**MỤC LỤC**

[MỤC LỤC i](#_Toc195469523)

[DANH MỤC HÌNH ẢNH i](#_Toc195469524)

[DANH MỤC BẢNG BIỂU ii](#_Toc195469525)

[DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT iii](#_Toc195469526)

[MỞ ĐẦU 1](#_Toc195469527)

[0.1. Đặt vấn đề 1](#_Toc195469528)

[0.2. Mục tiêu của đề tài 1](#_Toc195469529)

[0.3. Định hướng giải pháp 1](#_Toc195469530)

[0.4. Bố cục khóa luận 2](#_Toc195469531)

[0.5. Kết luận 3](#_Toc195469532)

[CHƯƠNG 1. TỔNG QUAN 3](#_Toc195469533)

[1.1 Tổng quan về công nghệ sử dụng 3](#_Toc195469534)

[1.1.1 Thư viện và công cụ sử dụng 3](#_Toc195469535)

[1.1.2 Hệ quản trị cơ sở dữ liệu SQL Server 4](#_Toc195469536)

[1.1.3 Front end 5](#_Toc195469537)

[1.1.4 Back end 6](#_Toc195469538)

[1.2 Mô tả bài toán cần xử lý 9](#_Toc195469539)

[1.2.1 Bài toán đặt ra 9](#_Toc195469540)

[1.2.2 Giải pháp 9](#_Toc195469541)

[1.2.3 Kết quả đạt được 10](#_Toc195469542)

[1.2.4 Khó khăn gặp phải 10](#_Toc195469543)

[1.3 Kết luận 11](#_Toc195469544)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 11](#_Toc195469545)

[2.1 Khảo sát hiện trạng 11](#_Toc195469546)

[2.2 Tổng quan chức năng 11](#_Toc195469547)

[2.3 Biểu đồ Use-case tổng quát 11](#_Toc195469548)

[2.4 Biểu đồ phân rã 11](#_Toc195469549)

[2.5 Biểu đồ hoạt động 11](#_Toc195469550)

[2.6 Đặc tả chức năng 11](#_Toc195469551)

[2.7 Yêu cầu phi chức năng 11](#_Toc195469552)

[2.7.1 Yêu cầu về giao diện 11](#_Toc195469553)

[2.7.2 Yêu cầu về bảo mật 12](#_Toc195469554)

[2.7.3 Yêu cầu về hiệu năng 12](#_Toc195469555)

[2.8 Thiết kế cấu trúc 12](#_Toc195469556)

[2.8.1 Lựa chọn cấu trúc phần mềm 12](#_Toc195469557)

[2.8.2 Thiết kế tổng quan 13](#_Toc195469558)

[2.8.3 Thiết kế chi tiết gói 15](#_Toc195469559)

[2.9Thiết kế chi tiết 15](#_Toc195469560)

[2.10Kết luận 15](#_Toc195469561)

[CHƯƠNG 3. XÂY DỰNG WEBSITE 15](#_Toc195469562)

[3.1 Kết quả đạt được 15](#_Toc195469563)

[3.2 Kết luận 15](#_Toc195469564)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc195469565)

[Kết luận: 16](#_Toc195469566)

[Hướng phát triển: 16](#_Toc195469567)

[TÀI LIỆU THAM KHẢO 16](#_Toc195469568)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG BIỂU

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Thuật ngữ** | **Ý nghĩa** |
| API | Application Programming Interface – Giao diện lập trình ứng dụng. Là cầu nối cho phép các thành phần phần mềm giao tiếp và trao đổi dữ liệu với nhau. |
| Angular | Một framework JavaScript mã nguồn mở được phát triển bởi Google, dùng để xây dựng ứng dụng web phía client với khả năng xử lý giao diện động, bảo mật cao và dễ mở rộng. |
| .NET | Một nền tảng phát triển phần mềm do Microsoft xây dựng, hỗ trợ nhiều ngôn ngữ lập trình như C#, F#... Phiên bản .NET 8.0 là phiên bản mới nhất, dùng để xây dựng API mạnh mẽ và linh hoạt. |
| SQL Server | Hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển, cho phép lưu trữ, truy vấn và quản lý dữ liệu một cách an toàn, hiệu quả. |
| Client | Khách – Thiết bị hoặc ứng dụng gửi yêu cầu đến máy chủ (server) để nhận dữ liệu hoặc dịch vụ. |
| HTML | HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản, dùng để xây dựng cấu trúc nội dung hiển thị trên các trang web. |
| CSS | Cascading Style Sheets – Ngôn ngữ định kiểu, dùng để thiết kế giao diện, màu sắc, bố cục cho các thành phần HTML. |
| CRUD | Create - Read - Update - Delete – Tập hợp các thao tác cơ bản trong quản lý dữ liệu: tạo, đọc, cập nhật và xóa. |
| TypeScript | Ngôn ngữ lập trình được phát triển dựa trên JavaScript, bổ sung kiểu dữ liệu tĩnh và các tính năng hướng đối tượng, được Angular sử dụng chính thức. |

# 

# MỞ ĐẦU

Đồ án này của em tập trung vào việc xây dựng một website quản lý thủ tục nội bộ cho Hệ thống Đầu tư công Sở Kế hoạch và Đầu tư, nhằm tối ưu hóa quy trình quản lý quản lý hồ sơ thủ tục nội bộ, lưu trữ hồ sơ và tài liệu liên quan. Website được thiết kế với mục tiêu mang đến trải nghiệm trực tuyến tiện lợi cho chuyên viên, lãnh đạo phòng, lãnh đạo Sở.

Website được phát triển dựa trên các công nghệ web hiện đại như HTML, CSS, JavaScript để tạo ra giao diện người dùng trực quan, thân thiện và tương thích tốt trên nhiều thiết bị khác nhau như điện thoại, máy tính bảng và máy tính cá nhân. Phần backend sử dụng ngôn ngữ lập trình API kết hợp với hệ quản trị cơ sở dữ liệu SQL Server, giúp xử lý các nghiệp vụ bên trong của phần mềm.

Các tính năng nổi bật của website bao gồm hiển thị chi tiết thông tin hồ sơ thủ tục, thêm mới, cập nhật, xóa hồ sơ. Tích hợp chức năng lập phiếu xử lý cho từng loại hồ sơ, chức năng in danh sách hồ sơ dễ dàng, tiện lợi.

