**BỘ GIÁO DỤC VÀ ĐÀO TẠO   BỘ NÔNG NGHIỆP VÀ PTNT**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

****

PHÙNG KÍ TÀI

**XÂY DỰNG WEBSITE VÀ TÍCH HỢP VỚI CHATBOT PHỤC VỤ TRUYỀN THÔNG VÀ GIỚI THIỆU ĐỊA ĐIỂM DI TÍCH LỊCH SỬ CỦA THỦ ĐỒ HÀ NỘI**

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2024

**BỘ GIÁO DỤC VÀ ĐÀO TẠO   BỘ NÔNG NGHIỆP VÀ PTNT**

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

PHÙNG KÍ TÀI

**XÂY DỰNG WEBSITE VÀ TÍCH HỢP VỚI CHATBOT PHỤC VỤ TRUYỀN THÔNG VÀ GIỚI THIỆU ĐỊA ĐIỂM DI TÍCH LỊCH SỬ CỦA THỦ ĐỒ HÀ NỘI**

|  |  |
| --- | --- |
| Ngành: | Công nghệ thông tin |
| Mã số: | 7480201 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | **Ths. Vũ Thành Vinh** |

HÀ NỘI, NĂM 2024

|  |  |
| --- | --- |
|  | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  --------------------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

|  |  |
| --- | --- |
| Họ tên sinh viên: **Phùng Kí Tài** | Hệ đào tạo: Đại học chính quy |
| Lớp: **62TH4** | Khoa: Công nghệ thông tin |
| Mã sinh viên: **2051063445** | Ngành: Công nghệ thông tin |

1. **TÊN ĐỀ TÀI**

XÂY DỰNG WEBSITE VÀ TÍCH HỢP VỚI CHATBOT PHỤC VỤ TRUYỀN THÔNG VÀ GIỚI THIỆU ĐỊA ĐIỂM DI TÍCH LỊCH SỬ CỦA THỦ ĐỒ HÀ NỘI

1. **CÁC TÀI LIỆU CƠ BẢN**

|  |  |
| --- | --- |
| [1] | *React*. (n.d.).<https://react.dev/> |
| [2] | *Index | Node.js v21.7.3 Documentation*. (n.d.). https://nodejs.org/docs/latest/api/ |
| [3] | *Frameworks built on Express*. (n.d.). https://expressjs.com/en/resources/frameworks.html |
| [4] | *MongoDB documentation*. (n.d.). MongoDB Documentation.https://www.mongodb.com/docs/ |
| [5] | *Cultural heritage: 7 successes of UNESCO’s preservation work*. (2023, July 18). UNESCO. https://www.unesco.org/en/cultural-heritage-7-successes-unescos-preservation-work |
| [6] | *Tourism at World Heritage Sites – Challenges and Opportunities: International tourism seminar, Çeşme (Izmir), Turkey, 26 March 2013*. (2015).<https://doi.org/10.18111/9789284416608> |
| [7] | *Cultural heritage: 7 successes of UNESCO’s preservation work*. (2023b, July 18). UNESCO. https://www.unesco.org/en/cultural-heritage-7-successes-unescos-preservation-work |
| [8] | Phan, D. (2022, December 30). *The best historical sites to visit in Vietnam*. Culture Trip.<https://theculturetrip.com/asia/vietnam/articles/the-10-best-historical-sites-to-visit-in-vietnam> |
| [9] | *OpenAI Platform*. (n.d.). https://platform.openai.com/docs/introduction |

1. **NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN**

|  |
| --- |
| **NỘI DUNG** |
| Chương 1: Cơ sở lý thuyết |
| Chương 2: Thiết kế và xây dựng hệ thống |
| Chương 3: Kết quả và đánh giá kết luận |

1. **BẢNG BIỂU, SƠ ĐỒ, HÌNH**
2. **NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày ............ tháng ......... năm 2023

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua

Ngày. . . . .tháng. . . . .năm 2024

**Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày... tháng... năm 2024

**Sinh viên làm Đồ án tốt nghiệp**

***(KÝ VÀ GHI RÕ HỌ TÊN)***

|  |  |
| --- | --- |
|  | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  --------------------  **BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP** |

**Tên đề tài: Xây dựng website và tích hợp với Chatbot phục vụ truyền thông và giới thiệu địa điểm di tích lịch sử của thủ đô Hà Nội**

*Sinh viên thực hiện*: Phùng Kí Tài

*Lớp*: 62TH4

*Mã sinh viên:* 2051063445

*Số điện thoại: 0346909328*

*Email:* 2051063445@e.tlu.edu.vn

*Giáo viên hướng dẫn*: Ths. Vũ Thành Vinh

**TÓM TẮT ĐỀ TÀI**

Hà Nội, thủ đô ngàn năm văn hiến của Việt Nam, là nơi chứa đựng những di sản văn hóa và lịch sử phong phú. Từ Lăng Chủ tịch Hồ Chí Minh, biểu tượng của lòng yêu nước và tự hào dân tộc, đến Nhà tù Hỏa Lò, nơi ghi dấu những trang sử đau thương nhưng cũng đầy kiêu hãnh, mỗi di tích ở Hà Nội đều là một chứng nhân sống động của lịch sử.

Tuy nhiên, việc tiếp cận và tìm hiểu sâu về những di tích này không phải lúc nào cũng thuận lợi cho du khách. Một số khó khăn mà du khách thường gặp phải bao gồm thông tin không đầy đủ hoặc không cập nhật trên internet, rào cản ngôn ngữ, và thiếu các tour du lịch được tổ chức chuyên nghiệp. Đặc biệt, trong bối cảnh du lịch đang phục hồi sau đại dịch, Hà Nội đang đối mặt với cả cơ hội và thách thức trong việc quản lý lượng khách du lịch tăng cao, đồng thời nâng cao chất lượng dịch vụ và trải nghiệm cho du khách.

Trên khắp thế giới, việc bảo tồn và tiếp cận các di tích lịch sử đang đối mặt với nhiều thách thức. Từ việc bảo vệ di sản văn hóa trước những tác động của biến đổi khí hậu, đến việc duy trì nguyên trạng của các địa danh lịch sử trong bối cảnh đô thị hóa và phát triển kinh tế. Các di sản này không chỉ là những bảo tàng sống động của quá khứ mà còn là nguồn cảm hứng cho sự sáng tạo và đổi mới.

Ở Việt Nam, sự phát triển nhanh chóng của đô thị đã tạo ra những áp lực lớn đối với việc bảo tồn các di tích lịch sử. Ví dụ, Khu di tích Cố đô Huế đang phải đối mặt với những thách thức từ sự mở rộng đô thị, ảnh hưởng đến cảnh quan và giá trị văn hóa của di sản. Bên cạnh đó, việc tiếp cận thông tin về các di tích cho du khách cũng gặp khó khăn do sự thiếu hụt thông tin đầy đủ và chính xác trên các phương tiện truyền thông.

Chính vì những thách thức này, việc xây dựng một website giới thiệu các di tích lịch sử ở Hà Nội bằng công nghệ hiện đại như React, ExpressJS, MongoDB, và NodeJS trở nên hết sức cần thiết. Website này không chỉ giúp cung cấp thông tin một cách đầy đủ và dễ dàng tiếp cận hơn cho du khách, mà còn góp phần quảng bá hình ảnh và giá trị văn hóa của Việt Nam ra thế giới. Đây sẽ là cầu nối giữa di sản văn hóa và công nghệ, giúp du khách có thể tìm hiểu sâu rộng và trải nghiệm phong phú khi khám phá Hà Nội, ngay cả trước khi họ đặt chân đến nơi đây.

Đề tài sẽ tập trung thực hiện các nội dung chính:

* Sử dụng React để xây dựng giao diện người dùng (UI) cho trang web với các thành phần tái sử dụng để tạo ra một trải nghiệm người dùng linh hoạt và thân thiện.
* Sử dụng ExpressJS để xây dựng một RESTful API cho việc tương tác với cơ sở dữ liệu và cung cấp dữ liệu cho phía frontend.
* Sử dụng NodeJS để xử lý các yêu cầu từ phía client và tương tác với cơ sở dữ liệu để truy xuất và cập nhật thông tin về các di tích lịch sử.
* Sử dụng một cơ sở dữ liệu như MongoDB hoặc MySQL để lưu trữ thông tin về các di tích lịch sử, bao gồm các trường như tên, mô tả, vị trí địa lý, hình ảnh, v.v.
* Tích hợp AI Chatbot để hỗ trợ du khách.

