

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN GIỮA KÌ MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG

SPRING BOOT FRAMEWORK ĐỂ PHÁT TRIỂN ỨNG DỤNG WEB TRÊN NỀN TẢNG JAVA

Người hướng dẫn: **GV. NGUYỄN THÁI DUY**

Người thực hiện: **TRẦN THỊ VỆ - 52100674**

NGUYỄN ANH TÚ - 52100138

PHAN CÔNG THỊNH - 52100113

Khoá : 25

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN GIỮA KÌ MÔN LẬP TRÌNH WEB VÀ ỨNG DỤNG

SPRING BOOT FRAMEWORK ĐỂ PHÁT TRIỂN ỨNG DỤNG WEB TRÊN NỀN TẢNG JAVA

Người hướng dẫn: **GV. NGUYỄN THÁI DUY**

Người thực hiện: **TRẦN THỊ VỆ - 52100674**

NGUYỄN ANH TÚ – 52100138

PHAN CÔNG THỊNH - 52100113

Khoá : 25

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Trong suốt quá trình học tập và rèn luyện, chúng em đã nhận được rất nhiều sự giúp đỡ tận tình, sự quan tâm, chăm sóc của GV. Ngoài ra, chúng em còn được GV truyền đạt những kiến thức, phương pháp mới về toán hay ho và thú vị, thầy cô còn giúp sinh viên có được nhiều niềm vui trong việc học và cảm thấy thoải mái, ... Chúng em xin chân thành cảm ơn các thầy cô rất nhiều trong suốt quá trình học tập này!

Bởi lượng kiến thức của chúng em còn hạn hẹp và gặp nhiều vấn đề trong quá trình học nên báo cáo này sẽ còn nhiều thiếu sót và cần được học hỏi thêm. Chúng em rất mong em sẽ nhận được sự góp ý của quý thầy cô về bài báo cáo này để chúng em rút kinh nghiệm trong những môn học sắp tới. Cuối cùng, chúng em xin chân thành cảm ơn quý thầy cô.

TP Hồ Chí Minh, ngày 15 tháng 04 năm 2023

Sinh viên:

Trần Thị Vẹn – 52100674

Nguyễn Anh Tú – 52100138

Phan Công Thịnh - 52100113

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của GV Doãn Xuân Thanh;. 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 đồ 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 tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Trần Thị Vẹn

Nguyễn Anh Tú

Phan Công Thịnh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Như chúng ta được biết, các ứng dụng hay phần mềm hiện nay xuất hiện càng nhiều và phát triển mạnh mẽ. Việc sử dụng Spring Boot để phát triển ứng dụng web trên nền tảng Java cũng là một hình thức phát triển phần mềm phổ biến.

Bài tiểu luận này sẽ giúp chúng ta hiểu rõ hơn về framework này và những lợi ích mà nó mang lại trong việc phát triển ứng dụng Java. Chúng ta sẽ được tìm hiểu về những tính năng tự động cấu hình, cung cấp các thư viện hỗ trợ và cấu hình tối giản của Spring Boot, giúp cho việc xây dựng ứng dụng trở nên dễ dàng và nhanh chóng hơn. Bên cạnh đó, chúng ta cũng đã tìm hiểu về tích hợp của Spring Boot với nhiều công nghệ khác như Spring Data, Spring Security và Spring Cloud, giúp cho việc xây dựng các ứng dụng phức tạp trở nên dễ dàng hơn. Chúng ta cũng sẽ được tìm hiểu về các cách triển khai ứng dụng sử dụng Spring Boot và sự hỗ trợ từ cộng đồng phát triển rất lớn và tốt của Spring Boot.

Ngoài ra để phát triển được ứng dụng cho bài báo cáo này cần vững về những kiến thức cơ bản về HTML, CSS, JavaScript để xây dựng giao diện người dùng cho ứng dụng. Để kết hợp với framework Spring Boot hoàn thành các chức năng của demo.

MỤC LỤC

TÓM TẮT	iv
MỤC LỤC	v
DANH MỤC HÌNH VẼ	vi
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ SPRING BOOT	1
1.1 Tổng quan về Spring boot	1
1.1.1 Lịch sử hình thành	1
1.1.2 Khái niệm	1
1.2 Ưu và nhược điểm của việc sử dụng Spring Boot	2
1.3 Một số tính năng rất quan trọng của Spring Boot	3
1.4 Công cụ để quản lý các phụ thuộc và build dự án	4
CHƯƠNG 2: CẤU TRÚC CỦA MỘT DỰ ÁN SPRING BOOT	5
1.1 Cấu trúc dự án Spring Boot	5
1.1.1 Cấu trúc chung của ứng dụng	5
1.1.2 Chi tiết cấu trúc source code	5
1.2 Tổ chức source code	6
1.2.1 Tổ chức source code theo mô hình 3 lớp	6
1.2.2 Tổ chức code theo tính năng	7
1.3 Cấu hình của dự án Spring Boot	8
1.4 Cài đặt Spring Tool Suite Cho Eclipse	9
1.1 Giới thiệu ứng dụng cần demo	13
1.2 Hình ảnh trang web và kết quả demo	15
TÀI LIỆU THAM KHẢO	I

DANH MỤC HÌNH VẼ

Hình 1: Cấu trúc source code.....	5
Hình 2: Mô hình 3 lớp MVC	6
Hình 3: Tổ chức source code theo tính năng	7
Hình 4: IDE Spring Tool Suite base	9
Hình 5: Truy cập Eclipse Marketplace	9
Hình 6: Install Spring Tools	10
Hình 7: Confirm để installed.....	10
Hình 8: Chọn accept và finish	11
Hình 9: Hiện thị các tính năng của Spring	11
Hình 10: Chọn Ok	12
Hình 11: Bạn sẽ thấy biểu tượng Spring ở góc phải của Eclipse	12
Hình 12: Tạo Spring Project	13
Hình 13: Các file html, css, img, js của trang web	13
Hình 14: Tổ chức source code theo tính năng đã trình bày ở trên.....	14
Hình 15: Database kèm theo các config	14
Hình 16: Các file cần đổi root để chạy được trang web	15
Hình 17: Giao diện của trang web	15
Hình 18: Giao diện của các sản phẩm	15
Hình 19: Giao diện login.....	16
Hình 20: Login thành công và có thể add sản phẩm	16
Hình 21: Thêm sản phẩm thành công và được localhost thông báo	17
Hình 21: Add vào giỏ có hiện số sản phẩm thành công	17

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN VỀ SPRING BOOT

1.1 Tổng quan về Spring boot

1.1.1 Lịch sử hình thành

Spring Framework là một trong những khung ứng dụng dựa trên Java phổ biến nhất. Spring Framework được Rod Johnson phát triển vào năm 2003. Spring Framework là một khung mã nguồn mở có thể được sử dụng để phát triển các ứng dụng Java một cách dễ dàng và với tốc độ nhanh chóng. Nó là một khung công tác nhẹ cũng cung cấp hỗ trợ cơ sở hạ tầng được xác định rõ ràng để phát triển các ứng dụng trong Java. Nói cách khác, bạn có thể nói rằng Spring xử lý cơ sở hạ tầng để bạn có thể tập trung hơn vào việc phát triển ứng dụng của mình.

Theo đà phát triển đó, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

Framework Spring là framework luôn nằm trong top 15 framework được các lập trình viên ưa thích nhất. Tuy nhiên, bản thân Framework này vẫn tồn đọng nhiều khuyết điểm và cần được khắc phục. Đó chính là lý do mà Pivotal cho ra đời Spring Boot - một phiên bản cải tiến hơn của framework Spring.

Mặc dù Spring là một Framework tuyệt vời để phát triển phần mềm nhưng nó vẫn tồn tại một số nhược điểm / hạn chế như cấu hình quá nhiều. Xong Spring Boot được xây dựng để khắc phục những nhược điểm đó, đồng thời cũng cung cấp một hướng phát triển phần mềm mới trong tương lai.

1.1.2 Khái niệm

- Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
- Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
- Spring Boot không yêu cầu cấu hình XML

- Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.