Thông qua đồ án này, em đã áp dụng thành công các kiến thức lập trình web vào thực tế, xây dựng một website quản lý thủ tục nội bộ cho Hệ thống Đầu tư công Sở Kế hoạch và Đầu tư hoàn chỉnh có thể phù hợp với yêu cầu và có thể nói là gần như hoàn thiện về mặt nghiệp vụ tương ứng với yêu cầu của Sở Kế hoạch và Đầu tư. Qua đó, em cũng rèn luyện được kỹ năng làm việc, xử lý các tình huống phát sinh trong quá trình phát triển dự án, và củng cố thêm những kiến thức chuyên môn thiết yếu cho ngành công nghệ thông tin.

## Đặt vấn đề

Trong bối cảnh công nghệ thông tin ngày càng phát triển mạnh mẽ và được ứng dụng sâu rộng vào hoạt động của các cơ quan quản lý nhà nước, việc số hóa quy trình xử lý công việc nội bộ là một xu hướng tất yếu nhằm nâng cao hiệu quả điều hành, minh bạch hóa quy trình và giảm thiểu thời gian xử lý. Đặc biệt, đối với lĩnh vực đầu tư công – một lĩnh vực có tính chất phức tạp, liên quan đến nhiều quy trình thủ tục, văn bản hành chính và các bên liên quan – việc triển khai hệ thống phần mềm hỗ trợ quản lý và theo dõi quy trình nội bộ là vô cùng cần thiết.

Hiện nay, tại Sở Kế hoạch và Đầu tư, việc quản lý và xử lý các thủ tục nội bộ phục vụ công tác đầu tư công chủ yếu còn mang tính thủ công, phân tán qua nhiều kênh, cũng như không có hệ thống, phần mềm chuyện biệt để lưu trữ dữ liệu, tài liệu liên quan đến thủ tục nội bộ… Điều này gây ra nhiều khó khăn trong việc theo dõi tiến độ xử lý, tra cứu thông tin lịch sử công việc. Ngoài ra, việc thiếu một hệ thống thống nhất còn dẫn đến tình trạng trùng lặp, sai sót dữ liệu và khó khăn trong công tác báo cáo, tổng hợp.

## Mục tiêu của đề tài

* Xuất phát từ thực trạng đó, đề tài "Xây dựng website quản lý thủ tục nội bộ cho Hệ thống Đầu tư công Sở Kế hoạch và Đầu tư" được thực hiện với mục tiêu xây dựng một hệ thống phần mềm quản lý thủ tục một cách tập trung, hiện đại và thân thiện với người sử dụng.
* Hệ thống sẽ hỗ trợ cán bộ, chuyên viên của Sở trong việc tạo lập, xử lý và quản lý các hồ sơ thủ tục nội bộ liên quan đến đầu tư công một cách nhanh chóng, chính xác và hiệu quả.
* Hỗ trợ quản lý hồ sơ, tìm kiếm hồ sơ một cách nhanh chóng.
* Hệ thống tích hợp việc lập phiêu xử lý hồ sơ thủ tục, xuất danh sách hồ sơ một cách tiện lợi và nhanh chóng.
* Hỗ trợ chuyên viên lập báo cáo, tổng hợp hồ sơ một cách dễ dàng.

Thông qua đề tài này, em hy vọng Việc xây dựng hệ thống này không chỉ góp phần nâng cao chất lượng công tác quản lý nhà nước trong lĩnh vực đầu tư công mà còn là một bước tiến quan trọng trong lộ trình chuyển đổi số của Sở Kế hoạch và Đầu tư.

## Định hướng giải pháp

* Sử dụng framework Angular để phát triển website, tận dụng ưu điểm về bảo mật cao, hiệu năng tốt, khả năng xử lý giao diện nhanh chóng và đặc bieejet là khả năng mở rông, bảo trì dễ dàng trong hệ thống lớn.
* Kết hợp sử dụng API để tận dụng lợi thế của giao tiếp giữa client và server một cách hiệu quả, giúp hệ thống dễ dang mở rộng, cải thiện khả năng tương tác và động bộ dữ liệu nhanh chóng giữa các thành phần của hệ thống.
* Áp dụng kiến trúc MVC (Model-View-Controller) nhằm giúp mã nguồn dễ dàng bảo trì, quản lý và nâng cấp.
* Xây dựng giao diện thân thiện, tương thích tốt trên cả máy tính và các thiết bị di động.
* Sử dụng cơ sở dữ liệu SQL Server để lưu trữ và quản lý dữ liệu.

## Bố cục khóa luận

Phần còn lại của báo cáo sẽ được tổ chức như sau:

**Chương 1: Khảo sát và phân tích yêu cầu**

Trong chương này, em sẽ tiến hành khảo sát nhu cầu từ phía khách hàng và dựa trên các tài liệu liên quan, web đầu tư công của bộ. Dựa trên kết quả khảo sát, em sẽ xác định các chức năng quan trọng cần phát triển, phân loại nhóm đối tượng sử dụng hệ thống (bao gồm chuyên viên xử lý và lãnh đạo ban ngành, lãnh đạo Sở) và đề xuất các kịch bản sử dụng (use case) cụ thể cho từng đối tượng. Ngoài ra, chương này cũng sẽ trình bày chi tiết các đặc tả về yêu cầu hệ thống, từ yêu cầu chức năng đến phi chức năng.

**Chương 2: Công nghệ sử dụng**

Chương này sẽ giới thiệu về các công nghệ cốt lõi được áp dụng trong quá trình xây dựng hệ thống, công cụ hỗ trợ và các cơ sở lý thuyết liên quan. Em sẽ phân tích ưu nhược điểm của từng công nghệ, chẳng hạn như HTML, CSS, JavaScript cho giao diện người dùng Angular, API cùng SQL Server cho phần backend, giải thích lý do lựa chọn chúng trong dự án này. Đồng thời, cấu trúc tổng quan của hệ thống cũng sẽ được mô tả chi tiết nhằm thể hiện cách các thành phần phần mềm tương tác với nhau.

