

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

NGUYỄN HUY QUANG

XÂY DỰNG ỨNG DỤNG HỌC TRỰC TUYẾN
VỚI FLUTTER

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC
NGÀNH KỸ THUẬT PHẦN MỀM

THÁI NGUYÊN, NĂM 2024

ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

NGÀNH KỸ THUẬT PHẦN MỀM

Đề tài:

**XÂY DỰNG ỨNG DỤNG HỌC TRỰC TUYẾN
VỚI FLUTTER**

Sinh viên thực hiện: NGUYỄN HUY QUANG

Lớp: KTPM K18A

Giáo viên hướng dẫn: ThS. ĐÀO THỊ THU

THÁI NGUYÊN, NĂM 2024

MỤC LỤC

DANH MỤC HÌNH ẢNH.....	5
LỜI MỞ ĐẦU	7
CHƯƠNG I: CƠ SỞ LÝ THUYẾT	9
1.1 Tổng quan về ngôn ngữ Dart và Flutter	9
1.1.1 Tổng quan về Dart.....	9
1.1.2 Một số đặc điểm về ngôn ngữ Dart.....	10
1.1.3 Cú pháp khai báo.....	10
1.1.4 Tổng quan về Flutter	11
1.1.5 Lịch sử phát triển của Flutter	11
1.1.6 Ưu nhược điểm của Flutter	12
1.2 Tổng quan về PHP và Laravel.....	13
1.2.1 Tổng quan về ngôn ngữ PHP	13
1.2.2 Cú pháp khai báo.....	14
1.2.3 Tổng quan về Laravel	17
1.2.4 Ưu điểm và nhược điểm của Laravel.....	18
1.3 Hệ quản trị cơ sở dữ liệu MySQL Workbench và Firebase	19
1.3.1 Hệ quản trị cơ sở dữ liệu MySQL Workbench	19
1.3.2 Giới thiệu về FireBase	21
CHƯƠNG II: KHẢO SÁT VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG	25
2.1. Khảo sát hệ thống	25
2.1.1. Thực trạng.....	25
2.1.2. Mô tả hệ thống.....	26
2.1.3. Phân tích yêu cầu	27
2.1.4. Đánh giá các hệ thống hiện tại.....	28
2.1.5. Đề xuất hệ thống mới.....	28
2.1.6. Mục tiêu	29
2.2. Phân tích hệ thống	30
2.2.1 Danh sách các tác nhân	30
2.2.2 Biểu đồ Use case tổng quát.....	31
2.2.3 Biểu đồ Use case phân rã.....	32

2.2.4	Biểu đồ hoạt động	34
2.2.5	Biểu đồ trình tự	39
2.2.6	Biểu đồ lớp	40
2.2.7.	Thiết kế cơ sở dữ liệu	41
CHƯƠNG III: CÀI ĐẶT CHƯƠNG TRÌNH		49
3.1	Giao diện phía người dùng	49
3.1.1	Giao diện trang đăng nhập	49
3.1.2	Giao diện trang đăng ký	50
3.1.3	Giao diện trang quên mật khẩu	51
3.1.4	Giao diện trang chủ	52
3.1.5	Giao diện trang xem chi tiết thông tin khoá học	53
3.1.6	Giao diện trang thanh toán	54
3.1.7	Giao diện trang xem video khoá học	55
3.2	Giao diện phía backend	58
3.2.1	Trang giao diện quản lý khoá học	58
3.2.2	Trang giao diện thêm khoá học	58
3.2.3	Trang giao diện sửa thông tin khoá học	59
3.2.4	Trang quản lý tài khoản	59
3.2.5	Trang phân quyền	60
KẾT LUẬN		61
TÀI LIỆU THAM KHẢO		63