1.2 Ưu và nhược điểm của việc sử dụng Spring Boot

Dưới đây là những ưu và nhược điểm của Spring Boot framework quyết định xem công cụ phát triển này có đáp ứng được nhu cầu của người sử dụng hay không:

- **Các ưu điểm nổi bật của Spring Boot là gì?**
 - Sở hữu đầy đủ các tính năng của Spring Framework
 - Tạo ứng dụng một cách độc lập, có thể chạy trên cả nền tảng Java Web
 - Cho phép nhúng trực tiếp các web server như Jetty, Tomcat,... mà không cần phải triển khai các file WAR.
 - Cung cấp nhiều plugin
 - Tối ưu hóa công đoạn cấu hình cho ứng dụng, không sinh ra code cấu hình và nó cũng không yêu cầu người dùng phải cấu hình lại bằng XML. Từ đó, giúp tiết kiệm thời gian viết code và tăng năng suất lao động.
 - Có thể đóng gói ứng dụng Spring dưới dạng là file JAR và có thể dễ dàng khởi động ứng dụng chỉ với một câu lệnh ngắn gọn, quen thuộc: `java - jar,...`
 - Giảm thiểu thời gian phát triển code, tăng hiệu suất phát triển chung của cả dự án.
 - Dễ dàng tích hợp các mô-đun liên quan như Spring-MVC, Spring Data, Spring Security, Spring Cloud,v.v...
 - Nó cung cấp các HTTPs servers như Tomcat, Jetty,.. để phát triển, kiểm thử, deploy một cách dễ dàng. Cung cấp công cụ CLI(Command Line Interface) cho việc phát triển và test ứng dụng nhanh chóng từ command line
 - Ngoài ra còn có nhiều plugins để phát triển nhanh chóng bằng các công cụ như Build như Maven hoặc Gradle.
- **Nhược điểm của Spring Boot**
 - Thiếu kiểm soát. Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.

- Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.
- Không thích hợp cho các dự án quy mô lớn. Hoạt động liên tục với các microservices, theo nhiều nhà phát triển, Spring Boot không phù hợp để xây dựng các ứng dụng nguyên khối.

1.3 Một số tính năng rất quan trọng của Spring Boot

- **Spring Application:** Khi lập trình xong và bạn chỉ muốn chạy thử nghiệm thì nên làm thế nào? Bạn chỉ cần gọi `run()` là được, vì Spring Boot được thiết kế theo dạng “just run”. Giúp cho các lập trình viên chỉ cần cấu hình ít Spring nhất, phần còn lại Spring Boot sẽ lo liệu.
- **Externalized Configuration:** Bạn mong muốn tạo nên một ứng dụng có thể chạy trên nhiều loại môi trường khác nhau? Spring Boot sẽ giúp bạn config cấu hình từ ngoài và ứng dụng của bạn sẽ có thể chạy thoải mái.
- **Profiles:** Nếu có nhiều config khác nhau, bạn có thể sử dụng Profile để phân chia từng loại cho từng môi trường để dễ dàng quản lý hơn.
- **Logging:** Tính năng này được sử dụng cho toàn bộ chức năng log trong phạm vi nội bộ và nó được quản lý mặc định.
- **Dependency Injection:** Tính năng này giúp kiểm soát việc khởi tạo các đối tượng trong ứng dụng Java. Nó giảm thiểu sự phụ thuộc các đối tượng trên nhau, một cách dễ dàng để cấu trúc hóa mã. Spring Boot sử dụng Dependency Injection (DI) để quản lý sự phụ thuộc của các đối tượng thành phần trong ứng dụng của bạn.
- **Actuator:** Tính năng này cung cấp các chức năng quản lý và giám sát ứng dụng bằng cách cung cấp các endpoint RESTful. Actuator cung cấp các API để tùy chỉnh hoạt động của ứng dụng, tạo ra các báo cáo khai thác và quản lý một số khía cạnh của ứng dụng, chẳng hạn như tiến trình, tài nguyên và lỗi.

Bên cạnh những tính năng quan trọng nói trên, còn một số tính năng tương tự khác như: Security, Messaging, Developing web Applications, Working with SQL

Technologies, Caching, Sending Email, Validation, Calling rest Services with RestTemplate/WebClient...

1.4 Công cụ để quản lý các phụ thuộc và build dự án

Có nhiều công cụ hỗ trợ để phát triển ứng dụng Spring Boot: Spring Tool Suite (STS), IntelliJ IDEA, Maven, Gradle, Postman, JUnit và Mockito và Eclipse...

- **Maven:** Apache maven được ra đời vào năm 2004 là một chương trình quản lý dự án cho phép các lập trình viên có thể quản lý về version, các thư viện sử dụng trong dự án, quản lý build,...
- **Gradle:** Gradle là một hệ thống tự động build mã nguồn mở dựa trên các khái niệm về Apache Ant và Apache Maven. Maven mình đã giới thiệu ở trên nên dưới đây mình sẽ giới thiệu qua về Ant.
- **Eclipse** cung cấp các plugin để hỗ trợ phát triển Spring Boot, bao gồm Spring Tools 4, một plugin miễn phí giúp bạn dễ dàng phát triển các ứng dụng Spring Boot trong Eclipse. Ngoài ra, Eclipse cũng có thể sử dụng Maven và Gradle để quản lý dự án và thư viện.

So sánh Gradle và Maven:

- Gradle dựa vào đồ thị phụ thuộc task. Trong khi đó Maven thì lại dựa trên mô hình cố định và tuyến tính của các giai đoạn
- Với Maven, các mục tiêu gắn liền với các giai đoạn của dự án và các mục tiêu có chức năng tương tự như các task của Gradle.
- Xét về hiệu suất làm việc thì cả hai đều cho phép các build module được chạy song song. Tuy nhiên, Gradle cho phép tăng khả năng build vì nó kiểm tra tác vụ nào được cập nhật hay không nên thời gian build của Gradle sẽ nhanh hơn so với Maven.
- Các tính năng khác biệt về mặt hiệu suất mà bạn có thể tìm thấy trên Gradle bao gồm:
 - Tăng khả năng compile cho các class Java
 - Sử dụng API cho việc tăng số lượng các task con

CHƯƠNG 2: CẤU TRÚC CỦA MỘT DỰ ÁN SPRING BOOT

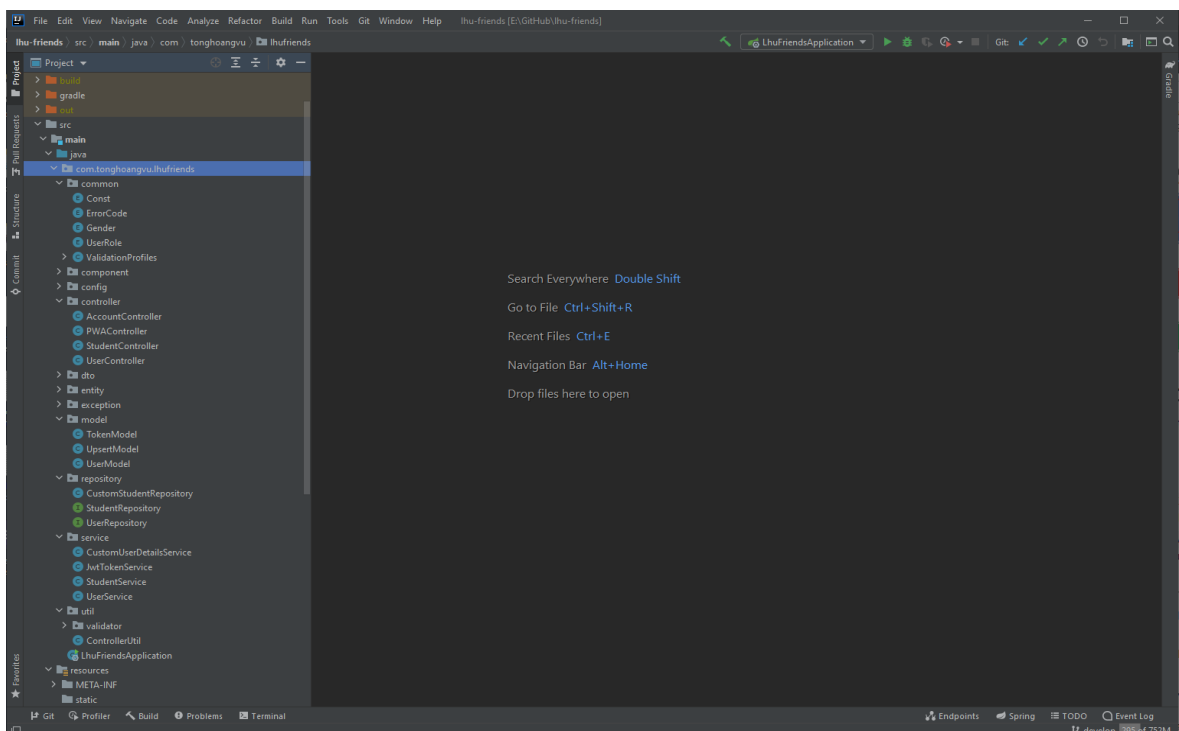
1.1 Cấu trúc dự án Spring Boot

1.1.1 Cấu trúc chung của ứng dụng

Dù cho project được tạo với Maven hay Gradle thì cấu trúc chung vẫn tương tự nhau, do tuân theo một template có sẵn (tên là Archetype):