**Chương 3: Quá trình phát triển và đánh giá hệ thống**

Chương này sẽ đi sâu vào các giai đoạn phát triển hệ thống, từ thiết kế tổng thể đến thiết kế chi tiết từng thành phần như giao diện người dùng, cơ sở dữ liệu và các mô-đun chức năng (quản lý hồ sơ, lập phiếu xử lý cho từng hồ sơ, tổng hợp báo cáo, xuất exel báo cáo, danh sách hồ sơ). Em sẽ trình bày quy trình kiểm thử phần mềm, bao gồm các phương pháp kiểm thử chức năng, hiệu năng và bảo mật được áp dụng nhằm đảm bảo chất lượng của hệ thống. Cuối cùng, chương này sẽ mô tả quá trình triển khai ứng dụng trong môi trường thực tế và đánh giá phản hồi từ người dùng để rút ra những bài học và hướng cải tiến cho các phiên bản tiếp theo.

## Kết luận

Tóm lại, trong chương đầu tiên, em đã trình bày tổng quan về đề tài, làm rõ các vấn đề thực tế mà hệ thống hướng đến giải quyết, đồng thời xác định mục tiêu chính cần đạt được. Bên cạnh đó, bố cục báo cáo khóa luận cũng đã được tóm tắt nhằm giúp người đọc có cái nhìn tổng quan về nội dung các chương tiếp theo

# TỔNG QUAN

Ở chương này, em sẽ đi sâu vào phân tích các công nghệ và nền tảng được sử dụng để giải quyết bài toán, đồng thời nêu rõ lý do lựa chọn các công nghệ đó. Hệ thống được xây dựng giao diện dựa trên Angular, xử lý backend API kết hợp cùng MySQL, giúp tách biệt rõ ràng giữa giao, diện xử lý logic và dữ liệu. Em sẽ trình bày chi tiết cách triển khai giao diện với Angular, backend với API, bao gồm cách tổ chức frontend, backend, cách liên kết giữa giao diện với logic và dữ liệu để đảm bảo hiệu suất, khả năng mở rộng của hệ thống.

## Tổng quan về công nghệ sử dụng

### Thư viện và công cụ sử dụng

*Bảng 1.1: Thư viện và công cụ sử dụng*

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| Công cụ lập trình (backEnd) | Visual Studio | <https://visualstudio.microsoft.com/> |
| Công cụ lập trình (FrontEnd) | Visual Studio Code | <https://code.visualstudio.com/> |
| Quản lý cơ sở dữ liệu | SSMS | <https://learn.microsoft.com/en-us/ssms/sql-server-management-studio-ssms> |
| Test API | Postman | <https://www.postman.com/> |
| Thiết kế cơ sở dữ liệu | SSMS | <https://learn.microsoft.com/en-us/ssms/sql-server-management-studio-ssms> |
| Front-end | Angular | <https://angular.dev/> |
| Back-end | API .Net 8.0 | <https://learn.microsoft.com/vi-vn/dotnet/api/?view=aspnetcore-8.0> |
| Quản lý code | Github | https://github.com/ |

### Hệ quản trị cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ do Microsoft phát triển. Đây là công cụ phổ biến và đáng tin cậy, được sử dụng rộng rãi trong các hệ thống ứng dụng doanh nghiệp nhờ khả năng lưu trữ và xử lý dữ liệu lớn, cùng với độ bảo mật cao và tính ổn định vượt trội.

SQL Server sử dụng ngôn ngữ T-SQL (Transact-SQL) để hỗ trợ các thao tác tạo, chỉnh sửa, truy vấn và quản lý dữ liệu trong cơ sở dữ liệu. Ngoài ra, nó hỗ trợ đa người dùng, xử lý đồng thời nhiều truy vấn, và tích hợp tốt với các hệ thống phần mềm hiện đại.

Trong khuôn khổ đồ án tốt nghiệp, SQL Server được lựa chọn để kết hợp với ASP.NET Core Web API (.NET 8.0) nhằm xây dựng một hệ thống quản lý thủ tục nội bộ có tính bảo mật cao, hiệu năng tốt và dễ mở rộng. Việc sử dụng SSMS (SQL Server Management Studio) giúp thuận tiện trong quá trình quản lý dữ liệu, thiết kế bảng, thực hiện truy vấn và theo dõi hoạt động của cơ sở dữ liệu một cách trực quan.

Khi kết hợp cùng API .NET 8.0 và giao diện Angular, cơ sở dữ liệu SQL Server đóng vai trò trung tâm trong việc lưu trữ, truy xuất và xử lý dữ liệu một cách hiệu quả, đảm bảo sự tương tác nhanh chóng giữa các thành phần của hệ thống.



*Hình 1.1: Công cụ quản trị cơ sở dữ liệu SSMS*

### Front end

#### **Angular**

Angular là một framework phát triển ứng dụng web mã nguồn mở do Google phát triển và duy trì. Angular sử dụng **TypeScript** – một ngôn ngữ lập trình mở rộng từ JavaScript – để xây dựng các ứng dụng web hiện đại, đặc biệt là **SPA (Single Page Application)**.



Hình 1.2: Hình ảnh framework Angular

Việc lựa chọn **Angular** làm công cụ thiết kế giao diện được dựa trên các ưu điểm sau:

* **Hiệu năng cao**: Angular xử lý dữ liệu và cập nhật giao diện nhanh chóng nhờ cơ chế binding dữ liệu hai chiều (two-way data binding).
* **Bảo mật tốt**: Angular hỗ trợ sẵn các cơ chế phòng chống tấn công phổ biến như XSS (Cross-site scripting).
* **Kiến trúc rõ ràng, dễ bảo trì**: Theo mô hình **Component-based**, giúp tái sử dụng và mở rộng dễ dàng.
* **Tích hợp tốt với API**: Angular thường được dùng kết hợp với RESTful API để trao đổi dữ liệu với backend.
* **Hệ sinh thái phong phú**: Hỗ trợ CLI (Command Line Interface), thư viện giao diện như Angular Material, và nhiều công cụ hỗ trợ phát triển khác

Nhờ những đặc điểm này, Angular trở thành một lựa chọn phù hợp trong việc xây dựng giao diện web hiện đại, giúp tối ưu hóa hiệu suất và trải nghiệm người dùng.

#### **Ngôn ngữ lập trình HTML/CSS/TypeScript [3]**

HTML, CSS và TypeScript là những công nghệ nền tảng không thể thiếu trong phát triển ứng dụng web hiện đại được sử dụng trong framework Angular.

