TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PHẠM ĐỨC MINH HIẾU – 52100796**

**PHAN HOÀNG PHÚ – 52100086**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG DẠY VÀ HỌC TRỰC TUYẾN ĐA NỀN TẢNG VỚI FLUTTER**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KỸ THUẬT PHẦN MỀM**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PHẠM ĐỨC MINH HIẾU – 52100796**

**PHAN HOÀNG PHÚ – 52100086**

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG DẠY VÀ HỌC TRỰC TUYẾN ĐA NỀN TẢNG VỚI FLUTTER**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KỸ THUẬT PHẦN MỀM**

Người hướng dẫn

**ThS. Mai Văn Mạnh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin được phép cảm ơn quý thầy cô khoa Công nghệ Thông Tin trường Đại học Tôn Đức Thắng đã tạo mọi điều kiện cho chúng em tạo mọi điều kiện tốt nhất để được thực hiện môn học Dự án công nghệ thông tin.

Chúng em cũng xin được gửi lời cảm ơn đến ThS Mai Văn Mạnh, thầy đã nhiệt tình giảng dạy, trang bị đầy đủ kiến thức để em có thể hoàn thành môn dự án công nghệ thông tin này.

Cuối cùng, do hạn chế về mặt kiến thức, kính mong thầy cô có thể bỏ qua những sai sót nhỏ và chỉ ra được những lỗi sai của chúng em trong bài báo cáo này để những bài báo cáo sau của chúng em được hoàn thiện hơn.

Một lần nữa chúng em xin chân thành cảm ơn thầy và toàn thể quý thầy cô khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng.

*TP. Hồ Chí Minh, ngày 20 tháng 07 năm 2024*

*Tác giả*

*Phạm Đức Minh Hiếu*

*Phan Hoàng Phú*

**PHIẾU ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tên giảng viên hướng dẫn:

Ý kiến nhận xét:

Điểm tổng theo phiếu đánh giá rubrik:

*TP. Hồ Chí Minh, ngày tháng năm 20*

*Giảng viên hướng dẫn*

*(Ký tên và ghi rõ họ tên)*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

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

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

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

*TP. Hồ Chí Minh, ngày 20 tháng 07 năm 2024*

*Tác giả*

*Phạm Đức Minh Hiếu*

*Phan Hoàng Phú*

**PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG DẠY VÀ HỌC TRỰC TUYẾN ĐA NỀN TẢNG VỚI FLUTTER**

**TÓM TẮT**

Nội dung của bài báo cáo này bao gồm 6 chương:

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vi](#_Toc168246901)

[DANH MỤC BẢNG BIỂU vii](#_Toc168246902)

[DANH MỤC CÁC CHỮ VIẾT TẮT viii](#_Toc168246903)

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc168246904)

[1.1 Lý do chọn đề tài 1](#_Toc168246905)

[1.2 Mục tiêu thực hiện đề tài 1](#_Toc168246906)

[1.3 Đối tượng và phạm vi nghiên cứu 1](#_Toc168246907)

[1.4 Phương pháp nghiên cứu 1](#_Toc168246908)

[1.5 Ý nghĩa thực tiễn của đề tài 1](#_Toc168246909)

[CHƯƠNG 2. TỔNG QUAN 2](#_Toc168246910)

[2.1 Giới thiệu 2](#_Toc168246911)

[2.1.1 Vấn đề 2](#_Toc168246912)

[CHƯƠNG 3. CƠ SỞ LÝ THUYẾT 3](#_Toc168246913)

[CHƯƠNG 4. PHƯƠNG PHÁP NGHIÊN CỨU 4](#_Toc168246914)

[4.1 4](#_Toc168246915)

[CHƯƠNG 5. PHÂN TÍCH DỮ LIỆU 5](#_Toc168246916)

[5.1 5](#_Toc168246917)

[CHƯƠNG 6. KẾT LUẬN 6](#_Toc168246918)

[6.1 6](#_Toc168246919)

[TÀI LIỆU THAM KHẢO 7](#_Toc168246920)

# DANH MỤC HÌNH VẼ

# DANH MỤC BẢNG BIỂU

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| BUS | Business Layer |
|  |  |

# MỞ ĐẦU

## Lý do chọn đề tài

Trong bối cảnh cách mạng công nghiệp 4.0, công nghệ số đang phát triển mạnh mẽ và thay đổi nhiều lĩnh vực, trong đó có giáo dục. Học trực tuyến đã trở thành một phương pháp học tập phổ biến và cần thiết, đặc biệt dưới thời kỳ đại dịch COVID-19 khi việc giãn cách xã hội trở thành yêu cầu bắt buộc. Học trực tuyến giúp tiết kiệm thời gian, chi phí và tạo điều kiện học tập linh hoạt cho người học.

