

TRƯỜNG ĐẠI HỌC TIỀN GIANG

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**ĐỒ ÁN NGÀNH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN 18**

**ĐỀ TÀI**

**XÂY DỰNG ỨNG DỤNG WEB**

**BÁN HÀNG ONLINE VỚI JHIPSTER FRAMEWORK**

Người hướng dẫn: **Nguyễn Thị Phương Linh**

Sinh viên thực hiện:

Nguyễn Ngọc Minh Yến 018101108

**LỜI NÓI ĐẦU**

**LỜI CẢM ƠN**

🙞🙞🕮🙜🙜

Đầu tiên em xin chân thành cảm ơn Ban Giám Hiệu cùng quý Thầy Cô Trường Đại học Tiền Giang đặc biệt là toàn thể quý Thầy Cô trong khoa Công nghệ thông tin đã tận tình giảng dạy, trang bị cho em những kiến thức quý báu trong quá trình học tập tại Trường để em có thể hoàn thành tốt đồ án này.

Em xin gửi lời cảm ơn sâu sắc, chân thành đến cô Nguyễn Thị Phương Linh – người đã luôn tận tình hướng dẫn em hoàn thành đồ án này. Bên cạnh đó là sự giúp đỡ tận tình của bạn bè cùng lớp đã góp phần tạo thêm động lực vượt qua những khó khăn trong qua trình thực hiện đồ án này. Qua đó, em đã đạt được nhiều tiến bộ về kiến thức cũng như những kĩ năng làm việc bổ ích.

Mặc dù đã rất cố gắng trong quá trình làm đồ án, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự thông cảm và góp ý tận tình của quý Thầy Cô và các bạn.

Cuối cùng em kính chúc quý Thầy/Cô dồi dào sức khỏe và thành công trong công việc.

Xin chân thành cảm ơn!

Tiền Giang, ngày tháng 11 năm 2020

Sinh viên thực hiện

Nguyễn Ngọc Minh Yến

**MỤC LỤC**

[Chương 1. GIỚI THIỆU ĐỀ TÀI 1](#_Toc56190171)

[1.1. Giới thiệu đề tài 1](#_Toc56190172)

[1.1.1. Đặc tả đề tài 1](#_Toc56190173)

[1.1.2. Lý do chọn đề tài 1](#_Toc56190174)

[1.1.3. Mục tiêu đề tài 1](#_Toc56190175)

[1.1.4. Nhiệm vụ 2](#_Toc56190176)

[1.1.5. Phạm vi nghiên cứu của đề tài 2](#_Toc56190177)

[1.2. Các vấn đề tập trung giải quyết 2](#_Toc56190178)

[Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN 4](#_Toc56190179)

[2.1. Cơ sở lý thuyết 4](#_Toc56190180)

[2.1.1. Giả thuyết nghiên cứu 4](#_Toc56190181)

[2.1.2. Phương pháp nghiên cứu 4](#_Toc56190182)

[2.1.3. Công nghệ sử dụng 4](#_Toc56190183)

[2.1.4. Cơ sở lý thuyết 5](#_Toc56190184)

[Chương 3: PHÂN TÍCH, THIẾT KẾ, CÀI ĐẶT GIẢI PHÁP 9](#_Toc56190185)

[3.1. Phân tích yêu cầu 9](#_Toc56190186)

[3.1.1. Phân tích yêu cầu chức năng 9](#_Toc56190187)

[3.1.2. Phân tích yêu cầu phi chức năng 9](#_Toc56190188)

[3.2. Phân tích thiết kế 10](#_Toc56190189)

[3.2.1. Sơ đồ usecase 10](#_Toc56190190)

[3.2.2. Kịch bản cho các chức năng đã chọn 10](#_Toc56190191)

[3.2.3. Thiết kế CSDL 15](#_Toc56190192)

[3.3. Cài đặt giải pháp 16](#_Toc56190193)

[Chương 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 25](#_Toc56190194)

[4.1. Kết quả đạt được 25](#_Toc56190195)

[4.2. Hạn chế 25](#_Toc56190196)

[4.3. Hướng phát triển 25](#_Toc56190197)

[TÀI LIỆU THAM KHẢO 25](#_Toc56190198)

**DANH MỤC HÌNH**

[Hình 1 Mô hình kiến trúc nguyên khối 5](#_Toc55676244)

[Hình 2 Sơ đồ các thành phần cơ bản của Angular và mối quan hệ giữa các thành phần này 6](file:///E:\MY-LOVER-APP\Do-an-nganh\doc\bao-cao_018101108.docx#_Toc55676245)