- Thư mục gốc chứa các file linh tinh như pom.xml (của Maven), build.gradle và các file khác như .gitignore,... dùng để cấu hình dự án.
- Thư mục .mvn hoặc .gradle là thư mục riêng của Maven và Gradle, đừng nên đụng tới hay exclude nó ra khỏi source code.
- Code được chứa trong thư mục src.
- Thư mục build ra chứa các file class, file JAR. Với Maven là target còn Gradle là build.

1.1.2 Chi tiết cấu trúc source code



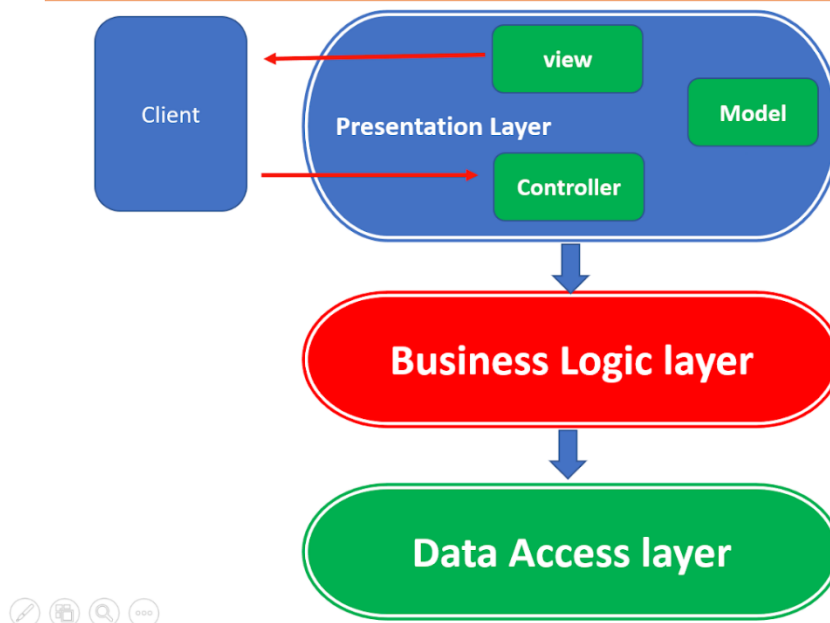
Hình 1: Cấu trúc source code

- Như hình, thư mục chính chúng ta cần quan tâm là `src/main/java/<tên package>`. Mọi code java đều nằm trong này:
 - Tên package chính được đặt dạng ngược với tên miền. Ví dụ như `tonghoangvu.com` thì đặt thành `com.tonghoangvu`. Cộng thêm tên project nữa.
 - Có các package con, mỗi package đại diện cho các class thuộc layer cụ thể (ví dụ như `service`, `controller`,...)
 - Thư mục `resources` chứa các tài nguyên của ứng dụng như hình ảnh, static file, properties file,...
 - Ngoài ra còn có `src/test` dùng để chứa các test class, dùng cho unit test.

1.2 Tổ chức source code

1.2.1 Tổ chức source code theo mô hình 3 lớp

Three-Tier architecture vs MVC pattern



Hình 2: Mô hình 3 lớp MVC

Tương ứng với từng thành phần của mô hình 3 lớp, thì chúng ta sẽ có các thư mục và cách đặt tên tương ứng:

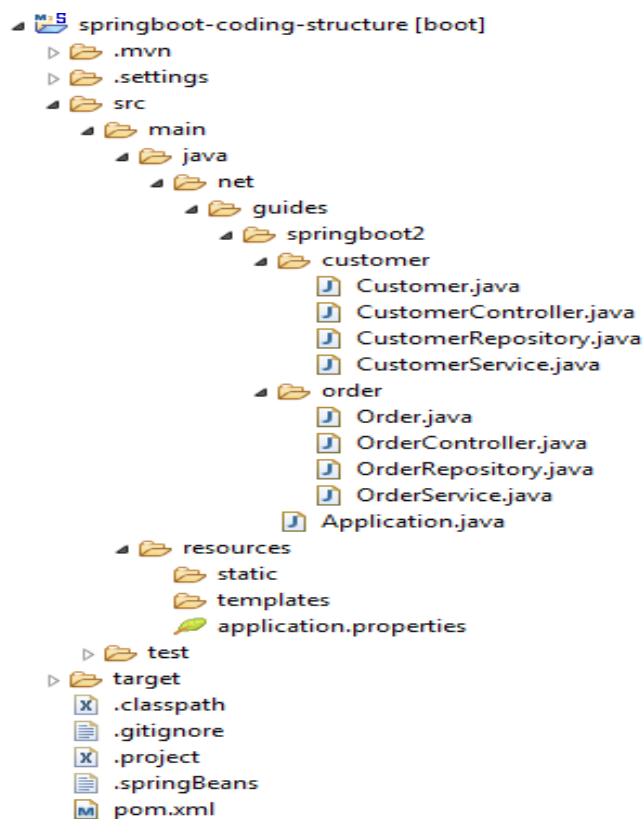
- Controller layer: đặt trong controller, các class là controller sẽ có hậu tố Controller (ví dụ `UserController`, `AuthController`,...)

- Service layer: đặt trong service, các class có hậu tố là Service và thường tương ứng với controller (ví dụ UserService,...)
- Data access layer: ca này khó, bởi vì layer này bao gồm repository (đặt trong repository và hậu tố tương tự), DTO, model, entity...

Ngoài ra, với các loại khác thì:

- util package chứa các lớp util (xử lý linh tinh), ví dụ như convert end date, tính toán đơn giản,...
- common package chứa các class định nghĩa như enum, interface, class dùng chung và đơn giản.
- exception package chứa các class có nhiệm vụ xử lý exception trong Spring Boot.
- component chứa các bean được định nghĩa còn lại nhưng không thuộc layer nào.

1.2.2 Tổ chức code theo tính năng



Hình 3: Tổ chức source code theo tính năng

Cụ thể thay vì chia thành các package dựa theo layer, thì cách này chia theo tính năng. Nghĩa là mỗi tính năng, ví dụ user package thì sẽ chứa nào là UserController, UserService,...

1.3 Cấu hình của dự án Spring Boot

Trong Spring Boot, chúng ta có thể quản lý cấu hình ứng dụng bằng cách sử dụng các tệp cấu hình như YAML hoặc Properties. Ví dụ, đối với tệp cấu hình `application.yml` với nội dung như sau:

```
server:
  port: 8080
spring:
  datasource:
    url: jdbc:mysql://localhost/
    username: root
    password: root
```

Nếu ta muốn sử dụng tệp Properties thì chỉ cần tạo một tệp `application.properties` và đặt các giá trị định nghĩa trong đó theo định dạng:

```
server.port=8080
spring.datasource.url=jdbc:mysql://localhost/test
spring.datasource.username=root
spring.datasource.password=root
args);
```

Bạn có thể sử dụng các IDE như IntelliJ IDEA, Eclipse hoặc NetBeans để tạo mới một dự án Spring Boot. Nếu sử dụng Eclipse, bạn có thể làm theo các bước sau:

- Mở Eclipse và chọn File -> New -> Other.
- Trong hộp thoại New, chọn Spring Boot -> Spring Starter Project và nhấn Next.
- Đặt tên cho dự án của bạn trong trường Project name.
- Chọn các thiết lập khác của dự án, ví dụ như package name, type of project, language.