Ngày nay, thiết bị di động như điện thoại thông minh và máy tính bảng đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày. Việc phát triển ứng dụng học trực tuyến trên các nền tảng di động sẽ giúp người học tiếp cận kiến thức mọi lúc, mọi nơi, từ đó nâng cao hiệu quả học tập.

Với sự đa dạng về hệ điều hành trên các loại điện thoại khác nhau, Flutter - một framework phát triển ứng dụng di động đa nền tảng do Google phát triển ra đời cho phép tạo ra các ứng dụng chất lượng cao trên cả iOS và Android và cả Web từ một mã nguồn duy nhất. Điều này không chỉ tiết kiệm thời gian và chi phí phát triển mà còn đảm bảo tính nhất quán và trải nghiệm người dùng tốt hơn.

Ứng dụng Mindify sẽ được thiết kế để đáp ứng nhu cầu cá nhân hóa học tập của từng người dùng. Tính năng này giúp người học có thể lựa chọn nội dung học tập phù hợp với nhu cầu và trình độ của mình, từ đó tối ưu hóa quá trình học tập. Ngoài ra, Mindify sẽ tích hợp các tính năng như video bài giảng, bài kiểm tra, và tài liệu học tập, giúp người học có thể học tập theo tiến độ của riêng mình và nhận được sự hỗ trợ kịp thời từ giáo viên thông qua các diễn đàn thảo luận và phản hồi cá nhân. Điều này giúp tạo ra một môi trường học tập linh hoạt và hiệu quả.

Việc phát triển ứng dụng học trực tuyến đa nền tảng với Flutter không chỉ đáp ứng nhu cầu học tập hiện tại mà còn bắt kịp xu hướng phát triển giáo dục hiện đại. Mindify hướng tới việc trở thành một công cụ hữu ích, hỗ trợ học tập cho mọi đối tượng người dùng, từ học sinh, sinh viên đến người đi làm.

Với những lý do trên, việc chọn đề tài "Phát triển ứng dụng di động dạy và học trực tuyến đa nền tảng với Flutter" là hết sức cần thiết và có ý nghĩa thực tiễn cao. Ứng dụng Mindify hứa hẹn sẽ mang lại nhiều lợi ích cho người học và góp phần vào sự phát triển của giáo dục trong thời đại công nghệ số.

## Mục tiêu thực hiện đề tài

Đề tài này nhằm phát triển một ứng dụng di động giúp cải thiện hiệu quả học tập trực tuyến cho người dùng, với các mục tiêu cụ thể như sau:

* **Xây dựng ứng dụng di động Mindify**: Phát triển một ứng dụng di động dạy và học trực tuyến đa nền tảng sử dụng Flutter, cung cấp các khóa học trực tuyến với nội dung phong phú và linh hoạt.
* **Tạo môi trường học tập hiệu quả**: Đảm bảo ứng dụng cung cấp các tính năng giúp người học có thể học tập một cách hiệu quả, như video bài giảng, bài kiểm tra, tài liệu học tập, và diễn đàn thảo luận.
* **Cá nhân hóa trải nghiệm học tập**: Thiết kế các tính năng cá nhân hóa để người học có thể lựa chọn nội dung học tập phù hợp với nhu cầu và trình độ của mình.
* **Tiết kiệm chi phí và thời gian phát triển**: Sử dụng Flutter để phát triển ứng dụng đa nền tảng, tiết kiệm chi phí và thời gian so với việc phát triển riêng biệt cho từng hệ điều hành.