[Hình 3 TypeScript và JavaScript 7](#_Toc55676246)

[Hình 4 Mô hình một ứng dụng sử dụng Spring Data 8](file:///E:\MY-LOVER-APP\Do-an-nganh\doc\bao-cao_018101108.docx#_Toc55676247)

[Hình 5 Sơ đồ usecase tổng quát của ứng dụng 10](file:///E:\MY-LOVER-APP\Do-an-nganh\doc\bao-cao_018101108.docx#_Toc55676248)

[Hình 6 Tạo thư mục flowermart 15](#_Toc55676249)

[Hình 7 Khởi tạo dự án trong thư mục vừa tạo 16](#_Toc55676250)

[Hình 8 Các mục cần chọn để tạo dự án 16](#_Toc55676251)

[Hình 9 Khởi tạo dự án thành công 17](#_Toc55676252)

[Hình 10 Cập nhật thông tin cấu hình kết nối database 18](#_Toc55676253)

[Hình 11 Chạy dự án trong môi trường phát triển 19](#_Toc55676254)

[Hình 12 Giao diện mặc định của dự án khi vừa khởi tạo 19](#_Toc55676255)

**DANH MỤC BẢNG**

[Bảng 1 Usecase đăng ký 10](#_Toc55722059)

[Bảng 2 Usecase đăng nhập 11](#_Toc55722060)

[Bảng 3 Usecase thay đổi mật khẩu 11](#_Toc55722061)

[Bảng 4 Usecase cập nhật thông tin cá nhân 12](#_Toc55722062)

[Bảng 5 Usecase xem thông tin sản phẩm 12](#_Toc55722063)

[Bảng 6 Usecase thêm sản phẩm cần mua vào giỏ hàng 13](#_Toc55722064)

[Bảng 7 Usecase thanh toán hóa đơn 13](#_Toc55722065)

[Bảng 8 Usecase thêm sản phẩm vào CSDL 14](#_Toc55722066)

[Bảng 9 Cập nhật thông tin sản phẩm 14](#_Toc55722067)

[Bảng 10 Xóa sản phẩm 15](#_Toc55722068)

**TÓM TẮT**

**1. Mục tiêu**

Tìm hiểu lý thuyết: Nắm được cơ chế hoạt động của kiến trúc nguyên khối, hiểu được kiến thức nền tảng của lập trình sử dụng Angular, TypeScript, Spring Boot, Spring JPA … Tìm hiểu về hoạt động bán hàng online.

Ứng dụng thực nghiệm: Áp dụng tất cả các kiến thức lý thuyết đã tìm hiểu và đã được học để xây dựng thành công ứng dụng web bán hàng online.

**2. Nhiệm vụ**

Tìm hiểu lý thuyết: Tìm hiểu nguyên lý hoạt động của Monolithic (Jhipster), nắm được các kiến thức, định nghĩa cốt lõi của Jhipster, Angular, TypeScript, Spring Boot, Spring JPA…

Ứng dụng thực nghiệm: Đặt ra và phân tích yêu cầu, liệt kê các chức năng cần thiết, thiết kế cơ sở dữ liệu, xây dựng cấu trúc ứng dụng, cấu hình ứng dụng, thiết kế giao diện, thiết kế các báo cáo, cài đặt giải thuật.

**3. Phương pháp giải quyết**

Tìm hiểu lý thuyết: Phương pháp nghiên cứu lý thuyết, phát hiện và giải quyết vấn đề.

Ứng dụng thực nghiệm: Phương pháp thực nghiệm khoa học, phân tích và tổng kết tiếp thu kinh nghiệm.

**4. Kết quả đạt được**

Về lý thuyết: Hiểu được cách hoạt động của Monolithic, hiểu được các kiến thức nền tảng để sử dụng Jhipster, Angular, TypeScript, Spring Boot, Spring JPA, …

Ứng dụng thực nghiệm: Hoàn thiện ứng dụng web bán hàng online.

# 

# Chương 1. GIỚI THIỆU ĐỀ TÀI

## Giới thiệu đề tài

### Đặc tả đề tài

Đề tài xây dựng ứng dụng web bán hàng có các chức năng cơ bản như sau:

* Đăng ký
* Đăng nhập
* Phân quyền người dùng
* Hiển thị thông tin theo quyền người dùng
* Tìm kiếm theo từ khóa
* Giỏ hàng
* Quản lý thông tin người dùng
* Quản lý thông tin sản phẩm
* Quản lý thông tin bán hàng
* Quản lý thông tin sự kiện & khuyến mãi

### Lý do chọn đề tài

Ngày nay, với tốc độ phát triển mạnh mẽ của công nghệ thông tin đã giúp con người phát triển nhiều lĩnh vực.

