'test' // tên cơ sở dữ liệu của bạn  });  // Kết nối đến cơ sở dữ liệu  db.connect((err) => {  if (err) {  console.error('Lỗi kết nối MySQL: ' + err.stack);  return;  }  console.log('Kết nối MySQL thành công');  }); |

## Tìm hiểu về Web Speech API

### **Tổng quan về Web Speech API**

Web Speech API là một API của trình duyệt, được phát triển nhằm cung cấp khả năng nhận diện giọng nói và tổng hợp giọng nói (speech synthesis) trực tiếp trong các ứng dụng web. Nó giúp các nhà phát triển xây dựng những ứng dụng có khả năng tương tác bằng giọng nói, tăng cường trải nghiệm người dùng thông qua giao diện thoại.



Hình 8 Web Speech API flow chart

Web Speech API bao gồm hai thành phần chính:

* **Speech Recognition (Nhận diện giọng nói)**: Cho phép trang web nhận dạng và xử lý đầu vào là giọng nói từ người dùng, chuyển đổi âm thanh thành văn bản.
* **Speech Synthesis (Tổng hợp giọng nói)**: Cho phép trang web phát văn bản thành âm thanh, tạo ra giọng nói tự động thông qua trình duyệt.

### **Lợi ích và ứng dụng của Web Speech API**

Web Speech API mang lại nhiều lợi ích cho các ứng dụng web, bao gồm:

* **Tăng cường khả năng tiếp cận**: Những người dùng gặp khó khăn trong việc sử dụng bàn phím hoặc chuột có thể tương tác với trang web bằng giọng nói.
* **Tạo ra trải nghiệm tương tác phong phú**: Web Speech API giúp tạo ra các ứng dụng hỗ trợ trợ lý ảo, công cụ dịch thuật, tìm kiếm bằng giọng nói, và các ứng dụng điều khiển bằng giọng nói.
* **Ứng dụng trong IoT**: Đối với các hệ thống IoT, Web Speech API có thể được tích hợp để điều khiển các thiết bị thông minh thông qua giọng nói.

### **Cấu trúc và hoạt động của Web Speech API**

##### **Speech Recognition API (Nhận diện giọng nói)**

**Speech Recognition API** là thành phần dùng để nhận diện giọng nói từ micro, xử lý và trả về kết quả dưới dạng văn bản. Nó hoạt động bằng cách lắng nghe đầu vào giọng nói từ người dùng, sau đó chuyển đổi thành các kết quả văn bản (transcript).

* **Các thuộc tính quan trọng:**
  + SpeechRecognition.interimResults: Nếu được đặt là true, API sẽ trả về kết quả tạm thời trong quá trình nhận diện, giúp cung cấp kết quả nhanh hơn.
  + SpeechRecognition.continuous: Nếu được đặt là true, API sẽ tiếp tục nhận diện giọng nói cho đến khi dừng.
* **Các sự kiện chính:**
  + result: Sự kiện được kích hoạt khi API có kết quả nhận diện.
  + end: Sự kiện được kích hoạt khi quá trình nhận diện giọng nói kết thúc.

**Ví dụ nhận diện giọng nói:**

Đoạn code dưới đây sẽ được nhúng vào trong trang web html và sẽ được chạy ở phía trình duyệt người dùng.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>Speech Recognition Example</title>  </head>  <body>  <h1>Speech Recognition Demo</h1>  <button id="start-btn">Start Recognition</button>  <p id="result">Speak something...</p>  <script>  // Kiểm tra xem trình duyệt có hỗ trợ Web Speech API không  if ('SpeechRecognition' in window || 'webkitSpeechRecognition' in window) {  const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;  const recognition = new SpeechRecognition();  recognition.interimResults = true;  recognition.addEventListener('result', event => {  const transcript = Array.from(event.results)  .map(result => result[0])  .map(result => result.transcript)  .join('');  document.getElementById('result').textContent = transcript;  });  document.getElementById('start-btn').addEventListener('click', () => {  recognition.start();  });  } else {  alert('Web Speech API is not supported in your browser.');  }  </script>  </body>  </html> |

### Tính tương thích và các trình duyệt hỗ trợ

Web Speech API hiện tại được hỗ trợ trên hầu hết các trình duyệt hiện đại như Chrome, Firefox, Edge (dựa trên Chromium), nhưng có thể không hoạt động trên một số trình duyệt cũ hoặc một số trình duyệt di động. Đặc biệt, với API nhận diện giọng nói, một số trình duyệt có thể yêu cầu kết nối HTTPS hoặc mạng internet để sử dụng.