- Chọn các dependency mà bạn muốn sử dụng trong dự án của mình.
- Sau khi hoàn tất, nhấn Finish để tạo dự án.

Sau khi tạo xong, Eclipse sẽ tạo ra một cấu trúc dự án Spring Boot mặc định với các tệp cấu hình, file chạy ứng dụng, và các tệp mã nguồn Java để bắt đầu phát triển ứng dụng của bạn.

1.4 Cài đặt Spring Tool Suite Cho Eclipse

Các bạn cũng có thể download sẵn bản IDE Spring Tool Suite base trên Eclipse tại: <https://spring.io/tools>

Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free. Open source.

4.18.0 - LINUX X86_64

4.18.0 - LINUX ARM_64

4.18.0 - MACOS X86_64

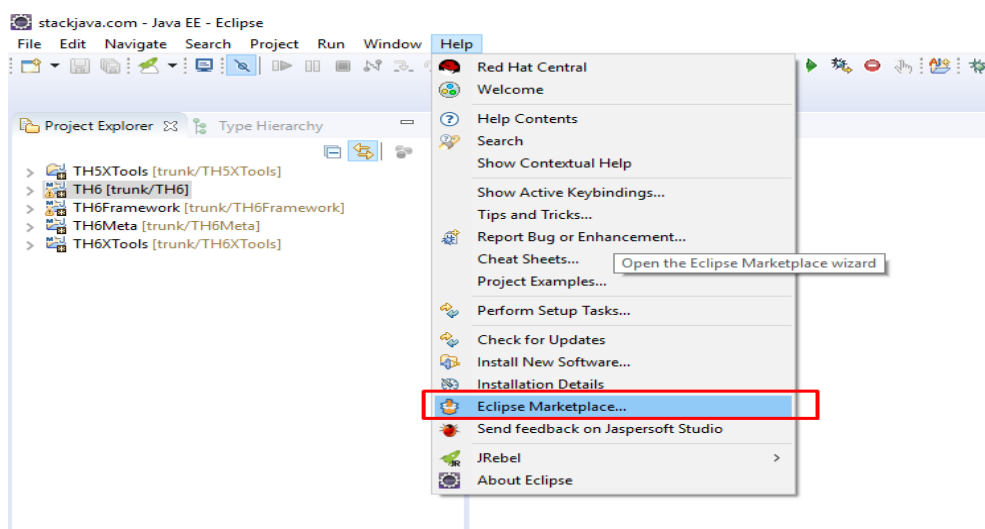
4.18.0 - MACOS ARM_64

4.18.0 - WINDOWS X86_64

Hình 4: IDE Spring Tool Suite base

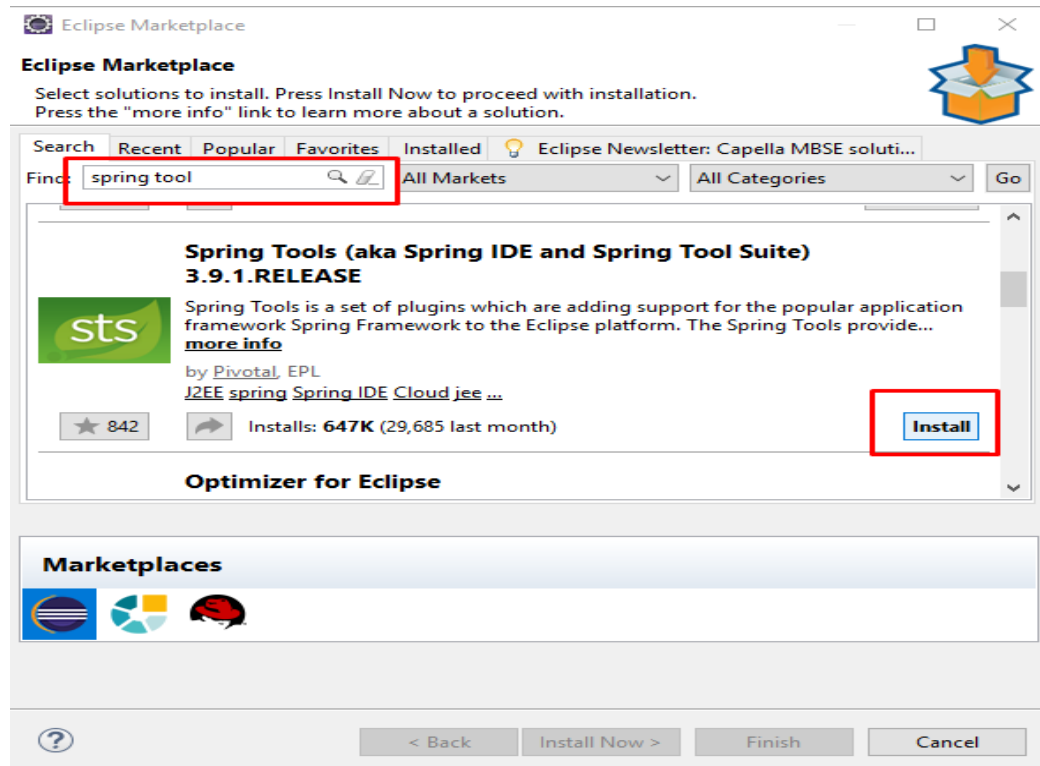
Nó có sẵn các phiên bản cho Window, Linux hay Mac