Trong đó, việc ứng dụng công nghệ thông tin vào lĩnh vực thương mại đã mang lại thành quả rất lớn. Từ đây việc mua bán ngày càng tiện lợi và nhanh chóng hơn. Các doanh nghiệp tiếp cận việc bán hàng qua mạng đem lại nhiều lợi ích hơn so với việc bán hàng truyền thống như:

* Dễ dàng tiếp cận hơn với người tiêu dùng.
* Giảm chi phí quảng bá.
* Tăng doanh thu cho cửa hàng.

### Mục tiêu đề tài

Tìm hiểu lý thuyết:

* + Nắm được cơ chế hoạt động của kiến trúc nguyên khối.
  + Hiểu được kiến thức nền tảng của lập trình về Angular X, Spring Boot, TypeScript, Java 11.
  + Tìm hiểu về trang web bán hàng online.
  + Rèn luyện kỹ năng phân chia thời gian, phân tích, giải quyết vấn đề, làm việc thực tế một các hiệu quả.
  + Ứng dụng những kiến thức đã học và kiến thức tìm hiểu xây dựng thành công ứng dụng web bán hàng online.

Ứng dụng thực nghiệm:

* Xây dựng được hệ thống quản lý cửa hàng: Quản lý các thông tin của nhân viên, quản lý các thông tin khách hàng, quản lý các thông tin sản phẩm, quản lý hóa đơn.
* Quản lý, lưu trữ được các thông tin các sản phẩm.
* Liên kết giao diện, thông tin các service liên quan được chính xác, nhanh chóng và hiệu quả.
* Ứng dụng có đáp ứng đủ cho cửa hàng, và có khả năng áp dụng vào thực tế.
* Ứng dụng hoạt động hiệu quả, nhanh chóng.
* Giao diện trực quan và dễ sử dụng.

### Nhiệm vụ

Tìm hiểu lý thuyết:

* + Tìm hiểu nguyên lý hoạt động của Monolithic (Jhipster).
* Tìm hiểu về Jhipster, Angular, TypeScript, Java 11, Docker, Spring Boot, Spring JPA,.. và các thư viện bên thứ 3 liên quan.
  + Tìm hiểu về trang web bán hàng online.

Ứng dụng thực nghiệm:

Đặt ra và phân tích yêu cầu, liệt kê các chức năng cần thiết, thiết kế cơ sở dữ liệu, xây dựng cấu trúc ứng dụng, cấu hình ứng dụng, thiết kế giao diện, thiết kế các báo cáo, cài đặt giải thuật.

### Phạm vi nghiên cứu của đề tài

Tập trung tìm hiểu các công nghệ để xây dựng nên 1 ứng dụng web bán hàng online đơn giản với chức năng cơ bản cần thiết.

## Các vấn đề tập trung giải quyết

Giải quyết các vấn đề sau:

* Thiết kế theo kiến trúc nguyên khối, đảm bảo tính bảo mật, hoạt động ổn định, kết nối truyền dữ liệu một cách chính xác.
* Giao diện đơn giản, dễ sử dụng : có bố cục hợp lí, các menu chức năng rõ ràng.
* Ứng dụng có thể chạy được trên các trình duyệt web trên máy tính như (Google Chrome, IE,…)

# Chương 2: CƠ SỞ LÝ THUYẾT VÀ THỰC TIỄN

## Cơ sở lý thuyết

### Giả thuyết nghiên cứu

Nếu xây dựng thành công hệ thống sẽ giúp cho việc quảng bá, cập nhật thông tin sản phẩm, tư vấn khách hàng, mua bán, quản lý một cách dễ dàng và tiện lợi.

Khách hàng có thể xem thông tin sản phẩm, tìm kiếm sản phẩm, đăng nhập mua hàng nhanh chóng.

### Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết: Nghiên cứu lý thuyết các vấn đề liên quan như cơ sở dữ liệu, ngôn ngữ lập trình, tài liệu.

Phương pháp quan sát khoa học: Quan sát khách quan các vấn đề liên quan đến quy trình quản lý bệnh nhân theo cây phả hệ, tài liệu thông tin bác sĩ gia đình.