**CÁC MỤC TIÊU CHÍNH**

Xây dựng trang web hoàn chỉnh: Phát triển một trang web đầy đủ chức năng và chất lượng cao để giới thiệu các di tích lịch sử ở Hà Nội.

Sử dụng công nghệ hiện đại: Áp dụng các công nghệ mới nhất như MongoDB,  React, ExpressJS và NodeJS để xây dựng ứng dụng web một cách hiệu quả và linh hoạt.

Tích hợp các tính năng tương tác: Tạo ra các tính năng tương tác như tìm kiếm, lọc, đánh giá và bình luận để người dùng có thể tương tác với trang web một cách thuận tiện và sinh động.

**KẾT QUẢ DỰ KIẾN**

1. Lý thuyết:

Hiểu biết sâu sắc về quy trình phát triển phần mềm và kiến thức vững về các công nghệ được sử dụng trong dự án, bao gồm MongoDB, React, ExpressJS và NodeJS.

Thành thạo trong việc triển khai các nguyên lý thiết kế web và các kỹ thuật lập trình để tạo ra các trang web đáng tin cậy, hiệu quả và dễ bảo trì.

Quyển báo cáo đồ án tốt nghiệp

1. Thực hiện:

Xây dựng và triển khai trang web hoàn thiện và hoạt động một cách mượt mà và hiệu quả trên mọi thiết bị và mọi môi trường truy cập và tạo ra một trải nghiệm tương tác và thú vị cho người dùng.

Tích hợp với AI Chatbot để để hỗ trợ du khách trong việc tìm hiểu về địa điểm du lịch của thủ đô Hà Nội.

**TÀI LIỆU THAM KHẢO**

| [1] | *React*. (n.d.).<https://react.dev/> |
| --- | --- |
| [2] | *Index | Node.js v21.7.3 Documentation*. (n.d.). https://nodejs.org/docs/latest/api/ |
| [3] | *Frameworks built on Express*. (n.d.). https://expressjs.com/en/resources/frameworks.html |
| [4] | *MongoDB documentation*. (n.d.). MongoDB Documentation.https://www.mongodb.com/docs/ |
| [5] | *Cultural heritage: 7 successes of UNESCO’s preservation work*. (2023, July 18). UNESCO. https://www.unesco.org/en/cultural-heritage-7-successes-unescos-preservation-work |
| [6] | *Tourism at World Heritage Sites – Challenges and Opportunities: International tourism seminar, Çeşme (Izmir), Turkey, 26 March 2013*. (2015).<https://doi.org/10.18111/9789284416608> |
| [7] | *Cultural heritage: 7 successes of UNESCO’s preservation work*. (2023b, July 18). UNESCO. https://www.unesco.org/en/cultural-heritage-7-successes-unescos-preservation-work |
| [8] | Phan, D. (2022, December 30). *The best historical sites to visit in Vietnam*. Culture Trip.<https://theculturetrip.com/asia/vietnam/articles/the-10-best-historical-sites-to-visit-in-vietnam> |
| [9] | *OpenAI Platform*. (n.d.). https://platform.openai.com/docs/introduction |

**TIẾN ĐỘ THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Thời gian** | **Nội dung công việc** | **Kết quả dự đoán đạt được** |
| 1 | 03/05/2024-05/05/2024 | Tìm hiểu đề tài: “Xây dựng website và tích hợp với Chatbot phục vụ truyền thông và giới thiệu địa điểm di tích lịch sử của thủ đô Hà Nội”.  Lên kế hoạch làm việc. | Phân tích các vấn đề chính của đề tài và hướng giải quyết. Lên kế hoạch cho từng công việc phải làm để đảm bảo đồ án thực hiện đúng tiến độ. |
| 2 | 06/05/2024-12/05/2024 | - Setting up routes  - Create user auth form  - Google authentication | - Thiết lập đường dẫn cho biểu mẫu xác thực người dùng  - Tạo biểu mẫu xác thực người dùng  - Cho phép người dùng đăng nhập vào ứng dụng bằng tài khoản Google  - Viết API cho chức năng Sign-in, Sign-up |
| 3 | 13/05/2024-19/05/2024 | - Create editor Page  - Create blog editor, navbar,  banner  - Create blog card, user card  - Create search Input  - Create pagination | - Tạo trang cho người dùng đăng tải nội dung về di tích lịch sử ở Hà Nội  - Xây dựng các thành phần giao diện như trình soạn thảo blog, thanh điều hướng, phân trang và banner đáp ứng yêu cầu thiết kế  - Tạo ra các thành phần giao diện để hiển thị thông tin về bài đăng blog và người dùng một cách rõ ràng và đẹp mắt  - Phát triển một ô nhập liệu cho phép người dùng nhập thông tin tìm kiếm và hiển thị kết quả tìm kiếm phù hợp  - Viết API cho chức năng soạn thảo blog, tìm kiếm blog và user |
| 4 | 20/05/2024-26/05/2024 | - Create user profile page  - Create content detail for blog page | - Tạo ra một trang hồ sơ cá nhân cho người dùng hiển thị thông tin cá nhân và hoạt động liên quan  - Tạo ra nội dung chi tiết và đầy đủ để hiển thị trên trang blog  - Viết API cho chức năng hồ sơ cá nhân |
| 5 | 27/05/2024-02/06/2024 | - Create edit blog  - Create comments cards | - Xây dựng chức năng cho phép người dùng chỉnh sửa các bài đăng trên trang blog một cách dễ dàng và hiệu quả  - Tạo ra các thẻ hiển thị bình luận cho các bài đăng  - Viết API cho chức năng chỉnh sửa blog |
| 6 | 03/06/2024-09/06/2024 | - Create like, comment post  - Create reply comment  - Hiding & Loading replies | - Xây dựng chức năng cho phép người dùng thích và bình luận các bài đăng  - Cho phép người dùng trả lời vào các bình luận trên bài đăng  - Chức năng cho phép người dùng ẩn và tải các câu trả lời của bình luận  - Viết  API cho các chức năng thích và bình luận |
| 7 | 10/06/2024-16/06/2024 | - Delete comments  - Create edit profile page  - Fix: User Profile Page  - Change password form | - Xây dựng chức năng cho phép người dùng xóa các bình luận mà họ đã đăng  - Xây dựng một giao diện cho phép người dùng chỉnh sửa thông tin cá nhân của họ  - Sửa các lỗi và cải thiện trang hồ sơ người dùng để đảm bảo hoạt động mượt mà  - Xây dựng một biểu mẫu cho phép người dùng thay đổi mật khẩu đăng nhập của họ  - Viết API cho chức năng xóa comments, edit profile và đổi mật khẩu - Sửa API cho hồ sơ cá nhân |
| 8 | 17/06/2024-23/06/2024 |  |  |
| 9 | 24/06/2024-30/06/2024 |  |  |
| 10 | 01/07/2024-07/07/2024 |  |  |
| 11 | 08/07/2024-14/07/2024 |  |  |

**LỜI CAM ĐOAN**

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao từ bất kì một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy trình.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  ***Chữ ký*** |

**LỜI CẢM ƠN**

Sau 4 năm học tập và rèn luyện tại Khoa Công nghệ thông tin – Trường Đại học Thuỷ Lợi, em đã được rèn luyện trong môi trường học tập chất lượng và nghiêm túc, em đã được sự chỉ dạy nhiệt tình và hướng dẫn chu đáo của các thầy cô trong Khoa cũng như các thầy cô trong trường.

Trước hết, em xin được gửi lời cảm ơn sâu sắc đến Ths. Vũ Thành Vinh đã quan tâm và nhiệt tình chỉ bảo, nhắc nhở và hướng dẫn em trong suốt thời gian thực hiện đồ án tốt nghiệp.

Em cũng xin chân thành bày tỏ lòng cảm ơn đối với các thầy cô trong Khoa Công nghệ thông tin nói riêng và các thầy cô Trường Đại học Thuỷ Lợi nói chung đã trang bị cho em những tri thức quý giá làm hành trang giúp đỡ em trong những năm làm việc sắp tới.