Web Speech API là công cụ mạnh mẽ giúp các nhà phát triển xây dựng ứng dụng có tính tương tác cao bằng giọng nói. Từ việc nhận diện giọng nói đến tổng hợp giọng nói, API này mở ra nhiều cơ hội cho việc phát triển các ứng dụng hỗ trợ giọng nói hiện đại, giúp cải thiện trải nghiệm người dùng, và phù hợp với các ứng dụng IoT.

API này không chỉ đơn thuần là một công nghệ mới mẻ mà còn là xu hướng quan trọng trong phát triển web, đặc biệt trong bối cảnh sự phổ biến của các thiết bị thông minh và trợ lý ảo.

## Kết luận chương

Như vậy chương 2 của luận văn đã trình bày các khái niệm cơ bản và lý thuyết liên quan đến các công nghệ quan trọng được sử dụng trong việc xây dựng hệ thống quản lý thư viện kết hợp với IoT. Qua đó, chúng ta đã tìm hiểu về IoT, Raspberry Pi, ESP32, Express.js, và MySQL – những thành phần chủ chốt giúp triển khai hệ thống thư viện thông minh.

* **IoT** là nền tảng của hệ thống, giúp kết nối các thiết bị, cảm biến và các hệ thống khác nhau để thu thập và trao đổi dữ liệu trong môi trường thư viện.
* **Raspberry Pi Zero 2 W** với kích thước nhỏ gọn và hiệu suất mạnh mẽ là lựa chọn tối ưu để chạy trang web quản lý thư viện, đồng thời tương tác với các thiết bị ngoại vi như ESP32 để quét mã vạch sách.
* **ESP32**, với khả năng giao tiếp không dây và xử lý tốt các kết nối mạng, sẽ giúp quét mã vạch sách và truyền tải dữ liệu về hệ thống quản lý thư viện.
* **Express.js** là một framework JavaScript mạnh mẽ cho Node.js, hỗ trợ phát triển các API web giúp quản lý sách, người dùng và các giao dịch mượn trả sách.
* **MySQL** là hệ quản trị cơ sở dữ liệu quan hệ, nơi lưu trữ toàn bộ thông tin liên quan đến sách, người dùng và các giao dịch trong hệ thống.

Những công nghệ này không chỉ đảm bảo tính hiệu quả và mở rộng của hệ thống mà còn giúp hệ thống thư viện trở nên thông minh hơn với các tính năng tự động hóa, tối ưu hóa quy trình mượn trả sách và quản lý dữ liệu một cách chính xác và nhanh chóng.

Phần tiếp theo của luận văn sẽ đi sâu vào việc thiết kế và triển khai hệ thống thư viện kết hợp với IoT, với các mô-đun phần mềm và phần cứng được tích hợp để đáp ứng các yêu cầu của đề tài.

# THIẾT KẾ HỆ THỐNG THƯ VIỆN

## Tổng quan về hệ thống thư viện

### Mục tiêu và chức năng của hệ thống

Hệ thống quản lý thư viện kết hợp IoT được xây dựng nhằm mục đích nâng cao hiệu quả quản lý tài nguyên thư viện, giảm thiểu sai sót và cải thiện trải nghiệm của người dùng. Thông qua việc ứng dụng công nghệ IoT và các thiết bị thông minh, hệ thống không chỉ giúp tối ưu hóa quy trình mượn, trả sách mà còn hỗ trợ tự động hóa các hoạt động liên quan đến quản lý sách và người dùng.

**Mục tiêu cụ thể của hệ thống**:

* **Tự động hóa quá trình quản lý thư viện**: Hệ thống sẽ tự động hóa các hoạt động như mượn, trả sách và kiểm soát tồn kho thông qua các thiết bị quét mã vạch và các cảm biến.
* **Giảm thiểu thời gian và công sức**: Hệ thống giúp giảm thiểu thời gian chờ đợi của người dùng khi mượn và trả sách. Đồng thời, cán bộ thư viện có thể quản lý sách và người dùng một cách hiệu quả hơn.
* **Cải thiện độ chính xác và hiệu suất quản lý**: Nhờ sử dụng các thiết bị IoT, dữ liệu được cập nhật nhanh chóng và chính xác, giúp tránh các lỗi sai trong quá trình xử lý thông tin.
* **Dễ dàng truy xuất thông tin**: Người dùng có thể dễ dàng tìm kiếm, tra cứu thông tin sách qua giao diện web của hệ thống và nhận thông tin cập nhật về tình trạng sách.