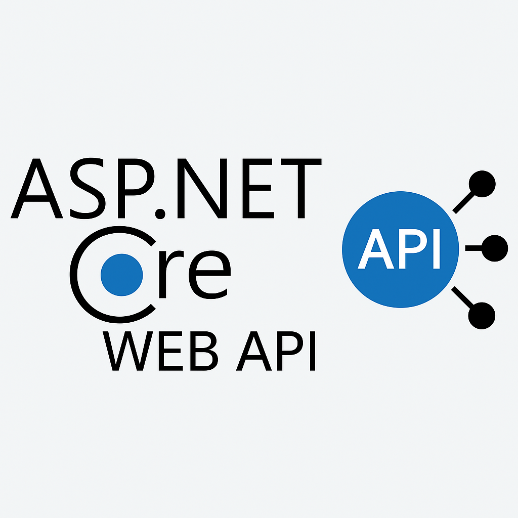
**HTML** (HyperText Markup Language) cung cấp cấu trúc cơ bản của trang web thông qua các thẻ định nghĩa các thành phần như tiêu đề, đoạn văn, hình ảnh, và liên kết.

**CSS** (Cascading Style Sheets) chịu trách nhiệm về giao diện và thiết kế, giúp trang web trở nên trực quan và đẹp mắt.

**TypeScript** là một ngôn ngữ lập trình được phát triển dựa trên JavaScript, có khả năng kiểm tra kiểu dữ liệu tĩnh và hỗ trợ lập trình hướng đối tượng. Đây là ngôn ngữ chính được sử dụng trong framework Angular, giúp quá trình phát triển ứng dụng trở nên an toàn, dễ bảo trì và mở rộng.

### Back end

API (Application Programming Interface) là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, và/hoặc cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. API áp dụng kiến trúc MVC mở rộng theo mô hình phân tầng (layered architecture hoặc clean architecture), tách biệt rõ ràng giữa các thành phần controller, entity, service và model, giúp tăng tính mô-đun, dễ bảo trì và mở rộng hệ thống.



*Hình 1.4: ASP.NET Core Web API*

#### Hệ thống routing linh hoạt

ASP.NET Core cung cấp cơ chế định tuyến (routing) cực kỳ linh hoạt, cho phép ánh xạ URL đến các hành động trong controller hoặc các hàm handler.

Hỗ trợ route parameters, query strings, và default values.

Có thể kết hợp với middleware như UseAuthentication, UseAuthorization để xử lý trước khi tới endpoint.

Hệ thống định tuyến có thể dễ dàng mở rộng với middleware tùy chỉnh, rate-limiting, hoặc logging.

#### 1.1.4.2 Quản lý cơ sở dữ liệu với Entity Framework Core

Entity Framework Core (EF Core) là ORM chính thức của Microsoft, tích hợp sâu trong ASP.NET Core để thao tác với cơ sở dữ liệu quan hệ như SQL Server, PostgreSQL, MySQL....

Ưu điểm của EF Core:

* Code-first: Thiết kế class trước rồi tự động sinh schema DB bằng Migration.
* Database-first: Sinh model từ database hiện có thông qua Scaffold-DbContext.
* LINQ support: Cho phép truy vấn dữ liệu bằng cú pháp gần gũi với lập trình viên C#.
* Lazy loading, eager loading, explicit loading cho phép tối ưu hóa truy vấn.

#### 1.1.4.3 MVC trong API

**MVC (Model-View-Controller)** là một mô hình thiết kế phần mềm được sử dụng rộng rãi trong phát triển ứng dụng, bao gồm cả ứng dụng API. Mô hình này giúp tách biệt các phần khác nhau trong ứng dụng để giảm thiểu sự phức tạp, tăng khả năng tái sử dụng và duy trì mã nguồn tốt hơn.

Trong mô hình MVC, **Model** đại diện cho dữ liệu và các quy tắc nghiệp vụ (business logic) của ứng dụng. Tuy nhiên, khi sử dụng **Entities**, vai trò của **Model** sẽ được thay thế hoặc cải tiến. Thực thể (Entities) là những lớp đại diện cho dữ liệu trong cơ sở dữ liệu và các mối quan hệ giữa các đối tượng trong ứng dụng. Các thực thể này sẽ trực tiếp ánh xạ với các bảng trong cơ sở dữ liệu và chứa các phương thức để xử lý dữ liệu như lưu trữ, truy xuất, và các tác vụ khác liên quan đến nghiệp vụ.

Các **Entities** đóng vai trò là mô hình dữ liệu chính trong ứng dụng, và có thể tích hợp các quy tắc nghiệp vụ vào chính các thực thể này thay vì phải có một lớp riêng biệt cho các logic xử lý dữ liệu.

* *Ánh xạ với cơ sở dữ liệu*: Các **Entities** tương ứng với các bảng trong cơ sở dữ liệu, với mỗi thực thể đại diện cho một bảng hoặc một đối tượng trong hệ thống. Các thuộc tính của thực thể sẽ ánh xạ với các cột trong bảng cơ sở dữ liệu.
* *Quy tắc nghiệp vụ*: Các quy tắc xử lý dữ liệu có thể được tích hợp trực tiếp vào các thực thể, giúp tách biệt logic nghiệp vụ khỏi phần còn lại của ứng dụng. Ví dụ, có thể xác thực dữ liệu trong các phương thức của thực thể, hoặc thực hiện các thao tác như tính toán, kiểm tra tính hợp lệ, v.v.

**Controller** trong mô hình MVC (API sử dụng Entity) có vai trò nhận các yêu cầu từ client, gọi các phương thức trong **Entities** để lấy hoặc thay đổi dữ liệu, sau đó trả kết quả cho client thông qua **View.**

Trong một API, **View** thường không phải là giao diện người dùng như trong ứng dụng web truyền thống mà là cách dữ liệu được trả về cho client. API sẽ trả về dữ liệu dưới dạng JSON hoặc các định dạng dữ liệu khác mà client có thể sử dụng. Ví dụ : Khi một client gửi yêu cầu GET tới /api/…, API sẽ trả về thông tin dưới dạng JSON.

#### 1.1.4.4 EPPlus – thao tác với tệp Excel

EPPlus là một thư viện mã nguồn mở cho phép thao tác với các tệp Excel (định dạng .xlsx) trong môi trường .NET. Thư viện này thường được sử dụng để tạo, đọc, và sửa đổi các tệp Excel mà không cần cài đặt Microsoft Office hoặc Excel trên máy chủ, giúp tiết kiệm tài nguyên hệ thống.