Trong quá trình thực hiện đồ án tốt nghiệp, tuy được sự chỉ bảo tận tình, tâm huyết của Ths. Vũ Thành Vinh cộng với kinh nghiệm của bản thân nhưng đồ án cũng như đồ án không thể nào tránh khỏi những sai sót. Em rất mong nhận được sự thông cảm cùng những đóng góp từ Thầy Cô và các bạn để đề tài được hoàn chỉnh và tiến bộ hơn.

**Em xin chân thành cảm ơn!**

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

**DANH MỤC BẢNG BIỂU**

**DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ**

**LỜI MỞ ĐẦU**

1. **Giới thiệu đề tài**

Trên thị trường hiện nay, với sự phát triển nhanh chóng của công nghệ thông tin, việc sử dụng trang web và chatbot đang trở thành một xu hướng quan trọng trong lĩnh vực truyền thông và quảng cáo du lịch. Đặc biệt, khi khám phá các di tích lịch sử, việc kết hợp cả hai yếu tố này không chỉ giúp tăng cường hiệu quả quảng cáo mà còn mang lại trải nghiệm tương tác tốt hơn cho khách hàng.

Việc xây dựng một nền tảng kết hợp trang web và chatbot cho du lịch không chỉ giúp các doanh nghiệp thu hút khách du lịch mà còn tạo ra một cách tiếp cận hiện đại và thuận tiện cho người dùng. Nhờ vào tính linh hoạt của chatbot, khách hàng có thể nhận thông tin ngay lập tức và hỗ trợ khi cần mà không cần phải chờ đợi.

Với sự phổ biến của điện thoại thông minh và internet, việc sử dụng trang web và chatbot trong lĩnh vực du lịch giúp tạo ra một kênh giao tiếp trực tuyến hiệu quả, giúp doanh nghiệp tiếp cận đến đông đảo khách hàng tiềm năng một cách nhanh chóng và tiện lợi. Đồng thời, việc áp dụng công nghệ này cũng giúp nâng cao trải nghiệm của khách hàng khi tìm hiểu thông tin về điểm đến du lịch và dễ dàng tương tác để có được những thông tin cần thiết.

1. **Lý do chọn đề tài**

Hiện nay, em nhận thấy các trang web thường không đáp ứng đủ nhu cầu của người dùng như việc chia sẻ cảm xúc và bình luận để trao đổi với những người khác về di sản lịch sử Hà Nội mà họ quan tâm. Điều này gây khó khăn khi họ cần thông tin chi tiết từ các bài viết mà không tìm thấy câu trả lời chính xác.

Vấn đề này đã truyền cảm hứng cho em từ sự yêu thích và quan tâm đặc biệt đến việc bảo tồn và phát huy giá trị của di sản văn hóa lịch sử của Hà Nội.

Việc phát triển một trang web kết hợp với chatbot sẽ không chỉ giúp chia sẻ thông tin một cách chính xác và dễ dàng về các địa điểm di tích, mà còn thúc đẩy sự tương tác giữa người dùng và văn hóa lịch sử sâu sắc của dân tộc. Điều này mở ra cơ hội để khám phá và truyền đạt những giá trị văn hóa lịch sử một cách sinh động và hiệu quả nhất.

Sử dụng công nghệ trong việc bảo tồn di sản văn hóa không chỉ giúp duy trì mà còn phát triển các giá trị truyền thống theo cách hiện đại và hấp dẫn hơn đối với giới trẻ.

Kết hợp giữa truyền thống và công nghệ không chỉ là một cách tiếp cận mới mà còn là yếu tố quan trọng trong việc thúc đẩy sự phát triển toàn diện cho cộng đồng, từ việc bảo tồn di sản đến việc tăng cường tinh thần quốc gia và nhận thức văn hóa trong cả nước. Đây là một cách tiếp cận sáng tạo và mang lại nhiều lợi ích vượt ra ngoài phạm vi dự án đơn lẻ, hướng tới một cộng đồng phát triển và sáng tạo hơn.

1. **Mục tiêu của đề tài**

Xây dựng trang web hoàn chỉnh: Phát triển một trang web đầy đủ chức năng và chất lượng cao để giới thiệu các di tích lịch sử ở Hà Nội.

Sử dụng công nghệ hiện đại: Áp dụng kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) để xây dựng ứng dụng web một cách hiệu quả và linh hoạt.

Tích hợp các tính năng tương tác: Tạo ra các tính năng tương tác như tìm kiếm, lọc, đánh giá và bình luận để người dùng có thể tương tác với trang web một cách thuận tiện và sinh động.

Mục đích cuối cùng là tăng cường hiểu biết về văn hóa lịch sử, quảng bá vẻ đẹp và giá trị du lịch của Hà Nội cũng như thúc đẩy sự phát triển của công nghệ và du lịch.

1. **Phương pháp nghiên cứu**

**4.1. Phân tích yêu cầu và thiết kế hệ thống:**

Xác định và phân tích yêu cầu cụ thể cho website và chatbot nhằm mục đích truyền thông và giới thiệu các địa điểm di tích lịch sử của Hà Nội.

Thiết kế kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) để đáp ứng các yêu cầu và tích hợp chatbot một cách hiệu quả.

**4.2. Phát triển website:**

Sử dụng ReactJS để phát triển giao diện người dùng thân thiện và dễ sử dụng.

Xây dựng backend bằng NodeJS và ExpressJS để quản lý logic nghiệp vụ và tương tác với cơ sở dữ liệu MongoDB.

**4.3. Phát triển chatbot:**

Tích hợp chatbot vào website bằng cách sử dụng Chat Coze Web SDK

Lập trình chatbot để giới thiệu và hỗ trợ người dùng tra cứu thông tin về các địa điểm di tích lịch sử của Hà Nội một cách nhanh chóng và hiệu quả.

**4.4. Kiểm thử và đánh giá:**

Tiến hành kiểm tra, đánh giá để đảm bảo tính ổn định, bảo mật và hiệu suất của website và chatbot.

Thu thập phản hồi từ người dùng để cải thiện và điều chỉnh sản phẩm.

1. **Đối tượng phạm vi nghiên cứu**

**Người dùng cuối:** Các khách tham quan và du khách quan tâm đến các di tích lịch sử của Hà Nội.

**Người quản trị hệ thống:** Các quản trị viên có nhiệm vụ duy trì và quản lý website và chatbot.

**Nhà phát triển:** Những cá nhân tham gia vào việc phát triển và cải tiến hệ thống.

**Người quản lý dự án:** Những người có trách nhiệm điều hành và giám sát tiến độ của dự án.

1. **Nhiệm vụ và kết quả đạt được**

Nghiên cứu và tìm hiểu một số trang web, phần mềm quảng bá di tích, địa điểm tham quan

Phân tích yêu cầu và đề xuất của phần mềm

Thiết kế đặc tả hệ thống

Xây dựng được cơ sở dữ liệu

Lập trình các module của hệ thống

1. **Bố cục của đồ án**

Trên cơ sở các nội dung nghiên cứu, để đạt mục tiêu đề ra và đảm bảo tính logic, ngoài phần *mở đầu* và *phần kết luận*, đồ án được tổ chức thành các chương như sau:

*Chương 1. Cơ sở lý thuyết*. Chương này trình bày các cơ sở lý thuyết về internet và website; trình bày kết quả nghiên cứu, tìm hiểu về các ngôn ngữ lập trình và kiến trúc hệ thống MERN (MongoDB, ExpressJS, ReactJS và NodeJS) được sử dụng trong việc xây dựng đề tài.

*Chương 2: Phân tích và thiết kế hệ thống*. Chương này giới thiệu bài toán và đưa ra các mô hình hệ thống; đề xuất các ý tưởng, chức năng cho phần mềm; trình bày các mô hình thực thể quan hệ, sơ đồ phân cấp chức năng. Phân tích chi tiết các chức năng của người sử dụng và người quản lý; thiết kế giao diện người dùng và thiết kế dữ liệu, chức năng cho hệ thống.

*Chương 3: Cài đặt chương trình và kết quả.* Trong chương này, nêu kết quả thực hiện được của đề tài và trình bày cách cài đặt chương trình.