Phương pháp thực nghiệm: Áp dụng lấy dữ liệu từ hệ thống thực tế.

Phương pháp nghiên cứu dựa theo quy trình: Quy trình thiết kế gồm các bước sau đây:

* Xác định yêu cầu: Thu thập thông tin từ các tài liệu liên quan.
* Phân tích: Phân tích, nghiên cứu để giải quyết vấn đề từ các yêu cầu thu thập được, từ bước xác định yêu cầu, vẽ các mô hình, xử lý dữ liệu của bài toán.
* Thiết kế: Thiết kế giao diện và các giải thuật, các lớp xử lí từ việc phân tích yêu cầu để có thể hoàn thành hệ thống.
* Cài đặt: Sau khi hoàn thành cơ bản chức năng của hệ thống, cần cho chạy thử nghiệm để kiểm tra, tìm lỗi và sửa lỗi...
* Kiểm chứng: thu thập, lấy ý kiến đánh giá của người sử dụng. Từ đó, tổng kết để thấy được những cái hoàn thiện, thiếu sót của hệ thống để có thể chỉnh sửa kịp thời.
* Phát triển thực tế: Nếu đề tài được ứng dụng sẽ triển khai và cài đặt trên host để có thể chính thức hoạt động.

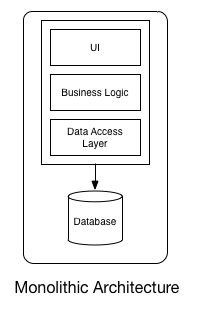
### Công nghệ sử dụng

* Hệ điều hành: Windows 10
* Trình soạn thảo mã nguồn mở: Visual Studio Code, Intellij IDEA 2020
* Phần mềm Docker
* Trình duyệt web: Google Chrome, Microsoft Edge Chromium, Mozilla Firefox,…

### Cơ sở lý thuyết

Ứng dụng được thiết kế dựa trên các kiến thức được học từ các môn “Nhập môn cơ sở dữ liệu”, “lập trình hướng đối tượng” và kiến thức tự tìm hiểu như: kiến trúc nguyên khối, Jhipster, Angular X, TypeScript, Spring Boot, Spring Data JPA.

1. Giới thiệu kiến trúc nguyên khối



Hình 1 Mô hình kiến trúc nguyên khối

Kiến trúc nguyên khối là một giải pháp truyền thống được áp dụng để xây dựng các ứng dụng với các ưu điểm như sau:

* Phát triển và triển khai một các đơn giản
* Hiệu suất tốt hơn

Kiến trúc nguyên khối thích hợp cho nhựng phần mềm có quy mô nhỏ như ứng dụng web bán hàng trong đồ án này.

1. Giới thiệu Angular X

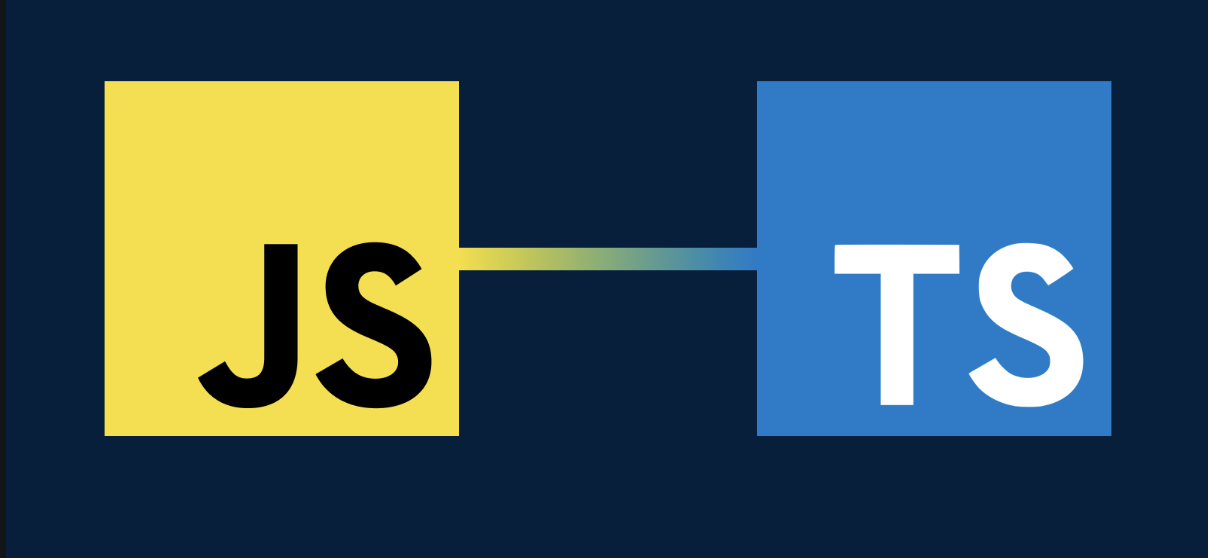
Hình Sơ đồ các thành phần cơ bản của Angular và mối quan hệ giữa các thành phần này