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Đối tượng nghiên cứu: Ứng dụng Mindify hướng đến đối tượng người học đa dạng, bao gồm học sinh, sinh viên, và người đi làm có nhu cầu học tập trực tuyến.

### Phạm vi môi trường

Môi trường để phát triển ứng dụng là các hệ điều hành: iOS, Android và Windows.

Phạm vi nghiên cứu của đề tài bao gồm:

* Nghiên cứu về nhu cầu và thói quen học tập trực tuyến của người dùng.
* Phân tích các tính năng cần thiết để xây dựng và quản lý một ứng dụng học trực tuyến hiệu quả.
* Áp dụng Flutter, Firebase, NodeJS và ReactJS để phát triển ứng dụng đa nền tảng, đảm bảo tính khả dụng và hiệu quả.

## Phương pháp nghiên cứu

Phương pháp thu thập dữ liệu:

* **Khảo sát:** Thực hiện khảo sát trên các trang web và ứng dụng học trực tuyến để thu thập thông tin về nhu cầu và thói quen học tập của người dùng.
* **Nghiên cứu:** Phân tích các ứng dụng học trực tuyến hiện có để xác định những tính năng quan trọng và ưu nhược điểm của chúng.

Phương pháp phân tích:

* **Phân tích dữ liệu:** Xử lý và phân tích dữ liệu thu thập được để hiểu rõ nhu cầu và mong muốn của người dùng.
* **Xác định tính năng:** Xác định các tính năng cần thiết và ưu tiên cho việc phát triển ứng dụng Mindify dựa trên kết quả phân tích.

Phương pháp thiết kế và phát triển:

* **Flutte**r: Sử dụng Flutter để thiết kế và phát triển ứng dụng di động đa nền tảng, đảm bảo ứng dụng hoạt động mượt mà trên cả iOS và Android.
* **Firebase**: Sử dụng Firebase làm cơ sở dữ liệu lưu trữ và quản lý người dùng cho ứng dụng, đảm bảo tính bảo mật và khả năng mở rộng.
* **Node.js**: Sử dụng Node.js để xây dựng backend, triển khai các chức năng cần thiết và kết nối với cơ sở dữ liệu.
* **ReactJS**: Sử dụng ReactJS để phát triển trang web quản lý ứng dụng, cho phép quản trị viên quản lý khóa học, người dùng, và các nội dung liên quan.
* **RESTful API**: Áp dụng mô hình RESTful API để kết nối giữa frontend và backend, đảm bảo khả năng mở rộng và dễ dàng bảo trì hệ thống.
* **Thử nghiệm**: Tiến hành thử nghiệm ứng dụng qua nhiều giai đoạn, bao gồm thử nghiệm nội bộ và beta test với người dùng thực tế.
* **Điều chỉnh**: Thu thập phản hồi từ người dùng và điều chỉnh ứng dụng để cải thiện tính năng và trải nghiệm người dùng.

## Ý nghĩa thực tiễn của đề tài

Đề tài phát triển ứng dụng di động dạy và học trực tuyến đa nền tảng với Flutter mang lại nhiều lợi ích thiết thực cho cả người học và ngành giáo dục:

* **Nâng cao hiệu quả học tập**: Ứng dụng Mindify sẽ giúp người học tiếp cận kiến thức một cách linh hoạt và hiệu quả, tối ưu hóa quá trình học tập cá nhân.
* **Tiết kiệm thời gian và chi phí**: Phát triển ứng dụng đa nền tảng với Flutter giúp tiết kiệm thời gian và chi phí so với việc phát triển ứng dụng riêng biệt cho từng hệ điều hành.
* **Góp phần vào sự phát triển giáo dục**: Mindify hướng tới việc trở thành một công cụ hữu ích, hỗ trợ học tập cho mọi đối tượng người dùng, từ học sinh, sinh viên đến người đi làm, góp phần vào sự phát triển của giáo dục trong thời đại công nghệ số.
* **Thúc đẩy ứng dụng công nghệ trong giáo dục**: Đề tài này sẽ khuyến khích việc áp dụng công nghệ hiện đại trong giáo dục, tạo điều kiện cho sự đổi mới và phát triển trong lĩnh vực này.