1. **Đóng góp của đồ án**

Trong dự án tốt nghiệp này, em đã sử dụng những kiến thức học được từ trường để thực hiện đề tài một cách linh hoạt. Không chỉ vậy, em còn tự tìm hiểu, nghiên cứu thêm về các kiến thức mới và ngôn ngữ lập trình mới để phát triển dự án một cách nhanh chóng và hoàn thiện hơn:

*Phân tích rõ ràng vai trò của người dùng trong hệ thống*. Đối với người quản *trị*, khách hàng đã đăng nhập hay chưa đăng nhập đều được giới hạn một số chức năng nhất định.

*Đưa ra các chức năng điều khiển dành cho người quản trị*. Nghiên cứu và triển khai thực nghiệm các chức năng nhằm đáp ứng nhu cầu cho người quản lý như quản lý các thông tin về bài viết, thông tin các người dùng.

*Đưa ra chức năng thả cảm xúc và bình luận cho người dùng*. Nghiên cứu và triển khai thực nghiệm chức năng thả cảm xúc và bình luận cho người dùng. Chức năng này cho phép người dùng thể hiện cảm xúc và ý kiến cá nhân về các địa danh, cũng như trao đổi với người dùng khác về những địa điểm họ quan tâm.

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **Khái niệm cơ bản về Internet**

**1.1. Giới thiệu chung**

Internet hay Mạng là một hệ thống thông tin toàn cầu có thể được truy nhập công cộng gồm các mạng máy tính được liên kết với nhau. Hệ thống này truyền thông tin theo kiểu nối chuyển gói dữ liệu (packet switching) dựa trên một giao thức liên mạng đã được chuẩn hóa (giao thức IP).

Hệ thống này bao gồm hàng ngàn mạng máy tính nhỏ hơn của các doanh nghiệp, của các viện nghiên cứu và các trường đại học, của người dùng cá nhân và các chính phủ trên toàn cầu, được liên kết bởi một loạt các công nghệ mạng điện tử, không dây và mạng quang.

Internet mang theo một loạt các tài nguyên và dịch vụ thông tin, chẳng hạn như các tài liệu và ứng dụng siêu văn bản được liên kết với nhau của World Wide Web (WWW), thư điện tử, điện thoại và chia sẻ file.

**1.2. Phân loại**

**Các máy tính trên mạng được chia làm 2 nhóm Client và Server.**

Client: Máy khách trên mạng, chứa chương trình Client.

Server: Máy phục vụ-máy chủ. Chứa các chương trình Server, tài nguyên (tập tin, tài liệu, …) dùng chung cho nhiều máy khách. Server luôn ở trạng thái chờ yêu cầu và đáp ứng yêu cầu của Client.

**Các dịch vụ thường dùng trong Internet.**

* Dịch vụ World Wide Web (gọi tắt Web)
* Dịch vụ Electronic Mail (gọi tắt là Mail)
* Dịch vụ FPT

Để truyền thông tin với những máy tính khác, mọi máy tính trên Internet phải hỗ trợ giao thức chung TCP/IP (Transmission Control Protocol/Internet Protocol - giao thức điều khiển truyền dữ liệu và giao thức Internet), là một giao thức đồng bộ, cho truyền thông điệp từ nhiều nguồn và tới nhiều đích khác nhau. Ví dụ có thể vừa lấy thư từ trong hộp thư, đồng thời vừa truy cập vào trang web. TCP đảm bảo tính an toàn dữ liệu, IP là giao thức chi phối phương thức vận chuyển dữ liệu trên Internet.

**1.3. World Wide Web**

**Các khái niệm cơ bản về World Wide Web**

World Wide Web (WWW) hay còn gọi là Web là một dịch vụ phổ biến nhất hiện nay trên Internet, 85% các giao dịch trên Internet ước lượng thuộc về WWW. Ngày nay số Website trên thế giới đã đạt tới con số khổng lồ. WWW cho bạn truy xuất thông tin văn bản, hình ảnh, âm thanh, video trên toàn thế giới. Thông qua Website, các quý công ty có thể giảm thiểu tối đa chi phí in ấn và phân phát tài liệu cho khách hàng ở nhiều nơi.

**Cách tạo trang web**

*Có nhiều cách để tạo trang Web:*

Có thể tạo trang web trên bất kỳ chương trình xử lý văn bản nào.

Tạo web bằng cách viết mã nguồn bởi một trình soạn thảo văn bản như:

* Notepad, Wordpad, v.v. là những chương trình soạn thảo có sẵn trong window.
* Thiết kế bằng các phần mềm chuyên nghiệp như: Frontpage, Dreamweaver, visual studio…

Để xây dựng một ứng dụng Web hoàn chỉnh và có tính thương mại, bạn cần kết hợp cả Client Script (Kịch bản trình khách) và Server Script (Kịch bản trình chủ) với một loại cơ sở dữ liệu nào đó, chẳng hạn như: MS Access, SQL Server, Oracle, ở đây em dùng MongoDB. Khi bạn muốn triển khai ứng dụng Web trên mạng, ngoài các điều kiện về cấu hình phần cứng, bạn cần có máy chủ Web thường gọi là Web Server.

**Trình duyệt web (Web Client hay Web Browser)**

Trình duyệt web là công cụ truy xuất dữ liệu trên mạng, là phần mềm giao diện trực tiếp với người sử dụng. Nhiệm vụ của web browser là nhận dữ liệu từ người dùng, gửi các yêu cầu đó qua mạng tới các Web Server và nhận các dữ liệu cần thiết từ Server để hiển thị lên màn hình. Để sử dụng dịch vụ WWW,  Client cần có một chương trình duyệt Web, kết nối vào Internet thông qua một ISP. Các trình duyệt thông dụng hiện nay là Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, …

**WebServer**

WebServer: Là một máy tính được nối vào Internet và chạy các phần mềm được thiết kế. WebServer đóng vai trò một chương trình xử lý các nhiệm vụ xác  định, như tìm trang thích hợp, xử lý tổ hợp dữ liệu, kiểm tra dữ liệu hợp lệ, … WebServer cũng là nơi lưu trữ cơ sở dữ liệu, là phần mềm đảm nhiệm vai trò Server cung cấp dịch vụ Web.

**WebServer hỗ trợ các công nghệ khác nhau:**

* IIS (Internet Information Server): Hỗ trợ ASP, mở rộng hỗ trợ PHP
* Apache: Hỗ trợ PHP
* Tomcat: Hỗ trợ JSP (Java Servlet Page)

**Phân loại Website**

* Website tĩnh
* Tài liệu được phân phát rất đơn giản từ hệ thống file Server..
* Định dạng các trang web tĩnh là các siêu liên kết, các trang định dạng Text, các hình ảnh đơn giản.

**Ưu điểm:** CSDL nhỏ nên việc phân phát dữ liệu có hiệu quả rõ ràng, Server có thể đáp ứng nhu cầu Client một cách nhanh chóng. Ta nên sử dụng Web tĩnh khi không thay đổi thông tin trên đó.

**Nhược điểm:** Không đáp ứng được yêu cầu phức tạp của người sử dụng, không linh hoạt.

Hoạt động của Web tĩnh được thể hiện như sau:

|  |
| --- |
| A black arrow pointing to the right  Description automatically generated |
| *Hình 1: Cách hoạt động của trang web tĩnh* |

* Website động

Website động là tập hợp của những trang web mà có nội dung có khả năng thay đổi. Sự thay đổi có thể là tùy theo thời gian, tùy theo người dùng, tùy theo ngữ  cảnh. Nội dung của website động được lấy từ cơ sở dữ liệu hoặc hệ thống quản  lý nội dung (CMS). Do đó, khi bạn cập nhật nội dung của cơ sở dữ liệu thì nội dung của trang web cũng được cập nhật theo.

Để tạo được website động, chúng ta thường sử dụng đến các ngôn ngữ phía server (server-side), chẳng hạn như Servlet, JSP, PHP, Python, C#...

Một  website động  thường được  đặt trên một máy chủ dịch vụ web (WebServer). Website động sử dụng kịch bản lập trình phía máy khách (Client)  hoặc máy chủ (Server) hoặc cả hai để tạo ra nội dung động.

Khi người dùng sử dụng máy tính của mình thao tác trên trang web sẽ tạo ra một yêu cầu (request) tới máy chủ. Phía máy chủ nhận và xử lý yêu cầu máy  khách dựa vào kịch bản trên máy chủ để tạo ra một trang đơn giản và gửi lại tới người dùng. Trình duyệt trên máy khách tải xuống và xử lý mã trong trang để  hiển thị thông tin cho người dùng.