Angular là một framework, một ngôn ngữ chức năng hoạt động trên tất cả các trình duyệt hiện đại nhất ngày nay

Angular có cấu trúc dễ hiểu và dễ áp dụng phổ biến, tương thích mạnh mẽ với MVC Core (MVC là viết tắt của Model View Controller - một mẫu cấu trúc thiết kế rất hiệu quả cho việc mở rộng ứng dụng). Thông thường những framework khác yêu cầu lập trình viên dành rất nhiều công sức để chia ứng dụng thành những phần MVC nhỏ, sau đó cần phải viết code để nối lại các thành phần đó. Tuy nhiên, khi dử dụng Angular, công việc của lập trình viên sẽ đơn giản hơn rất nhiều do chỉ cần tạo thành phần và Angular sẽ hoàn thành cho bạn tất cả những phần còn lại bao gồm kết nối và quản lý. Không chỉ vậy, framework này sẽ mang lại các dòng code “sạch”, rất dễ đọc và dễ để sử dụng phổ biến.

Angular hoạt động dựa chủ yếu vào mã HTML để xây dựng và phát triển web, giúp lập trình viên bỏ qua thủ tục viết JavaScript cho các phần từ giao diện, khiến cho việc xây dựng web dễ dàng và nhẹ nhàng hơn rất nhiều. Nhờ vào sử dụng HTML, lập trình viên sẽ mất ít thời gian hơn trong việc định hình cấu trúc thẩm mỹ của giao diện và thao tác xử lý, hơn nữa còn có thể giúp truy cập và sửa code trực tiếp trên file HTML mà không cần cài đặt thêm bất kì phần mềm nào.

1. Giới thiệu TypeScript



Hình 3 TypeScript và JavaScript

TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft, nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript.

1. Giới thiệu Spring Boot

* 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.
* Ưu điểm của SpringBoot:

+ Có các tính năng của Spring Framework.

+ Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)

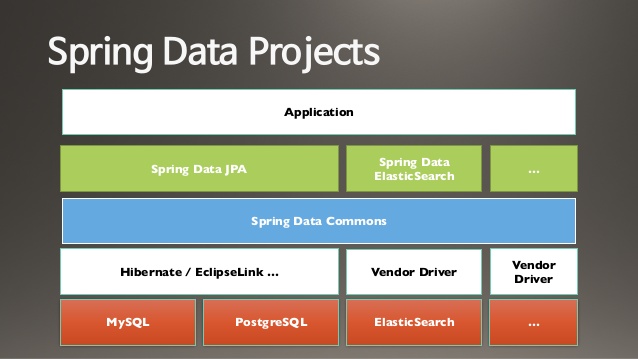
+ Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR

+ Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)

+ Không yêu cầu XML config…

+ Cung cấp nhiều plugin

+ Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

1. Giới thiệu Spring Data JPA

Hình Mô hình một ứng dụng sử dụng Spring Data

* Là một phần của Spring Data, nó làm cho việc truy cập dữ liệu của các ứng dụng dựa trên Spring một các dễ dàng.
* Các tính năng của Spring Data JPA
  + Hỗ trợ xây dựng kho lưu trữ dựa trên Spring và JPA
  + Kiểm tra minh bạch của lớp miền.
  + Hỗ trợ phân trang, thực thi truy vấn động, khả năng tích hợp mã truy cập dữ liệu tùy chỉnh.
  + Xác thực các câu truy vấn có chú thích @Query tại thời gian khởi động.
  + Hỗ trợ ánh xạ thực thể dựa trên XML
  + Cấu hình kho lưu trữ dựa trên JavaConfig

Đồ án này chủ yếu sử dụng Spring Data JPA để thực hiện các câu truy vấn dữ liệu tới database.

1. Giới thiệu về Jhipster

JHipster( viết tắt của Java Hipster) là cách đơn giản để chúng ta tạo ra một project xung quanh những công nghệ được ưa thích nhất với Spring technologies và Angular/React.