# CƠ SỞ LÝ THUYẾT

## Flutter

Flutter là một framework do Google phát triển và lần đầu tiên ra mắt vào tháng 5 năm 2017. Flutter cho phép các nhà phát triển xây dựng các ứng dụng native cho cả iOS và Android bằng cách sử dụng một mã nguồn duy nhất. Điều này giúp tiết kiệm thời gian và công sức so với việc phát triển từng ứng dụng riêng lẻ cho từng nền tảng.

Flutter cung cấp khả năng gọi mã native (Swift, Kotlin, Java) từ Dart và ngược lại thông qua Platform Channels. Điều này cho phép tích hợp các tính năng và thư viện native mà Flutter chưa hỗ trợ trực tiếp.

Một số ưu điểm của Flutter:

Hiệu suất cao: Flutter sử dụng Dart biên dịch trực tiếp thành mã máy, giúp ứng dụng chạy nhanh và mượt mà.

Phát triển nhanh chóng: Với tính năng Hot Reload, các nhà phát triển có thể thử nghiệm và xem kết quả ngay lập tức.

Giao diện tùy chỉnh: Flutter cung cấp khả năng tùy chỉnh cao, cho phép tạo ra các giao diện phức tạp và độc đáo.

Hỗ trợ đa nền tảng: Một mã nguồn duy nhất có thể chạy trên cả iOS, Android, web và desktop.

## NodeJS và ExpressJS

### NodeJS

Node.js là một nền tảng mã nguồn mở và đa nền tảng cho phép chạy môi trường runtime JavaScript trên máy chủ. Được xây dựng trên nền tảng JavaScript V8 của Google Chrome, Node.js đã trở thành một công cụ phổ biến để phát triển các ứng dụng web, API và các dịch vụ máy chủ khác.

Ưu điểm của NodeJS:

Hiệu suất cao: Nhờ vào mô hình I/O không chặn và vòng lặp sự kiện, Node.js có thể xử lý nhiều kết nối đồng thời một cách hiệu quả.

Cộng đồng lớn: Node.js có một cộng đồng lớn và phát triển mạnh mẽ, cung cấp nhiều thư viện và tài nguyên hỗ trợ.

JavaScript: Sử dụng JavaScript cho cả phía máy khách và máy chủ giúp giảm thiểu sự phức tạp và tăng cường tính nhất quán.

### ExpressJS

Express.js là một framework web ứng dụng tối giản nhưng mạnh mẽ cho Node.js, cung cấp một tập hợp các công cụ và tính năng để xây dựng các ứng dụng web và API. Được phát triển và duy trì bởi cộng đồng, Express.js là một trong những framework web phổ biến nhất cho Node.js.

Express.js được ra mắt lần đầu vào tháng 11 năm 2010 bởi TJ Holowaychuk và đã trở thành một công cụ không thể thiếu cho các nhà phát triển Node.js. Express.js giúp đơn giản hóa quá trình xây dựng các ứng dụng web và API bằng cách cung cấp một bộ công cụ mạnh mẽ và linh hoạt.

Ưu điểm của ExpressJS:

Đơn giản và linh hoạt: Express.js cung cấp một API đơn giản nhưng mạnh mẽ, cho phép các nhà phát triển xây dựng các ứng dụng web và API một cách nhanh chóng và dễ dàng.

Cộng đồng lớn: Với một cộng đồng lớn và nhiều tài nguyên, các nhà phát triển có thể dễ dàng tìm kiếm sự hỗ trợ và các plugin mở rộng.

Hiệu suất cao: Express.js được tối ưu hóa cho hiệu suất cao, giúp xử lý hàng nghìn yêu cầu đồng thời một cách hiệu quả.

## ReactJS

React là một thư viện JavaScript mã nguồn mở do Facebook phát triển, dùng để tạo giao diện người dùng (UI) cho các ứng dụng web. Ra mắt lần đầu vào năm 2013, React đã nhanh chóng trở thành một trong những thư viện UI phổ biến nhất trên toàn cầu.