Hoạt động của Web động được thể hiện như sau:

|  |
| --- |
|  |
| *Hình 2: Cách hoạt động của trang web động* |

**1.4. Ngôn ngữ sử dụng**

**1.4.1. Javascript là gì?**

Javascript chính là một ngôn ngữ lập trình với khả năng đem tới sự sinh động  khi thiết kế website. Đây là dạng ngôn ngữ theo kịch bản, dựa trên chính đối  tượng phát triển có sẵn, hoặc là tự định nghĩa ra. Chính vì tính tiện lợi, hiệu quả  ứng dụng cao mà ngôn ngữ lập trình này được tin dùng ngày càng nhiều, ứng dụng rộng rãi trong các website hiệu quả.

Việc sử dụng **ngôn ngữ JS** có thể ứng dụng cho mọi trình duyệt khác nhau, hiện  được sử dụng phổ biến như Chrome, hay Firefox,… Hơn nữa, đây còn là ngôn  ngữ lập trình hoạt động hiệu quả, được hỗ trợ đầy đủ trên các trình duyệt của thiết bị di động. Bởi thế mà việc sử dụng đa dạng, có thể đáp ứng tốt cho nhiều nhu cầu, những đòi hỏi khác nhau của người dùng.

Nhìn chung, ngôn ngữ **lập trình Javascript** khi sử dụng được đánh giá dễ học,  được phát triển bởi Netscape và sử dụng trong hơn 92% các website ngày nay.  Việc sử dụng JS có thể gắn trong một element trong website, hoặc có thể trong  một sự kiện của trang web tiện lợi và hiệu quả. Chính việc có thể hoạt động  nhanh, nhẹ và mượt mà, … nên ngôn ngữ lập trình này càng được tin tưởng và  ứng dụng nhiều hơn cho nhu cầu của người dùng. Chúng ta có thể thêm trực tiếp JS trực tiếp với HTML, cũng có thể lưu lại trên một files hoàn toàn riêng  biệt và sử dụng khi cần thiết.

**Javascript là ngôn ngữ lập trình dễ dàng sử dụng**

Đơn giản, dễ dàng tìm hiểu và sử dụng là đặc điểm nổi bật của ngôn ngữ lập  trình này. Với những cú pháp khá tương đồng với tiếng Anh thì việc **sử dụng  Javascript** trở nên dễ dàng, dễ tiếp cận hơn rất nhiều. Thông qua mô hình DOM được sử dụng, cung cấp tới nhiều tính năng hữu ích, được viết sẵn đem lại khả năng đáp ứng tốt cho những nhu cầu, những đòi hỏi khác nhau từ phía người  dùng.

Với nhiều tính năng hữu ích mà ngôn ngữ **lập trình Javascript** mang lại thì việc có thể phát triển kịch bản nhằm giải quyết yêu cầu, mục đích nào đó trở nên dễ dàng hơn. Những mục đích tùy chính của con người được đáp ứng hiệu quả với sự hỗ trợ của JS.

**Sử dụng JS** đảm bảo dễ học, dễ dàng phát hiện lỗi khi xuất hiện, đồng thời nó hoạt động nhanh chóng và nhẹ nhàng hơn nhiều ngôn ngữ lập trình khác. Bởi thế, quá trình sử dụng có thể thích hợp với nhiều đối tượng người dùng, đem lại hiệu quả cao và tránh những ảnh hưởng không mong muốn có thể xảy ra.

**Javascript sở hữu giao diện thân thiện, nhiều tính năng**

Nếu so sánh với nhiều ngôn ngữ lập trình khác thì việc **sử dụng** **Javascript** mang tới ưu điểm ở giao diện thân thiện, có thể phù hợp với mọi đối tượng người dùng. Đem tới tính tương tác cao giúp quá trình ứng dụng của mỗi người có được kết quả tốt đẹp. Cùng với nhiều tính năng mà ngôn ngữ này mang tới chắc chắn sẽ giúp website chúng ta tạo ra có được hiệu quả sử dụng lý tưởng.

Có thể kể đến như việc tiến hành thêm slide, hay slideshow, thực hiện hiệu ứng cuộn trang khi chuột di tới, tính năng kéo thả,… đều được cung cấp đầy đủ. Từ đó giúp việc sử dụng ngôn ngữ này trong lập trình đem lại lợi ích lớn, thiết thực.

**So sánh Javascript với các ngôn ngữ lập trình khác**

Mỗi ngôn ngữ lập trình khác nhau mang những đặc điểm khác biệt nhất định. Đó là điều hiển nhiên mà chúng ta có thể tìm hiểu và xác định cụ thể và chính xác. Theo đó, ngôn ngữ Javascript nếu đem ra so sánh với những ngôn ngữ khác thì điểm khác biệt lớn, tạo nên tính phổ biến của nó chính là tính linh hoạt. Bởi thế, có khá nhiều các lập trình viên lựa chọn Javascript trở thành ngôn ngữ chính, sử dụng các ngôn ngữ khác là phụ, bổ sung khi cần thiết.

Javascript: Với ngôn ngữ lập trình này đảm bảo giúp tăng hiệu quả tương tác trên website ở mức tốt nhất. Với Script này khi hoạt động trên các trình duyệt của người dùng, thay vì sử dụng trên các server hiệu quả. Không chỉ vậy, nó thường được dùng trên thư viện của một bên thứ ba, từ đó giúp tăng thêm chức năng cho website dễ dàng, thay vì việc phải thông qua code ngay từ ban đầu.

HTML: (viết tắt của từ HyperText Markup Language, hay còn gọi là "Ngôn ngữ Đánh dấu Siêu văn bản"), là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web trên World Wide Web. Được tin dùng nhiều hiện nay trên các website, giúp việc xây dựng các khối chính của trang web được thực hiện hiệu quả.

PHP: Đây là ngôn ngữ ở phía server, nó hoàn toàn khác biệt so với JS được sử dụng chạy trên các máy client. Ngôn ngữ này được ứng dụng chủ yếu trong các hệ quản trị nội dung nền PHP, tiêu biểu như WordPress. Ngoài ra, sử dụng PHP còn được ứng dụng trong việc lập trình backend, giúp tạo ra những kênh để truyền thông tin hiệu quả nhất từ database.

CSS: (viết tắt của từ Cascading Style Sheets) có nhiệm vụ chính là việc giúp webmaster có thể xác định được styles, cũng như định nghĩa tốt nhiều loại nội dung khác nhau. Người dùng có thể tiến hành thực hiện những công việc hoàn toàn thủ công, với những yếu tố có trong HTML. Tuy nhiên, nếu thực hiện theo cách này đòi hỏi chúng ta phải lặp đi lặp lại các thành phần đó, để sử dụng ở nhiều nơi khác nhau.

**Nói tóm lại**, tìm hiểu về bản chất của từng ngôn ngữ lập trình giúp chúng ta có cái nhìn tổng quan nhất, cơ bản nhất về những ngôn ngữ thông dụng này. Nếu chúng ta nhìn nhận việc lập trình như là xây dựng một căn nhà thì trong đó: HTML có nhiệm vụ chính là định dạng được kiến trúc của ngôi nhà, CSS có nhiệm vụ là thảm và tường giúp căn nhà được trang trí đẹp hơn, trong khi đó **Javascript** có nhiệm vụ giúp tăng cường tương tác của ngôi nhà như hệ thống cửa, hay hệ thống đèn chiếu sáng.

Chúng ta hoàn toàn có thể lập trình, tạo nên một trang web mà không cần tới sự hỗ trợ,sử dụng ngôn ngữ JS. Tuy nhiên, một thực tế cho thấy là khi ngôn ngữ Javascript  không được ứng dụng thì nguy cơ chúng ta sẽ tạo ra một website lỗi thời, không đem tới tính ứng dụng cao như những ở thập niên 90 của thế kỷ 20 là điều hoàn toàn có thể xảy ra. Điều này vô tình gây ra tác động tiêu cực tới quá trình ứng dụng web, đáp ứng cho nhu cầu, đòi hỏi ngày càng nhiều, càng cao của người dùng.