# Chương 3: PHÂN TÍCH, THIẾT KẾ, CÀI ĐẶT GIẢI PHÁP

## Phân tích yêu cầu

### Phân tích yêu cầu chức năng

Các chức năng cơ bản của chương trình được chọn để thực hiện trong đồ án này

* Đăng ký tài khoản người dùng
* Đăng nhập
* Thay đổi mật khẩu
* Cập nhật thông tin cá nhân
* Xem thông tin sản phẩm, loại sản phẩm
* Thêm sản phẩm vào giỏ
* Thanh toán
* Quản lý thông tin sản phẩm
  + Thêm sản phẩm
  + Cập nhật thông tin sản phẩm
  + Xóa sản phẩm

### Phân tích yêu cầu phi chức năng

* Giao diện: đơn giản, dễ sử dụng, phù hợp với các nhân viên, bác sĩ của bệnh viện.
* Bảo mật: Không bị khai thác lỗi.
* Có thể hoạt động trên các trình duyệt cơ bản trên máy tính.
* Chạy nhanh, ổn định.

## Phân tích thiết kế

### Sơ đồ usecase

Hình Sơ đồ usecase tổng quát của ứng dụng

### Kịch bản cho các chức năng đã chọn

* Đăng ký:

#### Bảng 1 Usecase đăng ký

|  |  |
| --- | --- |
| Tên | Đăng ký |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Đăng ký thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu nhập lại mật khẩu |
| Sự kiện chính | 1. Người dùng nhấn vào nút “tài khoản → đăng ký”. 2. Người dùng nhập thông tin tài khoản (tên đăng nhập, email, mật khẩu, xác nhận lại mật khẩu). 3. Nếu mật khẩu có đủ độ mạnh (có ít nhất 4 kí tự, có số) thì sẽ hiện thông báo đăng ký thành công. 4. Người dùng chờ người quản lý “kích hoạt” tài khoản để đăng nhập |
| Ngoại lệ | 3.1 Nếu mật khẩu không đủ độ mạnh thì không cho phép nhấn vào nút đăng ký. |

* Đăng nhập:

#### Bảng 2 Usecase đăng nhập

|  |  |
| --- | --- |
| Tên | Đăng nhập |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Tài khoản người dùng đã tồn tại trong hệ thống |
| Hậu điều kiện | Đăng nhập thành công vào hệ thống |
| Điều kiện tối thiểu | Yêu cầu đăng nhập lại |
| Sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập thông tin tài khoản và mật khẩu 3. Nhấn “Đăng nhập” 4. Hệ thống chứng thực tài khoản 5. Hệ thống thông báo đăng nhập thành công |
| Ngoại lệ | 4.1. Hệ thống chứng thực thất bại, yêu cầu nhập lại thông tin tài khoản, mật khẩu. |

* Thay đổi mật khẩu:

#### Bảng 3 Usecase thay đổi mật khẩu

|  |  |
| --- | --- |
| Tên | Thay đổi mật khẩu |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Có tài khoản và đã đăng nhập |
| Hậu điều kiện | Thay đổi mật khẩu thành công |
| Điều kiện tối thiểu | Nhập đúng email |
| Sự kiện chính | 1. Khi người dùng đã đăng nhập nhấn vào nút “tài khoản → mật khẩu”, nếu chưa đăng nhập thì nhấn “quên mật khẩu”. 2. Người dùng nhập nhập khẩu hiện tại, mật khẩu mới và xác nhận lại mật khẩu mới. 3. Hệ thống gửi thông báo về email của người dùng. 4. Hệ thống cập nhật thay đổi mật khẩu của người dùng. |
| Ngoại lệ | 2.1. Người dùng nhập sai mật khẩu hiện tại.  2.2. Người dùng nhập mật khẩu mới không đủ độ mạnh.  2.3. Người dùng nhập xác thực mật khẩu mới không trùng với mật khẩu mới. |

* Cập nhật thông tin cá nhân:

#### Bảng 4 Usecase cập nhật thông tin cá nhân

|  |  |
| --- | --- |
| Tên | Cập nhật thông tin cá nhân |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Có tài khoản và đã đăng nhập |
| Hậu điều kiện | Thay đổi thông tin cá nhân thành công |
| Điều kiện tối thiểu | Yêu cầu người dùng nhập lại thông tin |
| Sự kiện chính | 1. Hiển thị giao diện thông tin cá nhân khi người dùng chọn “tài khoản” 2. Thay đổi các thông tin cá nhân theo ý muốn 3. Nhấn “lưu lại” 4. Hệ thống kiểm tra 5. Hệ thống lưu lại thay đổi |
| Ngoại lệ | * 1. Hệ thống lưu thất bại |