**Chức năng chính của hệ thống**:

* **Quản lý sách**: Hệ thống hỗ trợ việc thêm, xóa, cập nhật thông tin sách và kiểm tra tình trạng tồn kho sách hiện có.
* **Quản lý người dùng**: Hỗ trợ đăng ký tài khoản, quản lý thông tin người dùng và theo dõi các hoạt động mượn, trả sách của người dùng.
* **Mượn và trả sách**: Hệ thống giúp người dùng quét mã vạch để thực hiện việc mượn và trả sách một cách tự động.
* **Tra cứu và tìm kiếm sách**: Cho phép người dùng tìm kiếm sách thông qua tên sách, tác giả hoặc thể loại.

### Phạm vi ứng dụng của hệ thống thư viện kết hợp IoT

Hệ thống quản lý thư viện kết hợp IoT được thiết kế để phục vụ trong môi trường thư viện trường học. Với quy mô vừa và nhỏ, hệ thống này phù hợp để triển khai tại các trường học, nơi mà việc quản lý tài nguyên sách là một yêu cầu quan trọng. Hệ thống không chỉ giúp cán bộ thư viện quản lý số lượng lớn tài liệu mà còn mang đến sự thuận tiện cho học sinh và giáo viên trong việc tiếp cận tài liệu học tập.

**Phạm vi ứng dụng cụ thể**:

* **Quản lý sách học tập và tài liệu tham khảo**: Hệ thống giúp quản lý sách giáo khoa, tài liệu tham khảo và các ấn phẩm khác phục vụ cho việc học tập.
* **Tích hợp thiết bị IoT trong quản lý**: Các thiết bị như Raspberry Pi, ESP32 kết hợp với các cảm biến và máy quét mã vạch sẽ tạo nên một hệ thống tự động, giúp đơn giản hóa các quy trình mượn, trả sách và theo dõi tình trạng tài liệu.
* **Ứng dụng trong các hoạt động học tập và nghiên cứu**: Hệ thống giúp học sinh và giáo viên dễ dàng truy cập vào thông tin tài liệu, tra cứu tình trạng sách và quản lý tài liệu phục vụ cho học tập và nghiên cứu.