DANH MỤC HÌNH ẢNH

Hình 1.1: Cách thức hoạt động của MySQL	20
Hình 1.2: Giới thiệu về Firebase	22
Hình 2.1: Biểu đồ use case tổng quát	31
Hình 2.2: Biểu đồ Use case phân rã của chức năng Đăng nhập	32
Hình 2.3: Biểu đồ Use case phân rã của chức năng tìm kiếm	33
Hình 2.4: Biểu đồ Use case phân rã của chức năng quản lý thông tin cá nhân	33
Hình 2.5: Biểu đồ Use case phân rã của chức năng quản lý tài khoản	34
Hình 2.6: Biểu đồ hoạt động chức năng đăng ký	34
Hình 2.8: Biểu đồ hoạt động chức năng thanh toán	37
Hình 2.9: Biểu đồ hoạt động chức năng tìm kiếm	38
Hình 2.10: Biểu đồ trình tự chức năng đăng nhập	39
Hình 2.11: Biểu đồ trình tự chức năng đăng ký	39
Hình 2.12: Biểu đồ trình tự chức năng tìm kiếm	40
Hình 2.13: Biểu đồ lớp	40
Hình 2.14: Cấu trúc các bảng cơ sở dữ liệu	41
Hình 2.15: bảng admin_user	41
Hình 2.16: Bảng admin_operation_log	42
Hình 2.17: Bảng admin_permissions	42
Hình 2.19: Bảng admin_role_menu	42
Hình 2.20: Bảng admin_role_permissions	43
Hình 2.21: Bảng admin_role_users	43
Hình 2.22: Bảng admin_roles	43
Hình 2.23: Bảng admin_user_permissions	44
Hình 2.24: Bảng admin_users	44
Hình 2.25: Bảng article_types	44
Hình 2.26: Bảng articles	45
Hình 2.27: Bảng course_types	45
Hình 2.28: Bảng courses	45
Hình 2.29: Bảng lessons	46
Hình 2.30: Bảng migrations	46

Hình 2.31: Bảng orders.....	46
Hình 2.32: Bảng password_reset_tokens	47
Hình 2.33: Bảng personal_access_tokens	47
Hình 2.34: Bảng user	47
Hình 2.35: Bảng usercall	48
Hình 2.36: Bảng users	48
Hình 3.1: Giao diện trang đăng nhập.....	49
Hình 3.2: Giao diện trang đăng ký	50
Hình 3.3: Giao diện trang quên mật khẩu.....	51
Hình 3.4: Giao diện trang chủ	52
Hình 3.5: Giao diện trang chi tiết khoá học.....	53
Hình 3.6: Giao diện trang thanh toán	54
Hình 3.7: Giao diện trang xem video khoá học.....	55
Hình 3.8: Giao diện trang trò chuyện	56
Hình 3.9: Giao diện trang quản lý thông tin cá nhân	57
Hình 3.10: Giao diện trang quản lý khoá học.....	58
Hình 3.11: Giao diện trang thêm khoá học.....	58
Hình 3.12: Giao diện trang sửa khoá học	59
Hình 3.13: Giao diện trang quản lý tài khoản	59
Hình 3.14: Giao diện trang phân quyền.....	60

LỜI MỞ ĐẦU

1. Lý do chọn đề tài

Trong thời đại mà công nghệ thông tin ngày càng tiến bộ và trở thành một phần không thể tách rời của cuộc sống hàng ngày, việc phát triển ứng dụng di động đã trở thành một lĩnh vực vô cùng hấp dẫn và phát triển mạnh mẽ. Sự lên ngôi của các nền tảng di động như Android và iOS đã mở ra một cánh cửa rộng lớn cho các nhà phát triển ứng dụng. Trong tương lai, việc tạo ra ứng dụng di động chất lượng và hiệu quả sẽ trở thành một mối quan tâm quan trọng hơn bao giờ hết.

Trong bối cảnh này, Flutter - một framework xây dựng ứng dụng di động do Google phát triển - cho phép việc xây dựng các ứng dụng di động đa nền tảng một cách linh hoạt. Flutter cung cấp một cách tiếp cận hiệu quả cho việc phát triển ứng dụng, giảm bớt sự phức tạp trong việc xây dựng và duy trì ứng dụng trên nhiều nền tảng khác nhau. Đặc biệt, Flutter được đánh giá cao về khả năng tạo ra giao diện người dùng đẹp mắt và tương tác mượt mà, mang lại trải nghiệm người dùng tốt nhất có thể.