* Xem thông tin sản phẩm:

#### Bảng 5 Usecase xem thông tin sản phẩm

|  |  |
| --- | --- |
| Tên | Xem thông tin sản phẩm |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Có khả năng truy cập Internet |
| Hậu điều kiện | Hiện thị danh sách sản phẩm |
| Điều kiện tối thiểu | Load lại trình duyệt |
| Sự kiện chính | 1. Hiển thị giao diện trang chủ khi người dùng truy cập đến đại chỉ trang web. 2. Khi người dùng chọn vào 1 sản phẩm thì sẻ hiển thị trang web thông tin chi tiết sản phẩm. |
| Ngoại lệ |  |

* Thêm sản phẩm cần mua vào giỏ:

#### Bảng 6 Usecase thêm sản phẩm cần mua vào giỏ hàng

|  |  |
| --- | --- |
| Tên | Thêm sản phẩm cần mua vào giỏ hàng |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Đã đăng nhập |
| Hậu điều kiện | Thêm sản phẩm vào giỏ thành công |
| Sự kiện chính | 1. Khi người dùng chọn 1 sản phẩm và nhấn “Thêm vào giỏ hàng” sẽ thêm món hàng vào giỏ hàng. 2. Người dùng chọn “Giỏ hàng” trên thanh Navbar để truy cập vào giao diện giỏ hàng. 3. Người dùng có thể thay đổi số lượng sản phẩm cần mua trong giao diện giỏ hàng. |
| Ngoại lệ | * 1. Sản phẩm đã bán hết, thêm thất bại |

* Thanh toán:

#### Bảng 7 Usecase thanh toán hóa đơn

|  |  |
| --- | --- |
| Tên | Thanh toán |
| Tác nhân chính | Người dùng |
| Tiền điều kiện | Đã đăng nhập  Có ít nhất 1 món hàng trong giỏ hàng |
| Hậu điều kiện | Thanh toán thành công |
| Sự kiện chính | 1. Người dùng truy cập vào giao diện “giỏ hàng” sẽ hiển thị danh sách các món hàng đã thêm vào giỏ. 2. Nhấn “Thanh toán” để tiến hành thanh toán hóa đơn. |
| Ngoại lệ | * 1. Thanh toán thất bại, giỏ hàng hiện không có sản phẩm nào.   2. Thanh toán thất bại, món hàng cần mua đã bán hết |

* Thêm sản phẩm và CSDL

#### Bảng 8 Usecase thêm sản phẩm vào CSDL

|  |  |
| --- | --- |
| Tên | Thêm sản phẩm |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Đã đăng nhập vào tài khoản có quyền quản trị |
| Hậu điều kiện | Thêm sản phẩm thành công |
| Sự kiện chính | 1. Hiển thị giao diện thêm sản phẩm 2. Người quản lý nhập các thông tin của sản phẩm 3. Nhấn nút lưu 4. Hệ thống kiểm tra 5. Hệ thống lưu lại thông tin sản phẩm |
| Ngoại lệ | * 1. Thêm sản phẩm thất bại, hệ thống đã tồn tài sản phẩm này |

* Cập nhật thông tin sản phẩm

#### Bảng 9 Cập nhật thông tin sản phẩm

|  |  |
| --- | --- |
| Tên | Cập nhật thông tin sản phẩm |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Đã đăng nhập vào tài khoản có quyền quản trị |
| Hậu điều kiện | Cập nhật thông tin sản phẩm thành công |
| Điều kiện tối thiểu | Nhập lại tên sản phẩm |
| Sự kiện chính | 1. Hiển thị giao diện danh sách sản phẩm 2. Người quản lý chọn một sản phẩm và nhấn nút cập nhật 3. Hiển thị giao diện cập nhật thông tin sản phẩm 4. Nhấn nút lưu 5. Hệ thống kiểm tra 6. Hệ thống lưu lại thông tin sản phẩm vừa cập nhật |
| Ngoại lệ | * 1. Cập nhật sản phẩm thất bại, tên sản phẩm đã tồn tại |