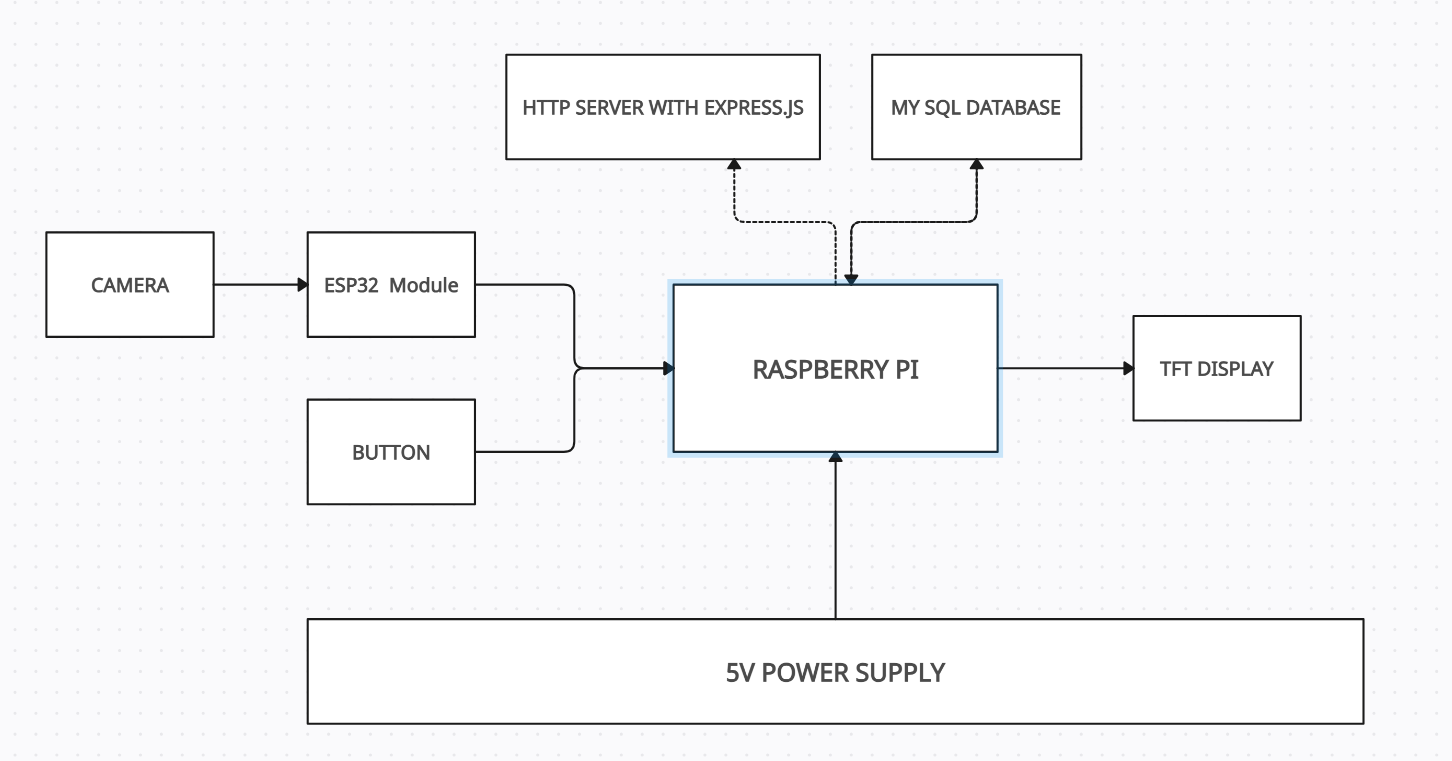
Với sự kết hợp của IoT, hệ thống không chỉ hoạt động trong phạm vi thư viện vật lý mà còn mở rộng khả năng theo dõi, giám sát từ xa thông qua các nền tảng web. Điều này giúp các bên liên quan có thể quản lý hệ thống thư viện một cách dễ dàng và linh hoạt hơn.

Phần này sẽ giúp cung cấp cái nhìn tổng quan về hệ thống thư viện kết hợp IoT và các chức năng, mục tiêu chính của nó trong phạm vi ứng dụng của chúng ta.

## Kiến trúc hệ thống

### Mô hình tổng thể hệ thống

Hệ thống quản lý thư viện kết hợp IoT được thiết kế với kiến trúc gồm hai thành phần chính: phần cứng và phần mềm. Các thiết bị phần cứng, như Raspberry Pi và ESP32, đóng vai trò trong việc kết nối và giao tiếp với các thiết bị ngoại vi như máy quét mã vạch và các cảm biến. Trong khi đó, phần mềm sẽ bao gồm ứng dụng web viết bằng Express.js, cơ sở dữ liệu MySQL và các API hỗ trợ giao tiếp và trao đổi dữ liệu giữa các thành phần.



Hình 9 Sơ đồ mô hình tổng thể hệ thống thư viện

Mô hình tổng thể của hệ thống bao gồm:

* **Thiết bị người dùng**: Người dùng truy cập hệ thống thông qua trình duyệt web để mượn và trả sách, quản lý tài khoản cá nhân và tra cứu tài liệu.
* **Hệ thống máy chủ**: Raspberry Pi Zero 2 W đóng vai trò là máy chủ, nơi vận hành ứng dụng web và xử lý các yêu cầu từ người dùng.
* **Module IoT ESP32**: Dùng để quét mã vạch của sách, giúp tự động hóa quy trình mượn, trả sách.
* **Cơ sở dữ liệu**: MySQL được sử dụng để lưu trữ và quản lý toàn bộ thông tin về sách, người dùng và lịch sử mượn trả.

### Các thành phần phần cứng

**+ Raspberry Pi Zero 2 W**   
 Đây là thành phần chính của hệ thống, đóng vai trò là máy chủ để vận hành ứng dụng web và quản lý cơ sở dữ liệu. Raspberry Pi là một máy tính nhỏ gọn, chi phí thấp nhưng có hiệu suất đủ mạnh để chạy các ứng dụng web và quản lý hệ thống IoT.