Trong bài báo cáo này, em sẽ tập trung vào việc nghiên cứu và áp dụng Flutter trong một lĩnh vực cụ thể: Xây dựng ứng dụng học trực tuyến với Flutter. Giáo dục trực tuyến đã và đang trở thành một xu thế ngày càng phổ biến, với hàng triệu người trên khắp thế giới đang tìm kiếm cơ hội học hỏi và phát triển bản thân thông qua các khóa học trực tuyến. Việc sử dụng Flutter để phát triển ứng dụng bán học trực tuyến không chỉ mang lại một trải nghiệm người dùng tốt, mà còn giúp tối ưu hóa quá trình phát triển và triển khai ứng dụng.

Chúng ta sẽ đi sâu vào các khía cạnh cụ thể của việc phát triển ứng dụng học trực tuyến với Flutter, bao gồm cách thiết kế giao diện người dùng, quản lý dữ liệu, xử lý thanh toán và tích hợp các tính năng nâng cao khác. Bằng cách tận dụng các tính năng và công cụ mạnh mẽ của Flutter, em hy vọng có thể cung cấp một cái nhìn tổng quan và hữu ích về cách áp dụng nền tảng này trong lĩnh vực giáo dục và đào tạo.

2. Mục đích nghiên cứu

2.1. Về lý thuyết

- Hiểu được và biết sử dụng các ngôn ngữ lập trình có các thuộc tính, phương thức, đối tượng nào.
- Biết phân tích thiết kế hệ thống và dùng công cụ để thiết kế.

- Biết sử dụng hệ quản trị cơ sở dữ liệu trong CSDL.
- Thiết kế giao diện và xây dựng ứng dụng (Flutter).

2.2. Về nghiên cứu thực tiễn

Ứng dụng học trực tuyến với mục đích chính là tạo điều kiện tiếp cận giáo dục dễ dàng và linh hoạt hơn. Người dùng có thể học từ mọi nơi, vào bất kỳ thời gian nào, và tùy chỉnh theo nhu cầu cá nhân, giúp khuyến khích học tập suốt đời. Đối với giáo viên và tổ chức giáo dục, ứng dụng trực tuyến cung cấp công cụ quản lý và giảng dạy hiệu quả, đồng thời giảm chi phí cơ sở vật chất.

Ngoài ra, các doanh nghiệp có thể xem ứng dụng học trực tuyến như một cơ hội kinh doanh, mở ra nguồn doanh thu từ các khóa học và nội dung giáo dục. Như vậy, ứng dụng học trực tuyến đóng góp vào việc mở rộng cơ hội giáo dục và tạo ra môi trường học tập đa dạng, phù hợp với nhiều đối tượng khác nhau.

3. Kết cấu đề tài

Đề án của em gồm có ba phần:

Chương I: Cơ sở lý thuyết

Chương II: Khảo sát và phân tích thiết kế hệ thống

Chương III: Cài đặt chương trình

CHƯƠNG I: CƠ SỞ LÝ THUYẾT

1.1 Tổng quan về ngôn ngữ Dart và Flutter

1.1.1 Tổng quan về Dart

Dart là ngôn ngữ lập trình đa mục đích mã nguồn mở được phát triển bởi Google, được thiết kế để xây dựng ứng dụng web, di động, máy tính và Internet of Things (IoT).

Ngôn ngữ Dart được phát triển bởi Google và đã trải qua một số giai đoạn quan trọng trong lịch sử của mình. Dưới đây là một tóm tắt về lịch sử ra đời của Dart:

- Ngày 10 tháng 10 năm 2011: Google công bố Dart chính thức tại sự kiện GOTO Conference ở Aarhus, Đan Mạch. Dart được giới thiệu như là một ngôn ngữ lập trình mới có khả năng xây dựng ứng dụng web hiệu quả và đáng tin cậy hơn.
- Giai đoạn đầu (2011-2013): Dart ra đời với sự hỗ trợ tích hợp cho trình duyệt web Dartium, một phiên bản của trình duyệt Chrome có sẵn để chạy mã Dart mà không cần biên dịch thành JavaScript. Google tiếp tục nghiên cứu và phát triển Dart như là một lựa chọn cho việc phát triển các ứng dụng web phức tạp.
- Dart 1.0 (Ngày 14 tháng 11 năm 2013): Dart 1.0 được phát hành, đánh dấu sự ổn định và chính thức của ngôn ngữ này. Trong giai đoạn này, Google tập trung vào việc tối ưu hóa hiệu suất và cải thiện sự tương thích với các trình duyệt không hỗ trợ Dart natively.
- Hướng tới Web và Flutter (2015-2018): Dart bắt đầu chuyển hướng về việc hỗ trợ xây dựng ứng dụng di động với sự xuất hiện của Flutter vào năm 2017. Flutter, một framework UI đa nền tảng, sử dụng Dart làm ngôn ngữ lập trình chính. Flutter giúp xây dựng ứng dụng di động đa nền tảng với giao diện người dùng đẹp mắt và hiệu suất cao.
- Dart 2 (Ngày 7 tháng 8 năm 2018): Dart 2 được giới thiệu với nhiều cải tiến, bao gồm cú pháp mới, kiểu dữ liệu có tên, và sự hỗ trợ cho JIT (Just-In-Time) và AOT (Ahead-Of-Time) compilation. Dart 2 là phiên bản Dart chủ đạo và đặt nền tảng cho việc phát triển ứng dụng Flutter.
- Ngày nay: Dart và Flutter đang trở nên ngày càng phổ biến trong phát triển ứng dụng di động và web. Google tiếp tục duy trì và cập nhật Dart để đáp ứng nhu cầu của cộng đồng và để nó trở thành một lựa chọn tối ưu cho việc phát triển ứng dụng chạy trên nhiều nền tảng khác nhau.

1.1.2 Một số đặc điểm về ngôn ngữ Dart

- Đa nền tảng: Dart hỗ trợ phát triển đa nền tảng, nơi bạn có thể xây dựng ứng dụng chạy trên nhiều loại thiết bị, bao gồm web, di động (sử dụng Flutter), máy tính và IoT.
- Tính nhất quán và Dễ đọc: Dart có cú pháp rành mạch, rõ ràng và dễ đọc, giúp những người mới học nhanh chóng làm quen, thích nghi với ngôn ngữ này. Nó có một số đặc điểm từ ngôn ngữ khác như JavaScript, Java và C#, giúp làm quen với những người lập trình có kinh nghiệm.
- Kiểu dữ liệu và Kiểu an toàn: Dart là một ngôn ngữ kiểu dữ liệu tĩnh, điều này tức là bạn cần phải xác định kiểu dữ liệu của biến trước khi sử dụng. Dart cũng hỗ trợ kiểu an toàn, giúp kiểm tra lỗi và giúp phát hiện các lỗi trước khi chạy chương trình.
- Garbage Collection: Dart có garbage collector, giúp quản lý bộ nhớ tự động và giảm bớt ta quan tâm về việc quản lý bộ nhớ thủ công.
- Package Management: Dart sử dụng quản lý gói Pub để quản lý và chia sẻ các thư viện và gói mã nguồn mở.
- Asynchronous Programming: Dart hỗ trợ lập trình không đồng bộ với Future và Stream, giúp xử lý tác vụ đồng thời một cách hiệu quả. Dart là một ngôn ngữ đa nhiệm, hiện đại và linh hoạt, đặc biệt phù hợp cho phát triển ứng dụng di động và web, đặc biệt là khi sử dụng cùng với Flutter.

1.1.3 Cú pháp khai báo

- Biến (Variables): Biến trong Dart được sử dụng để lưu trữ dữ liệu. Một biến có thể được khai báo bằng từ khóa var hoặc cụ thể với kiểu dữ liệu như int, double, String, và nhiều kiểu khác.

```
var age = 25; // Biến age kiểu integer
var name = 'John'; // Biến name kiểu string
```

- Hằng (Constants): Hằng là giá trị không thay đổi trong quá trình chạy của chương trình. Dart cung cấp từ khóa final hoặc const để khai báo hằng.

```
final pi = 3.14; // Hằng pi có giá trị không thay đổi
const appName = 'MyApp'; // Hằng appName có giá trị không thay đổi
```