* **Tạo và đọc các tệp Excel (.xlsx)**: Bạn có thể tạo các bảng tính, thêm nội dung vào các ô, định dạng dữ liệu, và lưu lại dưới dạng tệp Excel mà không cần phần mềm Excel cài đặt trên máy.
* **Định dạng bảng tính**: EPPlus cho phép bạn định dạng văn bản, số, màu sắc, phông chữ, cỡ chữ, và các thuộc tính khác của ô trong bảng tính Excel.
* **Hỗ trợ cho các tính năng nâng cao**: EPPlus còn hỗ trợ các tính năng như vẽ biểu đồ, quản lý vùng dữ liệu, bảo vệ bảng tính, và nhiều tính năng hữu ích khác.

Lợi ích khi sử dụng EPPlus trong API:

* **Hiệu suất tốt**: EPPlus không yêu cầu Microsoft Office cài đặt và hoạt động rất nhanh, phù hợp cho các ứng dụng web hoặc API cần xuất báo cáo nhanh chóng.
* **Dễ sử dụng**: Với cú pháp dễ hiểu, việc sử dụng EPPlus trong API trở nên rất thuận tiện và nhanh chóng.
* **Mã nguồn mở**: EPPlus là mã nguồn mở và miễn phí sử dụng, giúp tiết kiệm chi phí bản quyền phần mềm, dễ tiếp cận với người mới.

## 1.2 Mô tả bài toán cần xử lý

### 1.2.1 Bài toán đặt ra

Hiện nay, trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ tại các cơ quan nhà nước, việc quản lý và xử lý các thủ tục hành chính, đặc biệt là các thủ tục liên quan đến đầu tư công, vẫn còn tồn tại nhiều bất cập khi thực hiện theo phương thức thủ công hoặc bán thủ công như:

* Hồ sơ, tài liệu trao đổi chủ yếu bằng giấy tờ hoặc email, dẫn đến việc lưu trữ, tra cứu, và theo dõi tiến độ thực hiện xử lý công việc gặp nhiều khó khăn.
* Thiếu công cụ hỗ trợ kiểm soát quy trình xử lý thủ tục từ khâu tạo mới, cập nhật tiến độ xử lý hồ sơ đến khi hoàn tất.
* Việc thống kê, báo cáo, lập phiếu xử ly hồ sơ, tổng hợp nhiều bộ hồ sơ trong năm còn thủ công, gây chậm trễ và không đảm bảo tính chính xác, minh bạch.

Chính vì vậy, đặt ra nhu cầu cấp thiết về việc xây dựng một hệ thống phần mềm quản lý thủ tục nội bộ giúp số hóa toàn bộ quy trình tạo lập và xử lý hồ sơ trong lĩnh vực đầu tư công tại Sở Kế hoạch và Đầu tư.

### 1.2.2 Giải pháp

Để giải quyết các vấn đề còn tồn tại trong công tác xử lý thủ tục đầu tư công tại Sở Kế hoạch và Đầu tư, khóa luận đề xuất xây dựng một hệ thống phần mềm quản lý thủ tục nội bộ với các giải pháp chính như sau:

* **Quản lý toàn bộ quy trình xử lý thủ tục đầu tư công một cách minh bạch, thống nhất và trực tuyến**: Hệ thống được thiết kế nhằm số hóa toàn bộ các bước xử lý hồ sơ, từ tạo lập, cập nhật, cập nhật kết quả, chuyển chuyên viên xử lý đến hoàn thiện, tổng hợp báo cáo hồ sơ giúp đảm bảo tính nhất quán, giảm thiểu tình trạng sai sót và phân tán thông tin. Quy trình được mô hình hóa theo các nghiệp vụ thực tế, cho phép kiểm soát tiến độ và trách nhiệm của từng cá nhân tham gia viễ ử lý hồ sơ.
* **Theo dõi trạng thái xử lý hồ sơ theo thời gian thực**: Mỗi hồ sơ khi được đưa vào hệ thống đều có thể được theo dõi chi tiết theo thời gian thực. Người quản lý và các cán bộ có liên quan có thể kiểm tra được hồ sơ đang ở bước xử lý nào, ai là người phụ trách, thời hạn xử lý ra sao, từ đó nâng cao hiệu quả giám sát và điều phối công việc.
* **Tự động tổng hợp báo cáo, thống kê phục vụ công tác quản lý, giám sát**: Hệ thống hỗ trợ chức năng báo cáo theo các tiêu chí từ đó giúp ban lãnh đạo có cái nhìn tổng thể và kịp thời ra quyết định điều hành. Việc tổng hợp dữ liệu được thực hiện tự động, giảm phụ thuộc vào thao tác thủ công của cán bộ.
* **Tăng hiệu suất làm việc của cán bộ, giảm thiểu sai sót và tiết kiệm thời gian xử lý thủ tục**: Việc ứng dụng công nghệ vào quy trình giúp chuẩn hóa thao tác nghiệp vụ, hạn chế việc xử lý sai quy trình hoặc thiếu sót thông tin. Giao diện thân thiện, rõ ràng sẽ hỗ trợ cán bộ xử lý nhanh chóng và chính xác hơn, từ đó nâng cao hiệu quả làm việc tổng thể.

Việc triển khai hệ thống không chỉ góp phần nâng cao hiệu quả quản lý nội bộ tại Sở Kế hoạch và Đầu tư mà còn đóng vai trò quan trọng trong việc thúc đẩy quá trình **chuyển đổi số** trong lĩnh vực đầu tư công, phù hợp với định hướng phát triển Chính phủ điện tử và chính quyền số do Chính phủ và Bộ Kế hoạch – Đầu tư đề ra.

### 1.2.3 Kết quả đạt được

Sau quá trình nghiên cứu, phân tích và triển khai, đề tài đã đạt được một số kết quả chính như sau:

* **Xây dựng thành công hệ thống website quản lý thủ tục nội bộ trong lĩnh vực đầu tư công**: Hệ thống được phát triển dựa trên nền tảng công nghệ web hiện đại, đảm bảo các tính năng quản lý hồ sơ, phân công xử lý, theo dõi tiến độ và tổng hợp báo cáo được vận hành hiệu quả.
* **Số hóa toàn bộ quy trình xử lý hồ sơ thủ tục**: Thay thế phương thức làm việc thủ công bằng quy trình điện tử hóa, giúp cán bộ dễ dàng tiếp cận, xử lý và lưu trữ hồ sơ trên nền tảng số. Các bước xử lý được tự động hóa một phần, đảm bảo đúng trình tự nghiệp vụ và giảm thiểu sai sót.
* **Giao diện người dùng thân thiện, dễ sử dụng**: Hệ thống có thiết kế đơn giản, rõ ràng, phù hợp với môi trường hành chính công, giúp người dùng nhanh chóng làm quen và sử dụng hiệu quả ngay cả khi không có kiến thức chuyên sâu về công nghệ.