**+ ESP32**  
 ESP32 là một vi điều khiển mạnh mẽ và linh hoạt, hỗ trợ kết nối Wi-Fi và Bluetooth, giúp hệ thống kết nối với các thiết bị ngoại vi như máy quét mã vạch. Trong hệ thống này, ESP32 sẽ được tích hợp để thực hiện nhiệm vụ quét mã vạch sách và gửi thông tin về Raspberry Pi để xử lý.

**+ Các thiết bị ngoại vi**

* **Máy quét mã vạch**: Được kết nối với ESP32 để quét mã sách, giúp người dùng thực hiện các thao tác mượn, trả sách tự động.
* **Cảm biến hoặc các thiết bị khác**: Các cảm biến có thể được sử dụng để giám sát trạng thái hoặc môi trường trong thư viện (tùy theo nhu cầu mở rộng).

### Các thành phần phần mềm

**+ Express.js**

Express.js là framework của Node.js, đóng vai trò là công cụ để xây dựng ứng dụng web phía máy chủ. Trong hệ thống, Express.js chịu trách nhiệm xử lý các yêu cầu HTTP từ người dùng, tương tác với cơ sở dữ liệu và giao tiếp với các thiết bị IoT.

**+ MySQL**

MySQL là hệ quản trị cơ sở dữ liệu quan hệ được sử dụng để lưu trữ và quản lý dữ liệu của hệ thống. Toàn bộ thông tin về sách, người dùng, và lịch sử mượn trả sách đều được lưu trữ trong MySQL, cho phép hệ thống truy vấn và xử lý dữ liệu nhanh chóng.

**+ Các API liên quan**  
Các API sẽ được sử dụng để giao tiếp giữa các thành phần trong hệ thống, bao gồm:

* **API giữa Raspberry Pi và ESP32**: Cho phép Raspberry Pi nhận dữ liệu từ ESP32, như thông tin mã vạch sách, và xử lý các yêu cầu mượn trả sách.
* **API giao tiếp giữa Raspberry Pi và cơ sở dữ liệu**: Express.js sử dụng các truy vấn MySQL để thao tác với dữ liệu, từ việc truy xuất thông tin sách đến cập nhật lịch sử mượn trả.

### Mối quan hệ giữa các thành phần trong hệ thống

Mối quan hệ giữa các thành phần trong hệ thống quản lý thư viện kết hợp IoT hoạt động như sau:

* **Người dùng**: Truy cập hệ thống thông qua trình duyệt web để mượn, trả sách hoặc tra cứu thông tin.
* **Máy chủ (Raspberry Pi)**: Nhận các yêu cầu từ người dùng qua giao diện web và từ ESP32 qua các API. Raspberry Pi xử lý dữ liệu, thực hiện các tác vụ logic, và tương tác với cơ sở dữ liệu MySQL.
* **ESP32**: Là thiết bị IoT giao tiếp với Raspberry Pi để cung cấp thông tin quét mã vạch từ sách.
* **MySQL**: Cơ sở dữ liệu lưu trữ toàn bộ dữ liệu của hệ thống, bao gồm thông tin về sách, người dùng, và các lịch sử hoạt động.

Hệ thống này đảm bảo sự tương tác giữa phần cứng và phần mềm một cách hiệu quả, cho phép quá trình tự động hóa hoạt động mượn trả sách diễn ra một cách mượt mà và chính xác. Việc sử dụng Raspberry Pi làm máy chủ đảm bảo khả năng mở rộng và linh hoạt, trong khi các thiết bị IoT như ESP32 giúp tối ưu hóa việc quản lý và giám sát từ xa.

## Thiết kế giao diện trang web

Giao diện của hệ thống quản lý sách đóng vai trò quan trọng trong việc giúp người dùng tương tác dễ dàng và hiệu quả với hệ thống. Để đảm bảo tính trực quan và thân thiện với người dùng, quá trình thiết kế giao diện cần tuân thủ các nguyên tắc cơ bản của thiết kế UI/UX, đồng thời đảm bảo tương thích với các thiết bị sử dụng, từ máy tính bàn cho đến điện thoại di động.