**Hạn chế còn tồn tại của ngôn ngữ Javascript:**

Tồn tại song song với những ưu điểm là hạn chế mà ngôn ngữ lập trình Javascript đang phải đối mặt. Việc tìm hiểu và xác định được những hạn chế giúp chúng ta chủ động trong đánh giá, đưa ra quyết định sử dụng sao cho thích hợp nhất.

*Là ngôn ngữ lập trình dễ bị khai thác:* Việc dễ dàng bị khai thác, với nhiều đối tượng người dùng khác nhau vô tình khiến việc bảo mật cho ngôn ngữ lập trình này không được đánh giá cao. Bởi thế, quá trình sử dụng luôn tồn tại những nguy cơ nhất định cần chúng ta tìm hiểu và chủ động chú ý cẩn trọng.

Một hạn chế của ngôn ngữ JS chính là việc chúng ta có thể sử dụng để thực thi mã độc trên máy tính của người dùng. Tình trạng này vô tình gây ra những thiệt hại, những ảnh hưởng không hề nhỏ tới công việc mà chúng ta đang thực hiện.

Trong một số trường hợp, trong một vài trình duyệt cụ thể việc sử dụng **ngôn ngữ lập trình Javascript** không thể thực hiện được. Nó gây ra ảnh hưởng tới việc ứng dụng, đáp ứng cho nhu cầu của từng người.

Sử dụng ngôn ngữ lập trình này có một hạn chế khác khi có JS code snippets khá lớn.

Khi có khả năng triển khai khác nhau tùy từng thiết bị, nó vừa là ưu điểm song cũng tồn tại những hạn chế nhất định. Chính điều này khiến nguy cơ không đồng nhất có thể xảy ra, gây ảnh hưởng tới quá trình thực hiện mục tiêu cuối cùng.

**1.4.2. Tổng quan về React**

**Giới thiệu React**

React (Hay React JS, ReactJS) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi Jordan  Walke, một kỹ sư phần mềm tại Facebook. Người bị ảnh hưởng bởi XHP (Một nền tảng thành phần HTML cho PHP). React lần đầu tiên được triển khai cho ứng dụng Newsfeed của Facebook năm 2011, sau đó được triển khai cho Instagram.com năm 2012. Nó được mở mã nguồn (open-sourced) tại JSConf US tháng 5 năm 2013.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

**Virtual DOM**

Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

Cơ chế one-way data binding – luồng dữ liệu 1 chiều. Dữ liệu được truyền từ parent đến child thông qua props. Luồng dữ liệu đơn giản giúp chúng ta dễ dàng kiểm soát cũng như sửa lỗi. Với đặc điểm ở trên, React dùng để xây dựng các ứng dụng lớn mà dữ liệu của chúng thay đổi liên tục theo thời gian. Dữ liệu thay đổi thì hầu hết kèm theo sự thay đổi về giao diện. Ví dụ như Facebook:  trên Newsfeed của bạn cùng lúc sẽ có các status khác nhau và mỗi status lại có số like, share, comment liên tục thay đổi. Khi đó React sẽ rất hữu ích để sử dụng.

**JSX**

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster - Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer - An toàn hơn. Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier - Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascript có thể sử dụng.

**Components**

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ.

Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác.

Mỗi component trong React có một trạng thái riêng, có thể thay đổi và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ trong React đều là component.

Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

**Props và State**

Props: Giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.

State: Thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

**1.4.3. Giới thiệu về NodeJS**

NodeJS là một JavaScript runtime được build dựa trên Chrome’s V8 JavaScript engine. NodeJS sử dụng mô hình event-driven, non-blocking I/O khiến nó trở nên nhẹ và hiệu quả.

V8 engine là một JavaScript engine mã nguồn mở chạy trên các trình duyệt Chrome, Opera và Vivaldi. Nó được thiết kế tập trung vào hiệu năng và chịu trách nhiệm cho việc dịch mã JavaScript sang mã máy để máy tính có thể hiểu và chạy được. Nhưng bạn cũng nên tránh nhầm lẫn rằng Node chạy trên trình duyệt. Cha đẻ của Node dựa trên V8 engine, cải tiến một số tính năng chẳng hạn file system API, thư viện HTTP và một số phương thức liên quan đến hệ điều hành. Điều đó có nghĩa là NodeJS là một chương trình giúp ta có thể chạy code JavaScript trên máy tính, nói cách khác nó là một JavaScript runtime.

**Mô hình hoạt động của NodeJS**

Để cho dễ hiểu, khi bạn connect đến một server truyền thống, chẳng hạn Apache, nó sẽ sinh ra một thread mới để xử lý request. Ở các ngôn ngữ như PHP hay Ruby, mỗi một phép toán I/O (ví dụ truy cập database) sẽ chặn execution trên code của bạn cho đến khi phép toán đó hoàn thành. Nói cách khác, server sẽ đợi cho đến khi database được duyệt xong mới xử lý kết quả. Nếu có những request mới, server lại tiếp tục sinh những thread mới để xử lý chúng. Điều này dẫn đến nguy cơ kém hiệu quả, khi một lượng lớn thread được tạo ra sẽ khiến cho hệ thống trở nên chậm chạp, tệ hơn nữa có thể khiến site bị sập. Cách thông thường để giải quyết tình trạng này là bổ sung thêm server.

NodeJS, mặt khác là single-threaded. Nó cũng thuộc dạng event-driven hay nói cách khác tất cả những gì xảy ra trong Node là để phản hồi lại với một sự kiện. Ví dụ, khi một request được gửi đến, server bắt đầu xử lý nó. Nếu nó gặp phải phép toán I/O, thay vì đợi cho phép toán này kết thúc, nó sẽ đăng ký một callback trước khi tiếp tục xử lý event tiếp theo. Khi phép toán I/O kết thúc, server sẽ chạy callback và tiếp tục làm việc trên request ban đầu. Ở tầng bên dưới, Node sử dụng thư viện libuv để thực hiện hoạt động asynchronous (non-blocking) này. Mô hình hoạt động này của Node giúp server có thể xử lý một lượng lớn kết nối đến đồng thời. Quan điểm truyền thống để scale một Node app là clone nó và để instance được clone chia sẻ công việc. NodeJS thậm chí có module built-in sẵn để giúp bạn thực hiện chiến lược clone này trên một server duy nhất.

**Ưu điểm của NodeJS**

Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Bạn có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất.

Giúp bạn dễ dàng mở rộng khi có nhu cầu phát triển website. Ứng dụng về các sự kiện và lắng nghe.

Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử Nodejs lý nhanh hơn rất nhiều.

Có khả năng xử lý nhiều Requests cùng một lúc trong thời gian ngắn nhất.

Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.

**Nhược điểm của NodeJS**

Nodejs gây hao tốn tài nguyên và thời gian. Nodejs được viết bằng C++ và JavaScript nên khi xử lý cần phải trải qua một quá trình biên dịch. Nếu bạn cần xử lý những ứng dụng tốn tài nguyên CPU thì không nên sử dụng Nodejs.

Nodejs so với các ngôn ngữ khác như PHP, Ruby và Python sẽ không có sự chênh lệch quá nhiều. Nodejs có thể sẽ phù hợp với việc phát triển ứng dụng mới. Tuy nhiên khi xây dựng và triển khai dự án quan trọng thì Nodejs không phải là sự lựa chọn hoàn hảo nhất.

**1.4.4. Giới thiệu về ExpressJS**

**Giới thiệu ExpressJS**

Trong thế giới lập trình, **ExpressJS** đã trở thành một từ khóa quen thuộc đối với các lập trình viên. Với khả năng xây dựng ứng dụng web một cách nhanh chóng, ExpressJS đã trở thành một lựa chọn hàng đầu cho việc phát triển ứng dụng web.

**ExpressJS là gì?**

ExpressJS là một framework mạnh mẽ và phổ biến được xây dựng trên nền tảng **NodeJS**. Nó giúp đơn giản hóa quá trình phát triển ứng dụng web bằng cách cung cấp các công cụ và thư viện cần thiết để xử lý yêu cầu và phản hồi HTTP một cách dễ dàng.

ExpressJS cho phép bạn xây dựng các ứng dụng web độc lập, linh hoạt và có hiệu suất cao. Với cú pháp đơn giản và dễ hiểu, ExpressJS giúp lập trình viên tập trung vào việc xây dựng logic ứng dụng thay vì phải lo lắng về các chi tiết cơ bản của việc xử lý yêu cầu và phản hồi.