Với các kết quả đạt được, đề tài đã phần nào đáp ứng được yêu cầu thực tiễn của đơn vị và đóng góp vào tiến trình chuyển đổi số trong lĩnh vực quản lý đầu tư công tại Sở Kế hoạch và Đầu tư.

### 1.2.4 Khó khăn gặp phải

Trong quá trình thực hiện đề tài, cá nhân thực hiện gặp phải một số khó khăn nhất định, chủ yếu xoay quanh việc tiếp cận nghiệp vụ thực tế và thu thập thông tin:

* **Không được khảo sát trực tiếp với cán bộ tại Sở Kế hoạch và Đầu tư**: Do giới hạn về thời gian, phạm vi đề tài cũng như yếu tố khách quan từ phía đơn vị, cá nhân không có cơ hội tiếp xúc trực tiếp với người sử dụng thực tế tại Sở để tìm hiểu chi tiết quy trình nghiệp vụ cũng như các tình huống phát sinh trong quá trình xử lý hồ sơ.
* **Phải tự nghiên cứu nghiệp vụ thông qua tài liệu thứ cấp**: Việc phân tích quy trình nghiệp vụ chủ yếu dựa trên các tài liệu thiết kế của một bên khác (nếu có), các văn bản pháp lý liên quan đến đầu tư công, và các nguồn tham khảo tìm kiếm trên internet. Điều này dẫn đến khó khăn trong việc nắm bắt đầy đủ, chính xác và cập nhật các bước xử lý hồ sơ, cũng như những quy tắc nghiệp vụ đặc thù tại từng địa phương.
* **Thiếu phản hồi từ người dùng thực tế**: Không có điều kiện để lấy ý kiến từ cán bộ trực tiếp sử dụng hệ thống nên cá nhân thực hiện gặp khó khăn trong việc đánh giá tính thân thiện, trực quan của giao diện cũng như hiệu quả của hệ thống trong thực tiễn vận hành.
* **Rủi ro sai lệch giữa thiết kế và thực tiễn**: Do các thông tin phân tích chủ yếu mang tính gián tiếp, có thể xảy ra tình trạng hệ thống được thiết kế chưa hoàn toàn phù hợp với quy trình thực tế, cần được hiệu chỉnh thêm nếu triển khai chính thức tại cơ quan nhà nước.

Mặc dù gặp nhiều hạn chế trong quá trình khảo sát và phân tích nghiệp vụ, nhóm thực hiện đã cố gắng tìm hiểu kỹ lưỡng từ các tài liệu có độ tin cậy cao, tham khảo nhiều nguồn chính thống và áp dụng các nguyên tắc thiết kế hệ thống hành chính phổ biến để đảm bảo tính thực tế và khả năng ứng dụng cao cho sản phẩm phần mềm.

## 1.3 Kết luận

Trong **chương 1**, em đã trình bày chi tiết về các công nghệ, phần mềm và thư viện hỗ trợ được sử dụng trong quá trình phát triển khóa luận. Những nội dung này không chỉ giúp làm rõ lý do lựa chọn từng công nghệ mà còn cung cấp cái nhìn tổng quan về cách chúng được áp dụng vào hệ thống. Việc hiểu rõ những nền tảng này sẽ giúp mọi người hình dung cụ thể hơn về cách em đã xây dựng và triển khai website, cũng như cách các thành phần trong hệ thống phối hợp để đảm bảo hiệu suất, bảo mật và khả năng mở rộng.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Trong **Chương 1**, em đã giới thiệu tổng quan về khóa luận, xác định các vấn đề cần giải quyết và đề xuất hướng tiếp cận phù hợp. Tiếp nối những nội dung này, **Chương 2** sẽ tập trung vào quá trình khảo sát và phân tích thực tiễn thông qua việc nghiên cứu, đánh giá các nền tảng quản lý tài liệu học tập hiện có. Dựa trên kết quả khảo sát, em sẽ phân tích thực trạng để xác định các chức năng quan trọng cần phát triển cho hệ thống. Bên cạnh đó, chương này cũng sẽ trình bày biểu đồ **use case tổng quan**, phân rã các chức năng chính thành các **use case chi tiết**, đồng thời tiến hành đặc tả các biểu đồ này để làm rõ từng luồng hoạt động trong hệ thống.

## 2.1 Khảo sát hiện trạng

## 2.2 Tổng quan chức năng

## 2.3 Biểu đồ Use-case tổng quát

## 2.4 Biểu đồ phân rã

## 2.5 Biểu đồ hoạt động

## 2.6 Đặc tả chức năng

## Yêu cầu phi chức năng

### 2.7.1 Yêu cầu về giao diện

Giao diện với tông màu sáng với màu trắng là chủ đạo kết hợp với màu xanh, đa số là bằng Tiếng Việt. Các thành phần trên giao diện được sắp xếp một cách logic và rõ ràng, giúp người dùng dễ dàng tìm thấy và truy cập vào các chức năng và tính năng mà hệ thống cung cấp.

### 2.7.2 Yêu cầu về bảo mật

Hệ thống cần phải đảm bảo tính bảo mật tuyệt đối cho người dùng. Ví dụ như mật khẩu của người dùng phải được mã hoá. Các request được gửi lên server phải được đính kèm session.