Truy cập Eclipse Marketplace



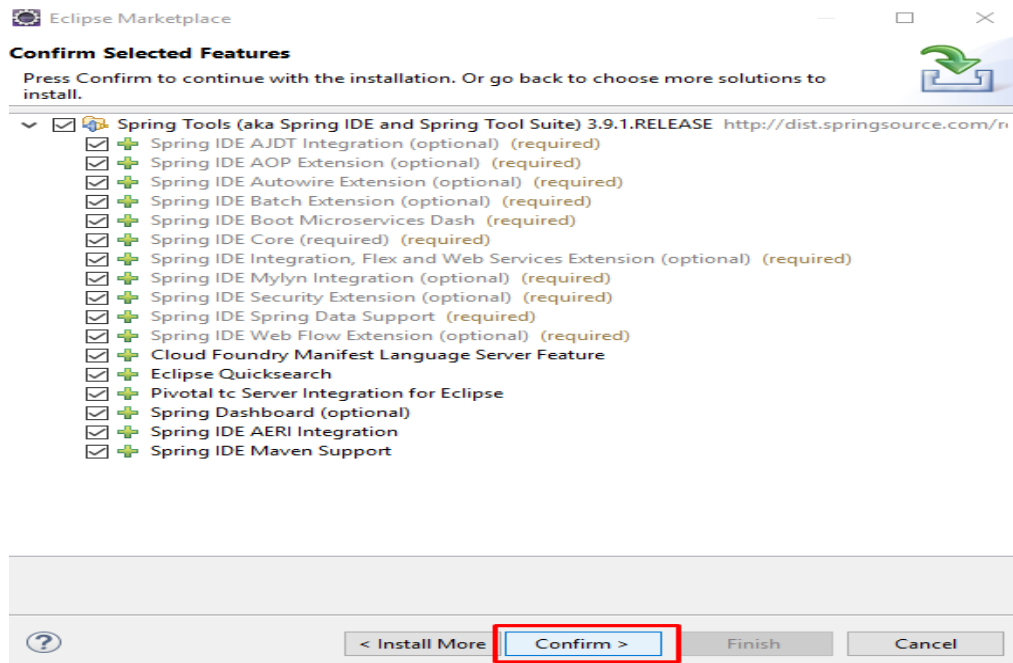
Hình 5: Truy cập Eclipse Marketplace

Gõ sts hoặc Spring tool để tìm kiếm Spring Tool > Click Install



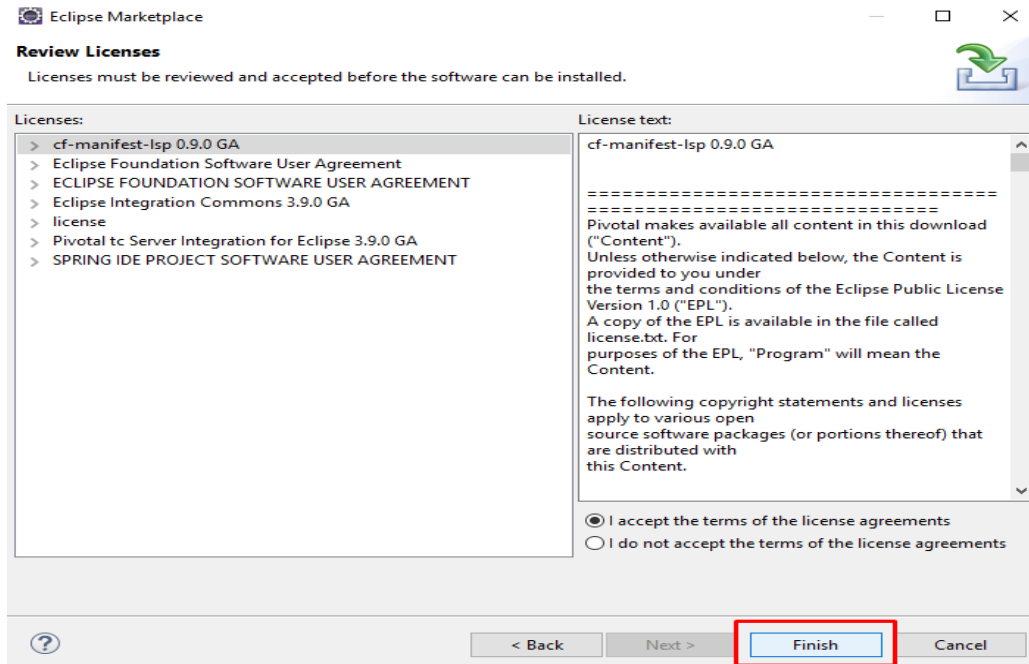
Hình 6: Install Spring Tools

Confirm



Hình 7: Confirm để installed

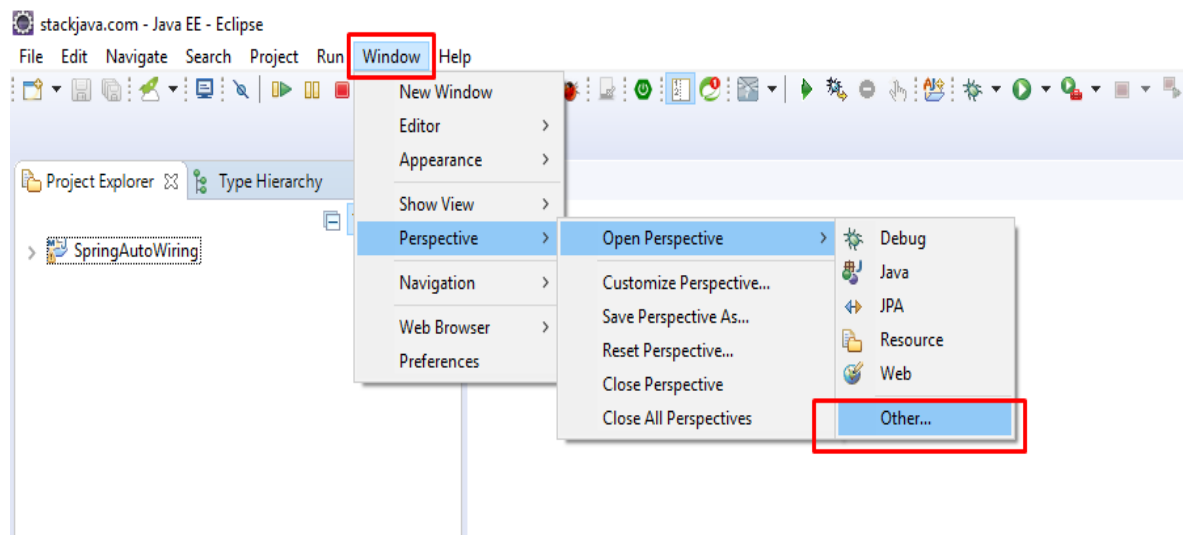
Chọn Accept và Finish



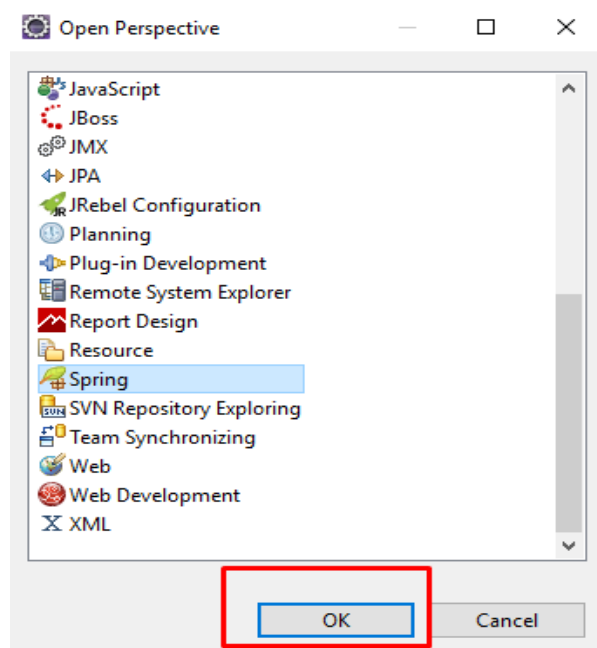
Hình 8: Chọn accept và finish

Sau khi cài đặt xong, Eclipse sẽ yêu cầu restart lại để áp dụng spring tool vào eclipse, click Yes để restart lại

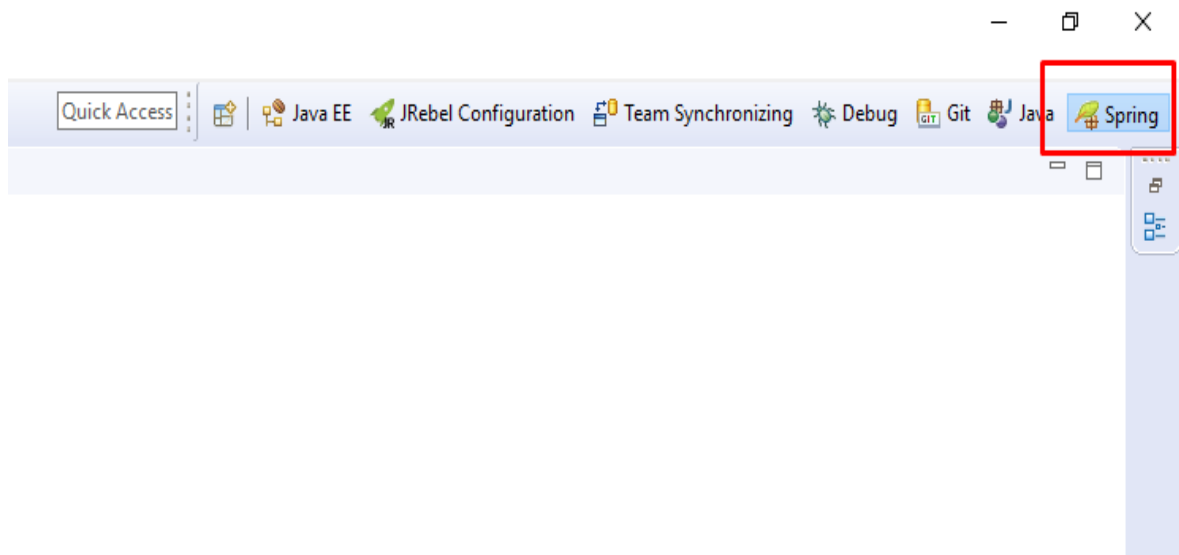
Hiển thị các tính năng của Spring: Window > Perspect > Open Perspect > Other > Spring