Điều đặc biệt về ExpressJS là khả năng sử dụng **middleware**. Middleware là các chức năng trung gian được thực thi trước khi yêu cầu của người dùng được xử lý hoặc sau khi phản hồi được gửi đi. Điều này giúp bạn thực hiện các tác vụ như xác thực người dùng, ghi log, xử lý lỗi và nhiều hơn nữa một cách dễ dàng.

**Tính năng của ExpressJS là gì?**

ExpressJS là một framework NodeJS mã nguồn mở phổ biến được sử dụng rộng rãi trong **lập trình web**. Nó đi kèm với một loạt tính năng mạnh mẽ, cho phép lập trình viên xây dựng và triển khai các ứng dụng web nhanh chóng và dễ dàng. Dưới đây là một vài tính năng chính của ExpressJS:

**Phát triển máy chủ nhanh hơn:** ExpressJS tối ưu hóa cú pháp và cung cấp các phương thức và hàm tiện ích để xử lý các tác vụ phổ biến trong lập trình web. Nhờ đó, bạn có thể viết code ngắn gọn và tối giản hóa quy trình phát triển.

**Định tuyến (Routing):** ExpressJS cung cấp một hệ thống định tuyến mạnh mẽ, cho phép bạn xác định các tuyến đường (routes) để xử lý yêu cầu từ người dùng và phản hồi tương ứng. Điều này giúp tổ chức và quản lý các thành phần của ứng dụng một cách dễ dàng.

**Middleware:** ExpressJS hỗ trợ middleware, cho phép bạn thêm các chức năng trung gian vào quy trình xử lý yêu cầu và phản hồi. Middleware giúp xác thực người dùng, ghi log, xử lý lỗi, nén dữ liệu và thực hiện nhiều tác vụ khác một cách linh hoạt.

**Cấu hình môi trường:** ExpressJS cung cấp một cách để cấu hình môi trường phát triển và môi trường sản xuất. Bạn có thể thiết lập các biến môi trường, cấu hình định dạng và quy tắc, tùy chỉnh ứng dụng của mình theo các môi trường khác nhau.

**Xử lý lỗi:** ExpressJS cung cấp cơ chế xử lý lỗi cho phép bạn kiểm soát và xử lý các lỗi xảy ra trong quá trình xử lý yêu cầu. Bạn có thể tạo ra các middleware để xử lý lỗi và phản hồi với các thông báo lỗi tùy chỉnh.

**Ưu điểm của ExpressJS**

Đơn giản và dễ sử dụng: ExpressJS có cú pháp đơn giản và dễ hiểu, giúp lập trình viên dễ dàng nắm bắt và triển khai các tính năng.

Linh hoạt: ExpressJS không áp đặt một cấu trúc cụ thể, cho phép lập trình viên tự do tùy chỉnh và xây dựng ứng dụng theo ý muốn.

Hỗ trợ middleware: ExpressJS cung cấp hệ thống middleware mạnh mẽ, cho phép thực hiện các chức năng như xác thực, ghi log, nén dữ liệu và xử lý lỗi một cách linh hoạt và dễ dàng.

Hiệu suất cao: ExpressJS được xây dựng trên NodeJS, nền tảng có hiệu suất cao, cho phép xử lý nhanh chóng các yêu cầu web đồng thời và có khả năng mở rộng tốt.

**Nhược điểm của ExpressJS**

Thiếu cấu trúc: Do ExpressJS không áp đặt một cấu trúc nghiêm ngặt, việc tổ chức dự án và quản lý mã nguồn có thể trở nên khó khăn, đặc biệt khi ứng dụng phát triển lớn và phức tạp.

Khả năng mở rộng: Khi ứng dụng phát triển lớn và phức tạp, việc quản lý mã nguồn và mở rộng có thể trở nên khó khăn với ExpressJS. Cần có sự kiểm soát cẩn thận để tránh sự phức tạp và rối rắm trong việc quản lý các module và tương tác giữa chúng.

Cộng đồng hỗ trợ: Mặc dù ExpressJS có một cộng đồng lớn và đầy đủ tài liệu, tuy nhiên, không đạt được mức độ hỗ trợ như các framework web khác như **Angular** hoặc **React**.

**1.4.5. Giới thiệu về MongoDB**

**Giới thiệu MongoDB**

MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

**Tìm hiểu về MongoDB: Khái niệm NoSQL**

NoSQL còn được xem là mảnh vá cho những hạn chế, khiếm khuyết của mô hình Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS). Đó là cải tiến về tốc độ, tính năng cũng như khả năng mở rộng..

NoSQL cho phép mở rộng dữ liệu mà không cần tạo khóa ngoại, khóa chính hay kiểm tra tính ràng buộc… Có thể thấy, dạng cơ sở dữ liệu này có thể linh động, sẵn sàng bỏ qua tính toàn vẹn cứng nhắc của transaction hay dữ liệu nhằm đổi lấy hiệu suất, cùng với khả năng mở rộng tốt. Chính vì ưu điểm này mà nhiều Tập đoàn lớn rất ưa chuộng NoSQL như Facebook, Google,…

**Hoạt động của MongoDB**

Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngầm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.

Mỗi bản ghi của MongoDB (document) được gắn một trường có tên “\_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “\_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.

Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.

Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

**Tính năng hệ cơ sở dữ liệu MongoDB**

**Truy vấn ad hoc:** Đây được xem là tính năng tốt nhất của MongoDB. Nó có tác dụng giúp các trường truy vấn phạm vi, kết hợp cùng việc tìm kiếm biểu thức nhằm trả về kết quả tài liệu cụ thể với kích thước nhất định. Ngoài ra, các trường dùng trong MongoDB còn được dùng để lập chỉ mục chính, phụ.

**Nhân rộng:** MongoDB cung cấp Replica Set cho phép nhân 2 hoặc nhiều bản sao của dữ liệu. Đồng thời, mỗi bản sao lại đóng vai trò chính và phụ.

* Khi nhân rộng, toàn bộ dữ liệu khi ghi và đọc được thực hiện trên bản sao chính.
* Bản sao thứ cấp sẽ dùng bản sao tích hợp để có thể duy trì các bản sao dữ liệu.

Trong trường hợp có bất kỳ bản sao chính nào bị thất bại thì Replica set sẽ chọn một bản sao thứ cấp để thay thế làm bản sao chính tiếp theo. Trong quá trình nhân rộng, Replica thứ cấp được tùy ý chọn các hoạt động nhưng dữ liệu cuối cùng vẫn phải tuân theo mặc định.

**Cân bằng tải:** MongoDB sử dụng Sharding nhằm chia tỷ lệ theo chiều ngang và xác định dữ liệu phân phối trong collection. Điều này giúp người dùng có thể chọn một Shard key.

Nói tóm lại, MongoDB cân bằng tải bằng cách dựa vào các Shard key để chia dữ liệu thành các phạm vi và phân phối đồng đều. Chúng có thể chạy trên nhiều máy chủ khác nhau và thực hiện chức năng sao chép dữ liệu hay cân bằng tải nhằm giữ hệ thống hoạt động liên tục trong trường hợp phát sinh lỗi về phần cứng.

**Lưu trữ tệp:** Khi tìm hiểu hệ cơ sở dữ liệu MongoDB thì bạn sẽ thấy, tính năng lưu trữ tệp được dùng như một hệ thống tệp (gọi là GridFS) đóng vai trò cân bằng tải, đồng thời, sao chép dữ liệu trên nhiều máy tính. Cụ thể, GridFS chia một tệp ra làm nhiều phần và lưu trữ thành các tài liệu riêng biệt. Sau đó, người dùng dễ dàng truy cập GridFS thông qua Mongofiles hay các plugin sử dụng cho Nginx và Lighttpd.

**Tập hợp:** Tính năng này chính là chương trình mang đến ba giải pháp để thực hiện tập hợp gồm Aggregation Pipeline, Mapreduce và Single-purpose Aggregation. Trong đó, Aggregation Pipeline được đánh giá là có hiệu suất tốt nhất.