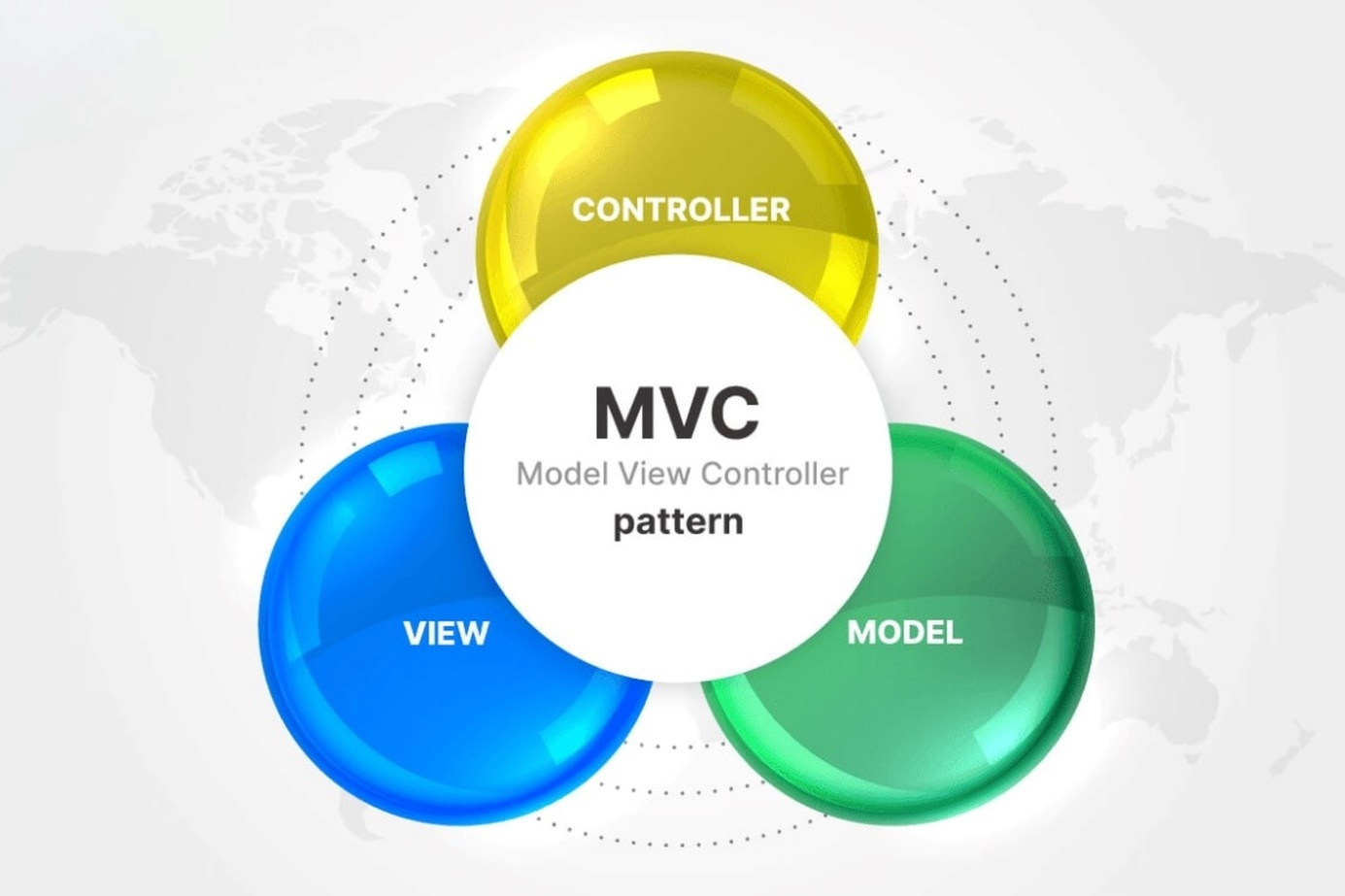
### 2.7.3 Yêu cầu về hiệu năng

Hệ thống được tối ưu để có hiệu năng cao, giúp web chạy ổn định 24/7, đảm bảo trải nghiệm tốt nhất cho người dùng.

## Thiết kế cấu trúc

### 2.8.1 Lựa chọn cấu trúc phần mềm

MVC (Model-View-Controller) là một kiến trúc thiết kế phần mềm phổ biến, giúp tách biệt rõ ràng các thành phần của ứng dụng thành ba phần riêng biệt: Model, View và Controller. Phương pháp này giúp tổ chức mã nguồn theo cách dễ quản lý, bảo trì và mở rộng, đồng thời tăng cường khả năng kiểm thử của ứng dụng [1].



*Hình 2.18: Mô hình MVC*

Hệ thống tuân theo mô hình **MVC**, trong đó:

Model (Mô hình) chịu trách nhiệm quản lý dữ liệu và xử lý các nghiệp vụ của ứng dụng. Tất cả các thao tác liên quan đến truy xuất, lưu trữ và cập nhật dữ liệu đều được thực hiện ở đây. Việc tách riêng phần xử lý dữ liệu không chỉ giúp bảo vệ thông tin mà còn giúp việc thay đổi hoặc mở rộng logic nghiệp vụ diễn ra một cách độc lập, không ảnh hưởng đến giao diện người dùng [1] [4].

View (Giao diện người dùng) đảm nhận vai trò hiển thị thông tin từ Model đến người dùng. Nó giúp người dùng tương tác với hệ thống thông qua các giao diện trực quan và thân thiện, đồng thời tự động cập nhật khi dữ liệu trong Model có sự thay đổi. Điều này giúp tách biệt rõ ràng phần giao diện khỏi logic xử lý, tạo điều kiện cho việc thiết kế và cải tiến giao diện mà không làm ảnh hưởng đến hoạt động của ứng dụng [1] [4].

Controller (Bộ điều khiển) hoạt động như một cầu nối giữa Model và View. Khi người dùng thực hiện các thao tác trên giao diện, Controller sẽ tiếp nhận các yêu cầu này, xử lý chúng và giao tiếp với Model để lấy dữ liệu cần thiết. Sau đó, Controller cập nhật lại thông tin cho View, đảm bảo rằng dữ liệu hiển thị luôn được đồng bộ và chính xác. Qua đó, mô hình MVC giúp xây dựng các ứng dụng có cấu trúc rõ ràng, dễ bảo trì và mở rộng theo thời gian [1] [4].

### 2.8.2 Thiết kế tổng quan

auk hi đã trình bày về mô hình kiến trúc của phần mềm phí trên, em sẽ tiếp tục mô tả thiết kế hệ thống bằng biểu đồ gói.

Ảnh có chứa biểu đồ, Hình chữ nhật, thiết kế, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

*Hình 2.21: Biểu đồ phụ thuộc gói*

Hệ thống bao gồm các gói được tổ chức theo kiến trúc MVC, với những chức năng riêng biệt. Mục đích nhiệm vụ tổng quan từng gói như sau:

* + Controller: chịu trách nhiệm xử lý logic chính.
  + Model: Ánh xạ tới database, kết hợp xử lý dữ liệu với controller.
  + View: Hiển thị giao diện cho người dùng, bắt các sự kiện khi người dùng tương tác trên trang web.