Hình 9: Hiển thị các tính năng của Spring

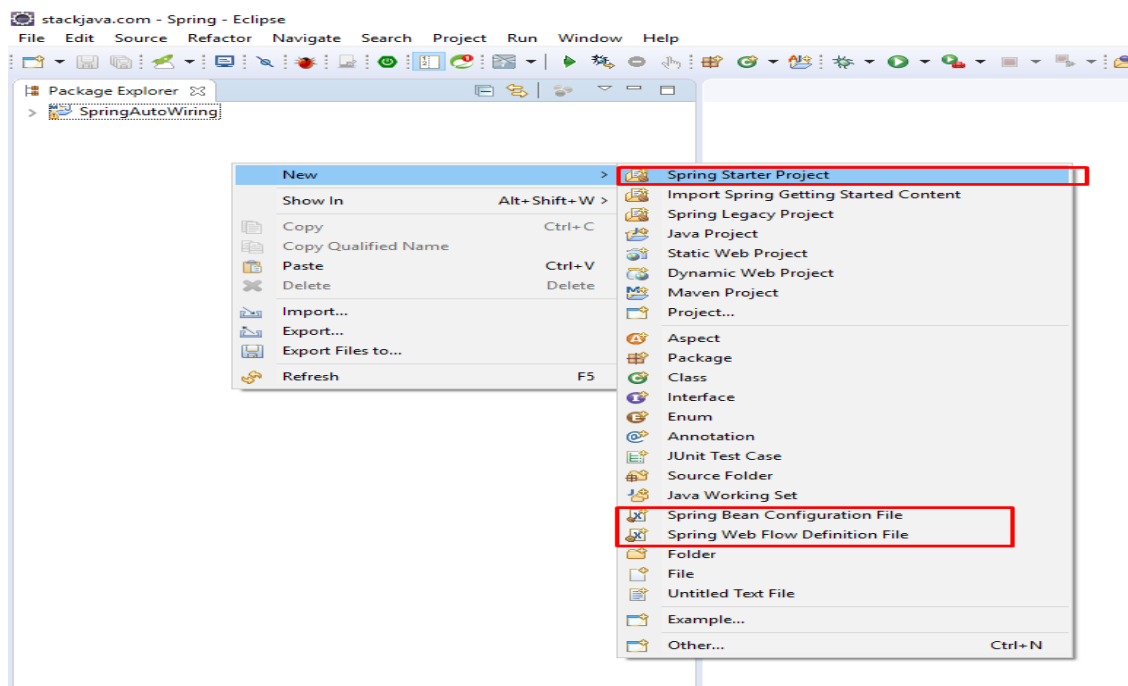


Hình 10: Chọn Ok



Hình 11: Bạn sẽ thấy biểu tượng Spring ở góc phải của Eclipse

Sau khi thêm Spring Tool vào Eclipse các bạn có thể dễ dàng tạo Spring Project hay các file bean cấu hình cho Spring.

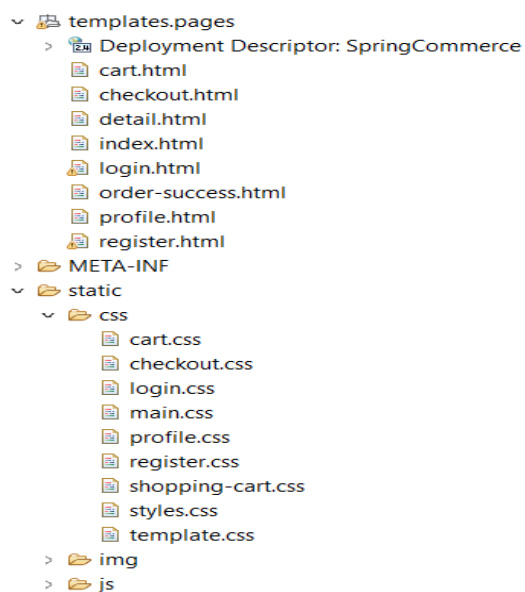


Hình 12: Tạo Spring Project

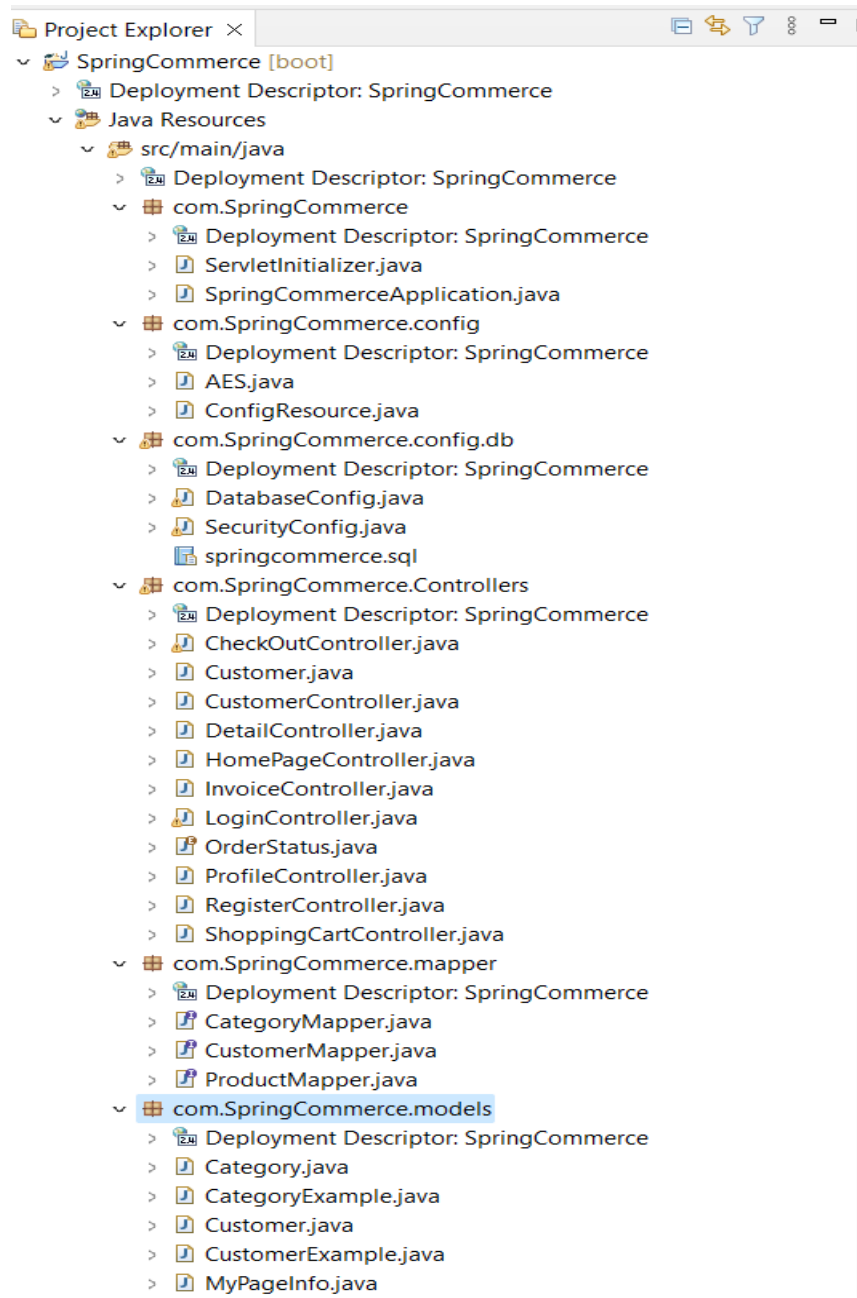
CHƯƠNG 3: DEMO VÀ PHÁT TRIỂN ỨNG DỤNG

1.1 Giới thiệu ứng dụng cần demo

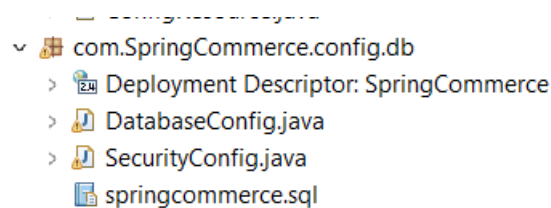
Demo web bán giày được sử dụng Spring Boot và Spring Security trên nền tảng Java.