**Giới hạn kích thước Collection:** Các collection được MongoDB hỗ trợ thường có kích thước cố định. Vì thế, người ta gọi chúng là collection giới hạn. Với kích cỡ cố định, kết hợp cùng việc theo sau thứ tự chèn giúp tăng hiệu suất của các hoạt động liên quan đến dữ liệu. Và khi dữ liệu vượt giới hạn thì những tài liệu cũ hơn sẽ tự động bị xóa mà bạn không cần thực hiện thao tác thêm bất kỳ dòng lệnh nào.

**Giao dịch:** Bắt đầu từ phiên bản 4.0 ra mắt vào tháng 6 năm 2018, MongoDB được bổ sung thêm tính năng hỗ trợ giao dịch ACID đa tài liệu.

|  |
| --- |
|  |
|  |

**Ưu điểm của MongoDB**

MongoDB mang đến cho người dùng khá nhiều lợi ích:

Linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Nhờ chúng được lưu dưới dạng JSON nên bạn thoải mái chèn bất kỳ thông tin nào tùy theo nhu cầu sử dụng.

Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa hoặc cập nhật dữ liệu. Nhờ MongoDB không có sự ràng buộc trong một khuôn khổ, quy tắc nhất định nào.

Bạn dễ dàng mở rộng hệ thống thông qua việc thêm node vào cluster. Cụm các node này đóng vai trò như thư viện chứa các dữ liệu giao tiếp với nhau.

Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM. Nhờ thế, những lượt truy vấn sau sẽ được rút ngắn thời gian vì chúng không cần đọc từ ổ cứng.

Trường dữ liệu “\_id” (đại diện cho giá trị duy nhất trong mỗi document) được tự động đánh chỉ mục nên hiệu suất luôn đạt mức cao nhất.

**Nhược điểm của MongoDB**

Bên cạnh các ưu điểm, MongoDB vẫn còn tồn tại một số điểm hạn chế mà bạn cần chú ý khi cài đặt và sử dụng:

Vì dữ liệu không bị ràng buộc nên trong quá trình sử dụng, bạn cần cẩn thận trong mọi thao tác nhằm tránh xảy ra những điều không mong muốn, làm ảnh hưởng đến dữ liệu.

Chương trình MongoDB tiêu tốn khá nhiều dung lượng bộ nhớ do dữ liệu được lưu dưới dạng key và value. Bên cạnh đó, một số collection chỉ có sự khác biệt về value nên việc lặp lại key là điều khó tránh khỏi. Điều này dẫn đến thừa dữ liệu.

Thông thường, thời gian để dữ liệu chuyển đổi từ RAM xuống ổ cứng khoảng 60s nên nguy cơ bị mất dữ liệu nếu xảy ra mất điện là điều có thể xảy ra.

**Nên dùng MongoDB trong trường hợp nào?**

Tuy MongoDB khá hữu dụng nhưng không phải bạn có thể dùng nó trong mọi tình huống. Theo kinh nghiệm của Hosting Việt, bạn chỉ dùng MongoDB khi gặp các trường hợp sau:

Khi website cần có tính năng thêm dữ liệu thường xuyên, bởi MongoDB có thể ghi với tốc độ cao và cực kỳ bảo mật. Điều này giúp trang web không bị downtime khi cập nhật dữ liệu.

Khi có hệ thống dữ liệu với dung lượng lớn và cần truy vấn nhanh.

Khi muốn tìm kiếm thông tin trong bộ nhớ nhanh chóng vì MongoDB có tốc độ tìm kiếm thông tin liên quan cực kỳ nhanh.

Mongo rất thích hợp để sử dụng trong trường hợp máy chủ chưa có hệ quản trị cơ sở dữ liệu.

**Không nên dùng MongoDB trong trường hợp nào?**

Nếu website cần nhiều ứng dụng giao dịch (transaction), điển hình như các trang web của ngân hàng thì không nên dùng MongoDB. Do chương trình này không hỗ trợ cơ chế transaction.

Những ứng dụng cần đến SQL.

**1.3. Coze AI Bot**

Coze là một nền tảng phát triển bot trò chuyện AI đa năng. Bất kể bạn có kỹ năng lập trình như thế nào, Coze cho phép bạn nhanh chóng tạo ra các loại bot trò chuyện phong phú được trang bị AI, có khả năng xử lý từ những câu hỏi đơn giản đến các cuộc đối thoại phức tạp. Hơn nữa, bạn có thể xuất bản bot của mình lên các ứng dụng khác nhau, tạo điều kiện cho tương tác động với người dùng trên những ứng dụng đó.

**A screenshot of a chatbot

Description automatically generated**

**Tại sao nên chọn Coze?**

Coze đơn giản hóa và tối ưu hóa quá trình phát triển bot AI với những tính năng sau:

**Plugins:** mở rộng khả năng của bot của bạn với bộ sưu tập plugin đa dạng.

**Built-in plugins**: tích hợp gần một trăm plugin hiện tại, giúp nâng cao tính năng của bot ở nhiều lĩnh vực như trợ giúp thông tin, hỗ trợ du lịch, nâng cao năng suất, hiểu hình ảnh, và nhiều hơn nữa.

**Custom plugins:** nhanh chóng tạo ra plugin tùy chỉnh bằng cách chỉ định đầu vào và đầu ra từ API hiện có.

**Knowledge:** thêm dữ liệu của bạn vào bot với cách quản lý và lưu trữ dữ liệu thân thiện và dễ hiểu.

**Database & variable:** sử dụng bộ nhớ dài hạn với cơ sở dữ liệu và biến số.

**Scheduled tasks:** tự động thực hiện các nhiệm vụ theo lịch trình.

**Workflow:** tạo luồng công việc linh hoạt với giao diện kéo và thả đơn giản.

**Multi agents:** chạy đa nhiệm trong một bot để xử lý các nhiệm vụ phức tạp của người dùng.

**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1. Phân tích yêu cầu đề tài**

**2.1.1 Phát biểu bài toán**

Tên bài toán: “Xây dựng website và tích hợp với Chatbot phục vụ truyền thông và giới thiệu địa điểm di tích lịch sử của thủ đô Hà Nội”

Bài toán đặt ra nhu cầu phát triển một trang web kết hợp với Chatbot để giới thiệu và cung cấp thông tin chi tiết về các địa điểm di tích lịch sử của thủ đô Hà Nội. Trang web này sẽ là một nền tảng trực tuyến cho phép người dùng truy cập, tìm kiếm và khám phá các địa điểm lịch sử trong thành phố Hà Nội. Với trang web giới thiệu các địa điểm di tích lịch sử của Hà Nội, blogger sẽ đăng thông tin về các địa điểm di tích lịch sử lên trang web. Việc đăng thông tin phải tuân theo một trình tự logic. Ví dụ, khi đăng thông tin về Lăng Chủ Tịch Hồ Chí Minh, thông tin phải bao gồm đầy đủ: tên địa danh, vị trí địa lý, thời gian xây dựng, người thiết kế, mục đích xây dựng, cấu trúc và kiến trúc, thông tin lịch sử,... Người quản trị trang web cũng có thể thêm mới, chỉnh sửa hoặc xóa bài viết. Mọi thay đổi sẽ được ghi lại trong cơ sở dữ liệu để cung cấp cho người dùng.

Đối với người dùng, họ có thể truy cập vào trang web để xem thông tin về các di tích lịch sử ở Hà Nội, nhưng không được phép thả cảm xúc hoặc bình luận trực tiếp trên các bài đăng. Để có thể thả cảm xúc hoặc bình luận, người dùng cần phải đăng ký làm thành viên của trang web. Sau khi trở thành thành viên, họ có thể đăng nhập vào trang web bằng email và mật khẩu của mình. Khi đó, họ sẽ được phép thả cảm xúc hoặc bình luận trên các bài đăng mà họ quan tâm. Ngoài ra, người dùng cũng có thể tìm kiếm địa danh theo tên hoặc người đăng (trong trường hợp này là Admin). Khi có người khác phản hồi vào bình luận của họ, người dùng sẽ nhận được thông báo. Họ có thể xem người đã phản hồi bình luận của mình là ai và ở bài đăng nào trong phần thông báo. Người quản trị cũng sẽ được thông báo khi có người dùng tương tác bằng cách thả cảm xúc, bình luận hoặc phản hồi vào bài đăng mà người quản trị đã tham gia bình luận.