### 2.8.3 Thiết kế chi tiết gói

## 2.9Thiết kế chi tiết

## 2.10Kết luận

Trong chương này, em đã trình bày chi tiết về quá trình khảo sát và phân tích hiện trạng hệ thống, từ đó xác định các yêu cầu cần thiết. Đồng thời, em đã xây dựng sơ đồ use case tổng quát, use case phân rã chức năng và đặc tả chi tiết các biểu đồ, giúp làm rõ mối quan hệ giữa các chức năng trong hệ thống. Những nội dung này không chỉ giúp định hình kiến trúc tổng thể mà còn tạo cơ sở cho các bước triển khai tiếp theo.

Chương tiếp theo sẽ tập trung vào việc giới thiệu các công nghệ, công cụ và phần mềm được sử dụng để thiết kế, phát triển hệ thống, từ đó làm rõ cách thức xây dựng và hiện thực hóa website.

# CHƯƠNG 3. XÂY DỰNG WEBSITE

Trong chương 2, em đã trình bày chi tiết về các công nghệ và thư viện được sử dụng để phát triển hệ thống, đồng thời nêu rõ lý do lựa chọn từng công nghệ. Những nội dung này giúp làm rõ nền tảng kỹ thuật và cách chúng hỗ trợ quá trình xây dựng ứng dụng.

Tiếp theo, trong chương 3, em sẽ tập trung vào **việc** trình bày **kết quả đạt được**, quá trình **kiểm thử hệ thống** để đánh giá hiệu suất và độ ổn định của ứng dụng.

## Kết quả đạt được

## 3.2 Kết luận

Trong chương này em đã trình bày chi tiết về quá trình phát triển ứng dụng, lựa chọn công nghệ, lựa chọn kiến trúc phần mềm, thiết kế cơ sở dữ liệu, thiết kế gói, lớp; giúp mọi người có thể hình dung rõ hơn về quy trình thiết kế một ứng dụng website là như thế nào. Bên cạnh đó, trong chương này em cũng đã trình bày chi tiết về kế hoạch kiểm thử ứng dụng; giúp ta có thể hình dung được ứng dụng sẽ hoạt động ra sao; sinh viên có thể dùng ứng dụng như thế nào cho phù hợp . Trong chương tiếp theo, em sẽ trình bày về những giải pháp và đóng gói nổi bật của khóa luận tốt nghiệp.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận:

Qua quá trình thực hiện dự án “Xây dựng website quản lý khách sạn”, em đã có cơ hội áp dụng và rèn luyện các kỹ năng từ khảo sát nhu cầu, thiết kế giao diện người dùng, lập trình, kiểm thử đến triển khai hệ thống. Dự án đã giúp em củng cố kiến thức chuyên môn, đồng thời rèn luyện khả năng tự giải quyết vấn đề, sáng tạo và quản lý thời gian hiệu quả. Em tin rằng website của em đã đáp ứng được mục tiêu cung cấp một công cụ quản lý khách sạn tiện lợi, nhanh chóng và thân thiện, hỗ trợ khách hàng trong việc tra cứu thông tin phòng, đặt phòng và sử dụng các dịch vụ kèm theo như đặt đồ ăn, đồng thời giúp khách sạn quản lý thông tin, đơn đặt phòng và doanh thu một cách hiệu quả. Em mong nhận được những góp ý, chỉ dẫn từ thầy/cô để dự án ngày càng hoàn thiện và phát triển hơn nữa.

## Hướng phát triển:

* **Nâng cao chức năng tìm kiếm:** Em sẽ tích hợp các bộ lọc nâng cao theo mức giá, loại phòng, tiện nghi và vị trí khách sạn, giúp khách hàng có thể tìm kiếm thông tin phòng một cách chính xác và nhanh chóng hơn.
* **Tích hợp bản đồ và định vị:** Phát triển chức năng xem bản đồ chi tiết và định vị tự động vị trí của khách hàng, từ đó đưa ra các gợi ý phòng và khách sạn phù hợp theo khu vực, nâng cao trải nghiệm người dùng.
* **Phát triển ứng dụng di động:** Em có kế hoạch triển khai phiên bản mobile cho website, giúp khách hàng dễ dàng tra cứu thông tin và đặt phòng trực tiếp qua điện thoại, tạo sự tiện lợi tối đa trong quá trình di chuyển.
* **Mở rộng dữ liệu và hợp tác:** Em sẽ tìm kiếm cơ hội hợp tác với các chuỗi khách sạn, đại lý và các trang chuyên ngành để liên tục cập nhật dữ liệu và mở rộng phạm vi cung cấp phòng cũng như dịch vụ kèm theo, từ đó tạo nền tảng vững chắc cho hệ thống.
* **Tăng cường bảo mật và hiệu năng:** Đầu tư vào các giải pháp bảo mật tiên tiến nhằm đảm bảo an toàn thông tin khách hàng và tối ưu hóa hệ thống, đáp ứng lưu lượng truy cập ngày càng tăng.

Những hướng phát triển trên sẽ giúp dự án nâng cao trải nghiệm khách hàng, mở rộng quy mô và tạo nền tảng vững chắc cho sự phát triển lâu dài của website quản lý khách sạn. Em mong nhận được những góp ý, chỉ dẫn từ thầy/cô để có thể hoàn thiện dự án hơn nữa trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] T.S. Huynh, D.T. Tran, Q.H. Vu, and L.A.T. Nguyen, "Design and Implementation of Web Application Based on MVC Laravel Architecture," \*EJECE, European Journal of Electrical Engineering and Computer Science\*, vol. 6, no. 4, pp. 23–29, Aug. 2022.

[2] Viblo Community, "Tất tần tật về mô hình MVC," Viblo, Aug.10, 2023. [Online]. Available: https://viblo.asia/p/tat-tan-tat-ve-mo-hinh-mvc-Rk74avjAJeO. [Accessed: Feb.3, 2025].

[3] Laravel Documentation, "Blade Templates," Laravel, 2024. [Online]. Available: https://laravel.com/docs/10.x/blade. [Accessed: Feb. 3, 2025].

[4] W3Schools - Hướng dẫn về PHP, MySQL, Bootstrap, và JavaScript: [https://www.w3schools.com](https://www.w3schools.com/)

[5] Tài liệu chính thức Bootstrap: [https://getbootstrap.com](https://getbootstrap.com/)