Các đặc điểm chính của ReactJS:

Virtual DOM (DOM Ảo): là một trong những cơ chế quan trọng nhất của React. Thay vì thao tác trực tiếp trên DOM thực, React sử dụng một bản sao ảo của DOM để theo dõi các thay đổi. Khi trạng thái của một component thay đổi, React sẽ cập nhật Virtual DOM trước. Sau đó, React so sánh Virtual DOM mới với phiên bản cũ để xác định những thay đổi thực sự và cập nhật DOM thực một cách tối ưu nhất, giảm thiểu số lần thao tác DOM và cải thiện hiệu suất

Components (Thành phần): React xây dựng giao diện người dùng dưới dạng các component. Mỗi component là một khối xây dựng độc lập và tái sử dụng được, có thể chứa logic riêng, trạng thái riêng và có khả năng nhận dữ liệu thông qua props

Hooks: là một tính năng mới được giới thiệu trong React 16.8, cho phép sử dụng state và các tính năng khác của React trong function components. Các hook phổ biến bao gồm:

useState(): Quản lý state trong function component.

useEffect(): Quản lý các side effect như gọi API, đăng ký sự kiện, và dọn dẹp.

useContext(): Sử dụng context để quản lý dữ liệu toàn cục.

## Firebase

Firebase là một nền tảng phát triển ứng dụng di động và web của Google, cung cấp một loạt các dịch vụ mạnh mẽ giúp các nhà phát triển xây dựng, cải thiện và phát triển ứng dụng của họ một cách nhanh chóng và dễ dàng

Firebase cung cấp một loạt các dịch vụ như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ, thông báo đẩy, và nhiều công cụ khác giúp tối ưu hóa quá trình phát triển ứng dụng. Firebase được thiết kế để hoạt động mượt mà với các nền tảng di động và web, giúp các nhà phát triển tập trung vào việc tạo ra trải nghiệm người dùng tuyệt vời mà không phải lo lắng về cơ sở hạ tầng.

### Firebase Authentication

Firebase Authentication cung cấp các phương thức dễ sử dụng để xác thực người dùng trong ứng dụng. Nó hỗ trợ nhiều phương thức xác thực như email/mật khẩu, số điện thoại, và đăng nhập qua các tài khoản mạng xã hội như Google, Facebook, Twitter.

### Cloud Firestore

Cloud Firestore là một cơ sở dữ liệu NoSQL được phát triển bởi Google, là một phần của Firebase. Dưới đây là một số điểm chính về Cloud Firestore:

Realtime Database: Cloud Firestore cung cấp khả năng cập nhật thời gian thực, cho phép dữ liệu được đồng bộ hóa ngay lập tức trên tất cả các thiết bị kết nối.

Scalability: Được thiết kế để mở rộng dễ dàng, từ ứng dụng nhỏ đến các ứng dụng quy mô lớn với hàng triệu người dùng.

Structured Data: Dữ liệu được lưu trữ dưới dạng bộ sưu tập (collections) và tài liệu (documents), giúp việc tổ chức và truy vấn dữ liệu trở nên dễ dàng hơn.

Offline Support: Cloud Firestore hỗ trợ lưu trữ dữ liệu ngoại tuyến, cho phép ứng dụng tiếp tục hoạt động ngay cả khi không có kết nối mạng.

Security: Tích hợp với Firebase Authentication và cung cấp quy tắc bảo mật mạnh mẽ để kiểm soát quyền truy cập vào dữ liệu.

Integration: Dễ dàng tích hợp với các dịch vụ khác của Firebase và Google Cloud, hỗ trợ cả ứng dụng web và di động

## HTTP Streaming Protocol

## Google Cloud Stogre

Google Storage là một dịch vụ lưu trữ đối tượng được cung cấp bởi Google Cloud Platform.



Dưới đây là một số điểm chính về Google Storage:

* Object Storage: Lưu trữ dữ liệu dưới dạng các đối tượng, mỗi đối tượng bao gồm dữ liệu và siêu dữ liệu kèm theo.
* Buckets: Đơn vị lưu trữ cơ bản, chứa nhiều đối tượng và được xác định duy nhất bởi tên. Buckets có thể được cấu hình với các thiết lập về kiểm soát truy cập, vị trí địa lý và chính sách vòng đời.
* Location: Chọn vị trí địa lý nơi dữ liệu sẽ được lưu trữ, bao gồm multi-region, dual-region và region.
* Storage Classes: Các lớp lưu trữ khác nhau phù hợp với nhu cầu truy cập:
* Standard Storage: Cho dữ liệu thường xuyên truy cập.
* Nearline Storage: Cho dữ liệu ít truy cập, khoảng một lần một tháng.
* Coldline Storage: Cho dữ liệu rất ít truy cập, khoảng một lần một năm.
* Archive Storage: Cho dữ liệu gần như không truy cập, nhưng cần lưu trữ lâu dài.
* Lifecycle Management: Chính sách vòng đời tự động quản lý các đối tượng dựa trên tiêu chí như tuổi của đối tượng và lớp lưu trữ hiện tại.
* Security: Cung cấp nhiều cơ chế kiểm soát truy cập, bao gồm IAM, ACLs và quản lý khóa mã hóa.
* Integration: Dễ dàng tích hợp với các dịch vụ khác của Google Cloud, hỗ trợ nhiều loại ứng dụng khác nhau.

## Mô hình RESTful API

RESTful API là một kiểu thiết kế API tuân theo các nguyên tắc của REST. Dưới đây là một số điểm chính về mô hình này:

* Resource-Based: RESTful API dựa trên khái niệm về "resources" (tài nguyên). Mỗi tài nguyên có một URI (Uniform Resource Identifier) duy nhất. Ví dụ: /users, /orders, /products.
* HTTP Methods: RESTful API sử dụng các phương thức HTTP để thực hiện các hành động CRUD (Create, Read, Update, Delete) trên tài nguyên.
* GET: Lấy dữ liệu từ máy chủ.
* POST: Tạo mới một tài nguyên.
* PUT: Cập nhật toàn bộ tài nguyên.
* PATCH: Cập nhật một phần tài nguyên.
* DELETE: Xóa tài nguyên.
* Stateless: Mỗi yêu cầu từ client đến server phải chứa đủ thông tin để server hiểu và xử lý yêu cầu. Server sẽ không lưu trữ trạng thái của client giữa các yêu cầu.
* Representation: Tài nguyên có thể được đại diện ở nhiều định dạng khác nhau, như JSON, XML, HTML. JSON thường được sử dụng phổ biến nhất. Client và Server có thể trao đổi định dạng mong muốn thông qua tiêu đề HTTP Content-Type và Accept.
* Layered System: Kiến trúc của RESTful API có thể được phân thành nhiều lớp, giúp tăng cường bảo mật, cân bằng tải và dễ dàng mở rộng.
* Caching: RESTful API có thể tận dụng cơ chế caching của HTTP để cải thiện hiệu suất. Server có thể chỉ định các header như Cache-Control, ETag để hướng dẫn client về cách lưu trữ tạm thời và làm mới dữ liệu.



# PHƯƠNG PHÁP NGHIÊN CỨU

## 

# PHÂN TÍCH VÀ THIẾT KẾ

## 

# XÂY DỰNG ỨNG DỤNG

## 5

# KẾT LUẬN

# TÀI LIỆU THAM KHẢO

Tiếng Việt

* 1. Hoàng Đinh, “Design Patterns là gì? Tại sao nó lại là trợ thủ đắc lực của Developers - Tổng hợp 23 mẫu Design Pattern”, Viblo, <https://viblo.asia/p/design-patterns-la-gi-tai-sao-no-lai-la-tro-thu-dac-luc-cua-developers-tong-hop-23-mau-design-pattern-GrLZDBQV5k0>.
  2. Giang Phan, “Tổng hợp các bài hướng dẫn về Design Pattern - 23 mẫu cơ bản của GoF”, Viblo, <https://viblo.asia/p/tong-hop-cac-bai-huong-dan-ve-design-pattern-23-mau-co-ban-cua-gof-3P0lPQPG5ox>.

Tiếng Anh

* 1. Refactoring Guru, “Design Pattern”, Refactoring Guru, <https://refactoring.guru/design-patterns/>.
  2. GeeksForGeeks, “Software Design Patterns Tutorial”, GeeksForGeeks, <https://www.geeksforgeeks.org/software-design-patterns/>.
  3. Source Making, “Design Patterns”, Source Making, <https://sourcemaking.com/design_patterns>.