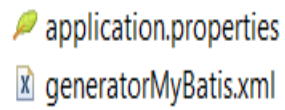
Hình 13: Các file html, css, img, js của trang web



Hình 14: Tổ chức source code theo tính năng đã trình bày ở trên

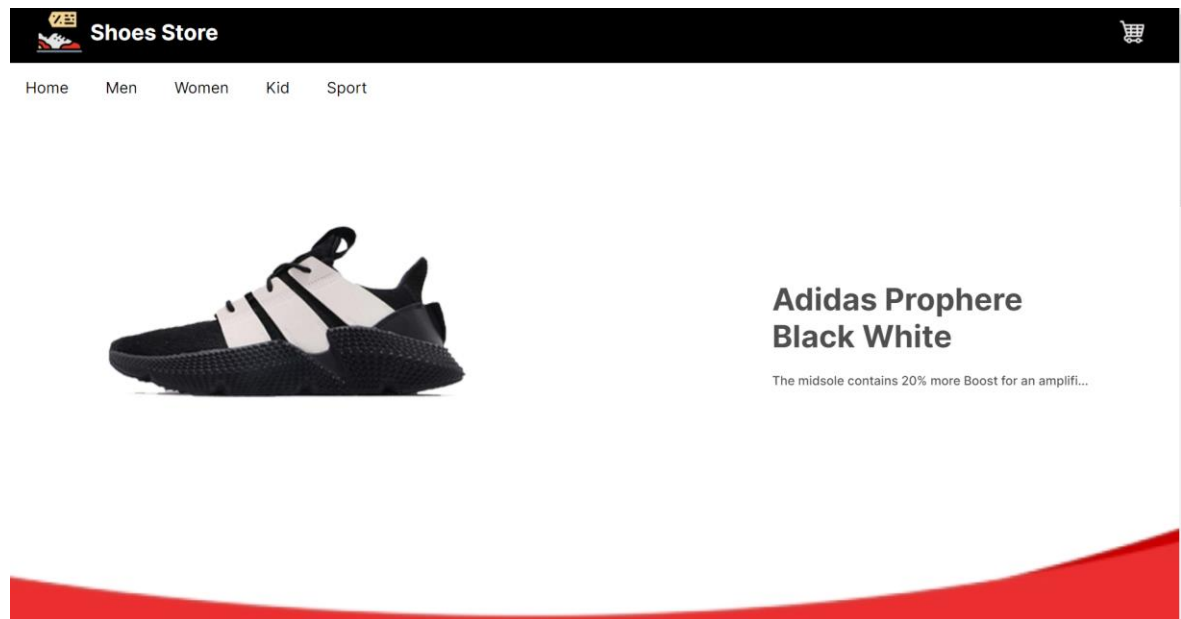


Hình 15: Database kèm theo các config

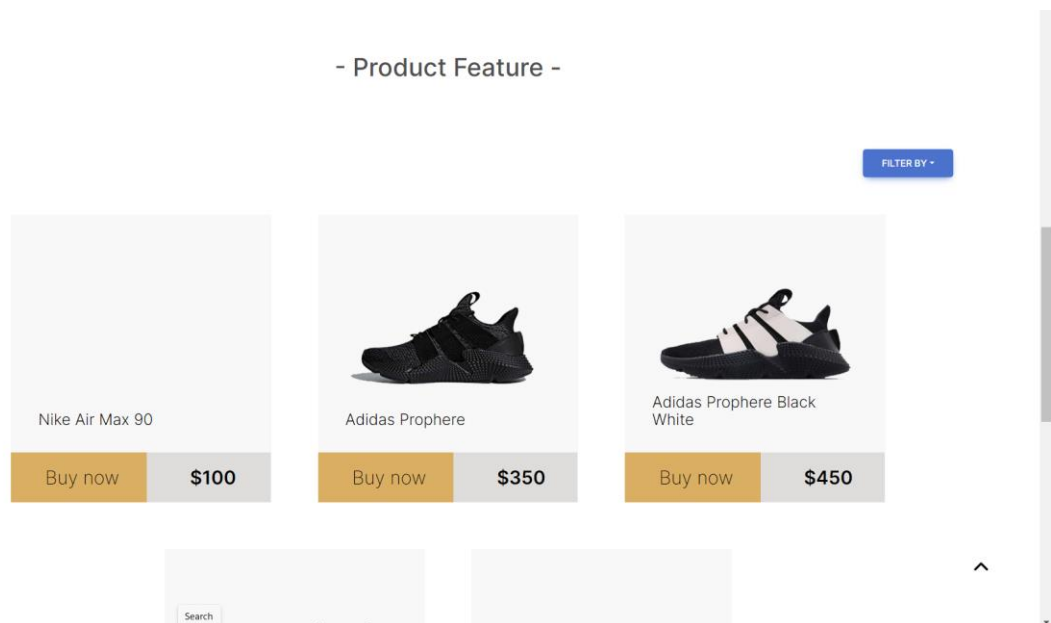


Hình 16: Các file cần đổi root để chạy được trang web

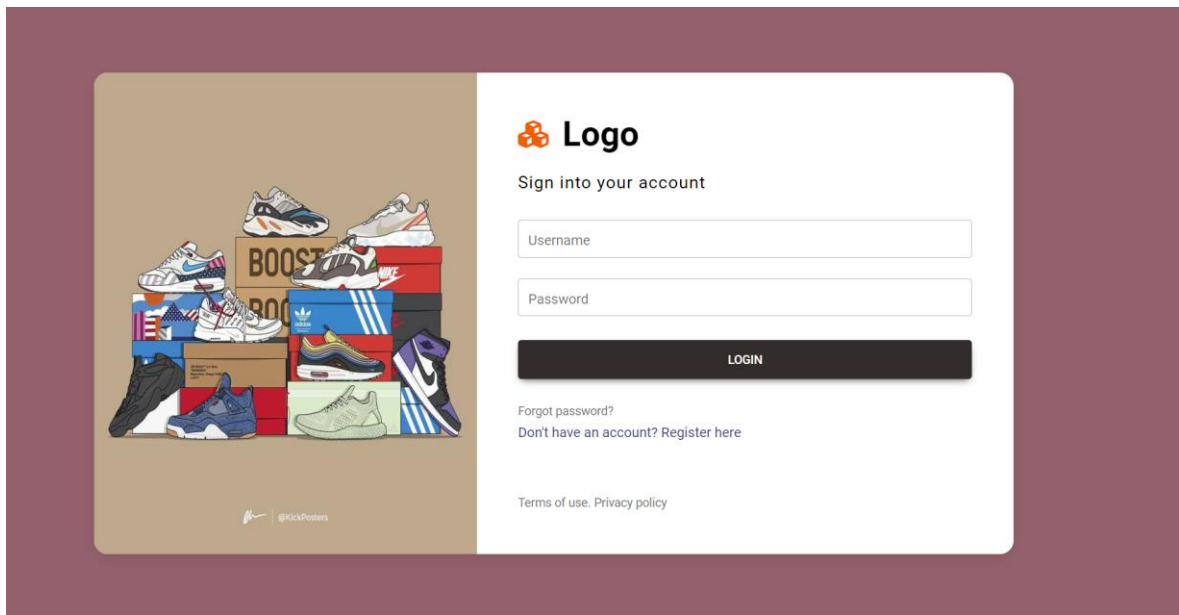
1.2 Hình ảnh trang web và kết quả demo



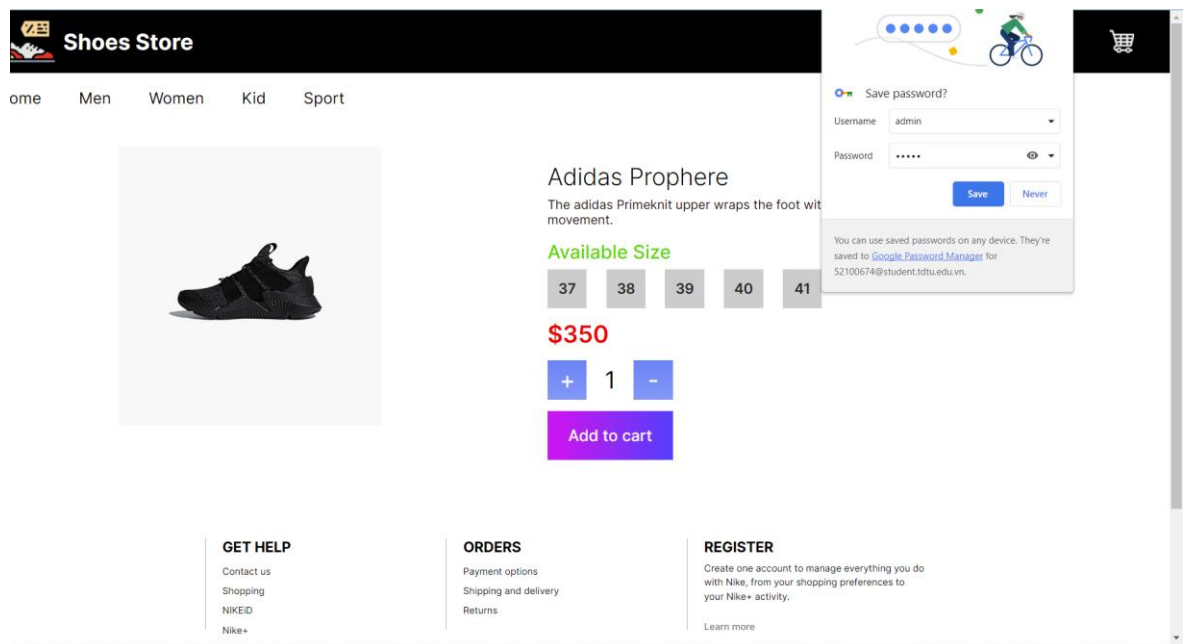
Hình 17: Giao diện của trang web



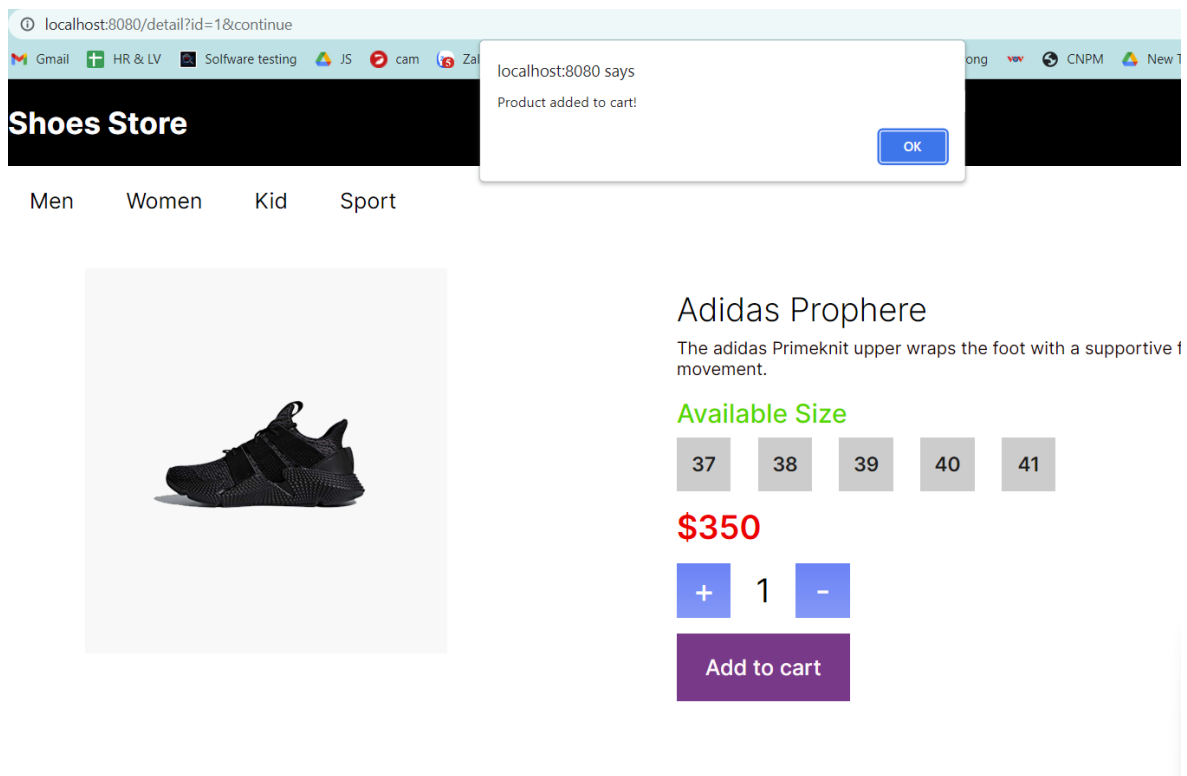
Hình 18: Giao diện của các sản phẩm



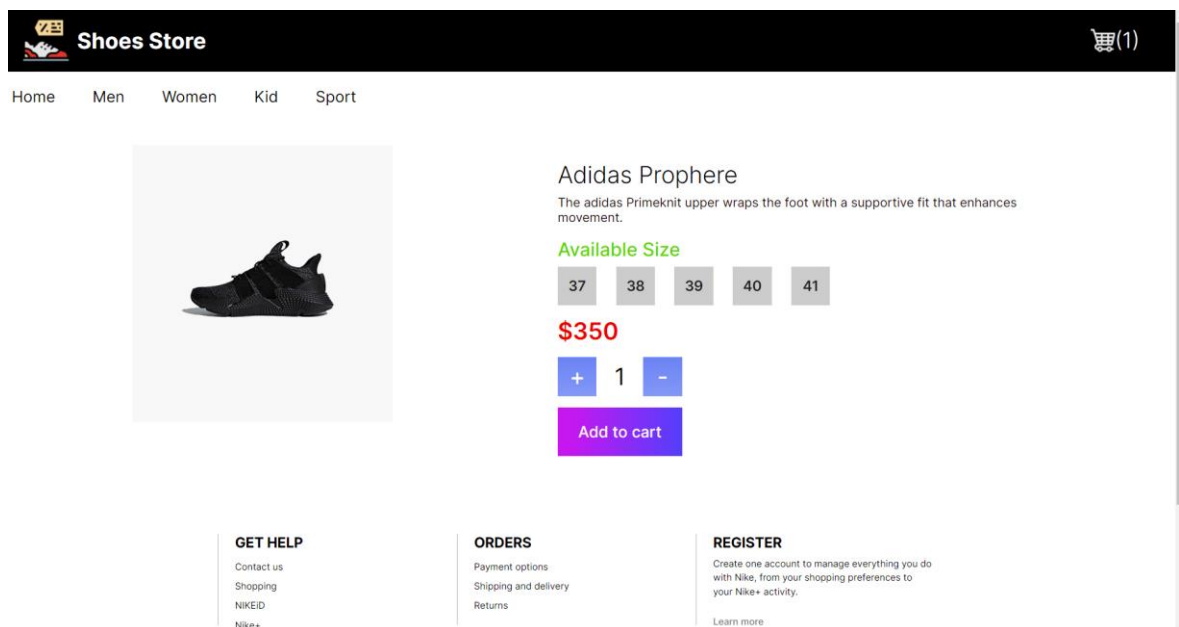
Hình 19: Giao diện login



Hình 20: Login thành công và có thể add sản phẩm



Hình 21: Thêm sản phẩm thành công và được localhost thông báo



Hình 21: Add vào giỏ có hiện số sản phẩm thành công

TÀI LIỆU THAM KHẢO

Tiếng Anh

- [1] F. Wiesen, M. Paluch and J. Winter, "A comparison of Java frameworks for building microservices: Spring Boot vs Micronaut," 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, 2019, pp. 376-381, doi: 10.1109/CCIS47457.2019.00080.
- [2] G. Alonso, S. Das, H. Wang, S. Gómez-Villamor and P. Martínez-Julia, "Microservices: A survey of practices, technologies, and challenges," IEEE Software, vol. 37, no. 3, pp. 24-35, May/June 2020, doi: 10.1109/MS.2019.2956927.
- [3] A. Almada-Lobo, J. R. Gonzalez-Sanchez and F. J. Dominguez-Mayo, "Towards a Framework for Dynamic Adaptation of Microservices-Based Systems," 2018 IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 2018, pp. 373-376, doi: 10.1109/SCC.2018.00059.
- [4] S. Raut and S. Potnis, "RESTful Web Service Using Spring Boot," 2018 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8599841.
- [5] S. Sharma and M. Singh, "A Comparative Analysis of Microservice Frameworks: Spring Boot and Vert.x," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), Chiang Mai, Thailand, 2019, pp. 1-6, doi: 10.1109/ICCIKE.2019.8868539.