Hình 10. Ý tưởng web site quản lý sách

### Nguyên tắc thiết kế

* **Đơn giản và trực quan:** Giao diện cần được tối ưu hóa để người dùng dễ dàng thao tác mà không cần phải học hỏi nhiều. Các chức năng chính như quản lý sách, mượn trả sách, tìm kiếm sách bằng giọng nói, và quét mã vạch cần được đặt ở các vị trí dễ thấy và dễ truy cập.
* **Thống nhất màu sắc và bố cục:** Sử dụng màu sắc nhã nhặn, hài hòa, đồng thời bố trí các thành phần giao diện một cách khoa học để đảm bảo sự nhất quán và dễ theo dõi.
* **Tương thích nhiều thiết bị:** Giao diện web phải đáp ứng tiêu chuẩn thiết kế responsive, giúp hiển thị tốt trên mọi kích cỡ màn hình, bao gồm máy tính để bàn, laptop, tablet, và điện thoại di động.

### Các thành phần chính của giao diện

* **Trang chủ:** Đây là trang đầu tiên khi người dùng truy cập vào hệ thống. Giao diện trang chủ cần có các thông tin tổng quan về hệ thống như số lượng sách hiện có, số lượng sách đang được mượn, và các thông báo quan trọng.
  + Thanh điều hướng (Navbar) chứa các liên kết đến các trang quan trọng như:
    - Trang quản lý sách
    - Trang quản lý người dùng
    - Trang quản lý mượn trả sách
    - Cài đặt hệ thống
* **Trang quản lý sách:** Cung cấp giao diện để người dùng có thể xem danh sách sách hiện có, thêm mới sách, sửa đổi thông tin sách, hoặc xóa sách. Mỗi cuốn sách sẽ có các trường thông tin như tên sách, tác giả, nhà xuất bản, mã ISBN, và số lượng sách hiện có.
* **Trang quản lý mượn trả sách:** Tích hợp chức năng quét mã vạch thông qua ESP32 để hỗ trợ việc mượn và trả sách. Người dùng có thể quét mã sách, kiểm tra thông tin mượn sách, và cập nhật trạng thái sách ngay trên giao diện này.
* **Trang quản lý người dùng:** Cho phép quản trị viên quản lý danh sách người dùng, bao gồm thêm mới, chỉnh sửa thông tin, và theo dõi lịch sử mượn trả sách của từng người dùng.
* **Trang báo cáo:** Hiển thị các báo cáo liên quan đến hoạt động của thư viện như số lượng sách đã được mượn, số lượng sách bị quá hạn, và tình trạng hiện tại của thư viện.

### Chức năng tìm kiếm sách bằng giọng nói

Chức năng tìm kiếm sách bằng giọng nói giúp người dùng dễ dàng tìm kiếm thông tin sách mà không cần nhập liệu thủ công, đặc biệt hữu ích khi người dùng không thể sử dụng bàn phím hoặc khi tìm kiếm cần diễn ra nhanh chóng.

* **Công nghệ sử dụng:** Chức năng này sử dụng **Web Speech API** để nhận dạng giọng nói của người dùng, sau đó gửi yêu cầu tìm kiếm tới hệ thống Node.js, nơi cơ sở dữ liệu sẽ được truy vấn để tìm sách phù hợp.
* **Thiết kế giao diện:** Trên giao diện trang chủ hoặc trang quản lý sách, sẽ có một biểu tượng micro. Khi người dùng nhấn vào biểu tượng này, hệ thống sẽ bắt đầu lắng nghe giọng nói. Kết quả tìm kiếm sẽ hiển thị sách tương ứng với từ khóa mà người dùng vừa nói.

Ví dụ: Người dùng nói "Harry Potter", hệ thống sẽ trả về danh sách các cuốn sách có tiêu đề chứa từ "Harry Potter".

* **Lưu ý:** Để nâng cao trải nghiệm người dùng, cần cung cấp phản hồi trực quan khi hệ thống đang lắng nghe (ví dụ: thay đổi màu sắc biểu tượng micro hoặc hiển thị thông báo “Đang lắng nghe…”).

### Công cụ và ngôn ngữ sử dụng

* **HTML5 và CSS3:** Sử dụng để xây dựng cấu trúc và giao diện người dùng của trang web. CSS3 sẽ giúp tối ưu hóa khả năng responsive của trang web, đảm bảo hiển thị tốt trên các thiết bị có độ phân giải khác nhau.
* **JavaScript (Express.js hoặc Vue.js):** Để xử lý các tương tác của người dùng trên giao diện, đảm bảo tính động và phản hồi nhanh chóng. Có thể sử dụng các thư viện frontend hiện đại như React.js hoặc Vue.js để xây dựng các component UI. Chức năng tìm kiếm giọng nói sẽ sử dụng JavaScript để kết nối với Web Speech API.
* **Bootstrap:** Một framework CSS phổ biến, giúp xây dựng giao diện responsive nhanh chóng và dễ dàng.
* **Node.js và Express.js:** Sử dụng làm backend để xử lý các yêu cầu từ giao diện người dùng, quản lý cơ sở dữ liệu, và tích hợp với Raspberry Pi để kết nối với các thiết bị IoT như ESP32.

### Giao diện mẫu (Wireframe)

* **Trang chủ:** Một bố cục đơn giản với thanh điều hướng ở đầu trang, khu vực thông báo chính giữa, và các số liệu thống kê nằm phía dưới. Biểu tượng micro để tìm kiếm giọng nói sẽ được đặt ở vị trí dễ thấy trên trang chủ.
* **Trang quản lý sách:** Bảng hiển thị danh sách các sách, có các nút chức năng thêm mới, sửa và xóa sách. Ngoài ra, mỗi sách sẽ có một nút chi tiết để hiển thị thông tin cụ thể của cuốn sách đó. Tìm kiếm giọng nói cũng được tích hợp tại đây.

# KẾT QUẢ VÀ PHÂN TÍCH

## Phương pháp tiếp cận

Trong phần này, cần nêu ra phương pháp tiếp cận dựa vào cơ sở lý thuyết đã nêu trong chương 2. Đối với mô phỏng, cần đưa ra và giải thích lưu đồ giải thuật. Đối với phần cứng, cần đưa ra sơ đồ khối hệ thống.

Các điều kiện đầu của mô phỏng cần được ghi chú đầy đủ. Ví dụ: nếu mô hình toán sử dụng để mô phỏng có liên quan đến biến A và B, cần đưa ra giá trị mô phỏng của A và B kèm giải thích tại sao lại chọn giá trị trên.

Với đo đạc thực nghiệm, kết quả đo đạc chỉ tin cậy nếu có khả năng lặp lại được thí nghiệm. Do đo đạc luôn có sai số, các thông số máy đo, điều kiện đo phải được ghi chú và giải thích cụ thể.

## Kết quả và phân tích

### Khảo sát thông số A (ghi chú: các mục này nên ghi theo câu hỏi nghiên cứu)

Đưa các hình vẽ, bảng biểu kết quả tương tự như đã hướng dẫn ở trên. Lưu ý hình vẽ và bảng biểu bắt buộc phải đi kèm phân tích. Các yếu tố có thể phân tích là ý nghĩa các cực trị, nguyên nhân tăng giảm, nguyên nhân khác biệt giữa các kết quả.

### Kết quả mô phỏng thông số B

Tương tự như trên cho các vấn đề nghiên cứu đã đặt ra.

## Kết luận chương

Kết luận ngắn chương 3. Tóm tắt lại các ý phân tích và trả lời các câu hỏi nghiên cứu.

# KẾT LUẬN

## Tóm tắt và kết luận chung

Trong phần kết luận, trả lời các câu hỏi nghiên cứu đã đặt ra ở chương 1 dựa vào kết quả thực hiện va phân tích trong chương 3. Nêu các điểm đạt, chưa đạt của luận văn. Nguyên nhân đạt, nguyên nhân chưa đạt và cách khắc phục nếu có.

## Hướng phát triển

Phần hướng phát triển nêu ngắn gọn, không tràn lan.

# PHỤ LỤC A

(Ghi chú: Code chương trình có thể để vào phụ lục A và các chứng minh toán học hỗ trợ có thể để vào phụ lục B)

## A.2 Code chương trình xử lý dữ liệu dùng Matlab

(Copy code vào đây)

# TÀI LIỆU THAM KHẢO

1. B. Klaus and P. Horn, *Robot Vision*. Cambridge, MA, USA: MIT Press, 1986. (ví dụ cách viết tài liệu tham khảo là sách)
2. A. Amador-Perez and R. A. Rodriguez-Solis, “Analysis of a CPW-fed annular slot ring antenna using DOE,” in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Jul. 2006, pp. 4301–4304. (ví dụ cách viết tài liệu tham khảo là bài báo hội nghị khoa học)
3. M. M. Chiampi and L. L. Zilberti, “Induction of electric field in human bodies moving near MRI: An efficient BEM computational procedure,” *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 2787–2793, Oct. 2011, doi: 10.1109/TBME.2011.2158315. (ví dụ cách viết tài liệu tham khảo là bài báo tạp chí khoa học)