* Xóa sản phẩm

#### Bảng 10 Xóa sản phẩm

|  |  |
| --- | --- |
| Tên | Xóa sản phẩm |
| Tác nhân chính | Người quản lý |
| Tiền điều kiện | Đã đăng nhập vào tài khoản có quyền quản trị |
| Hậu điều kiện | Xóa sản phẩm thành công |
| Điều kiện tối thiểu | Chọn lại sản phẩm cần xóa |
| Sự kiện chính | 1. Hiển thị giao diện danh sách sản phẩm 2. Người quản lý chọn một sản phẩm và nhấn nút xóa 3. Hiển thị thông báo hỏi người quản lý có thật sự muốn xóa 4. Nhấn nút Ok 5. Hệ thống Xóa sản phẩm khỏi CSDL |
| Ngoại lệ | * 1. Cập nhật sản phẩm thất bại, tên sản phẩm đã tồn tại |

### Thiết kế CSDL

Thiết kế được sử dụng để thực hiện các chức năng cơ bản trong đồ án này, gồm các bảng với các thuộc tính như sau:

TAIKHOAN(**MaTK**, TenDangNhap, Password, Ten, Ho, Email, KichHoat)

THONGTINNGUOIDUNG(**MaTT**, GioiTinh, SDT, DiaChi1, DiaChi2, Tinh, MaTK)

LOAISP(**MaLoai**, TenLoai, MoTa)

SANPHAM(**MaSP**, TenSP, MoTa, Gia, KichThuoc, Hinh)

SANPHAM\_LOAISP(**MaLoai**, **MaSP**)

SUKIEN(**MaSK**, TenSK, NoiDung, ThoiGian)

HOADON(**MaHD**, NgayDatHang, GhiChu, MaKH)

CHITIETHOADON(**MaCT**, SoLuong, TongTien, GhiChu, MaHD, MaSP)

THONGTINLIENHE(**MaLH**, Email, SDT)

## Cài đặt giải pháp

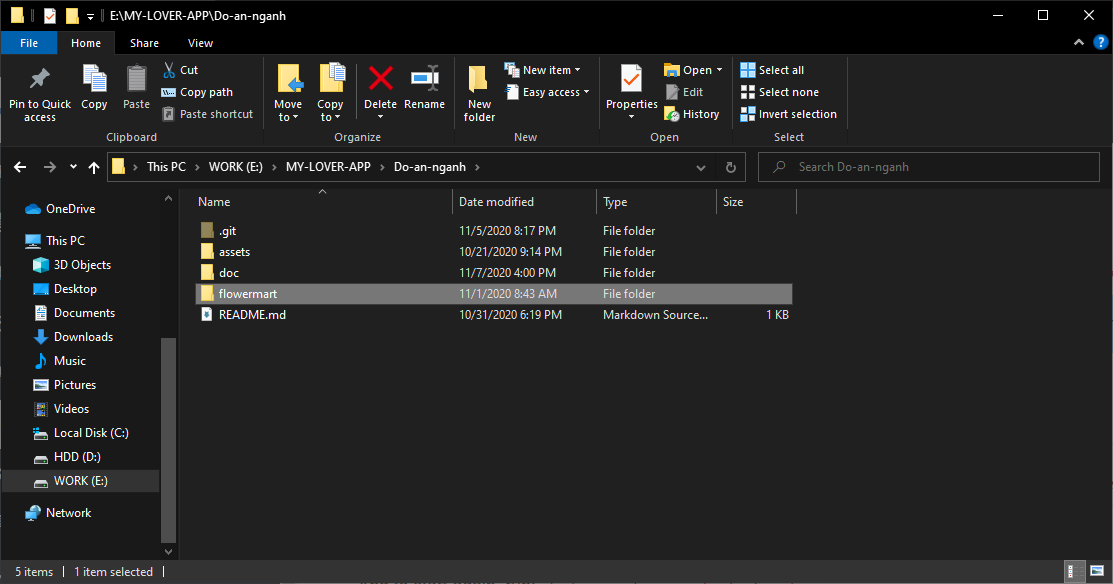
* + 1. Khởi tạo dự án bằng Jhipster

Trước hết cần cài đặt các công cụ cần thiết như sau để Jhipster hoạt động.

* Cài đặt Java 11.
* Cài đặt Nodejs (phiên bản LTS)
* Cài đặt Jhipster bằng câu lệnh: npm install -g generator-jhipster
* Cài đặt hệ quản trị CSDL PostgreSQL để phát triển ứng dụng với hệ quản trị CSDL này.

Sau khi đã cài đặt xong các công cụ trên, ta tiến hành khởi tạo dự án

1. Tạo thư mục có tên “flowermart”



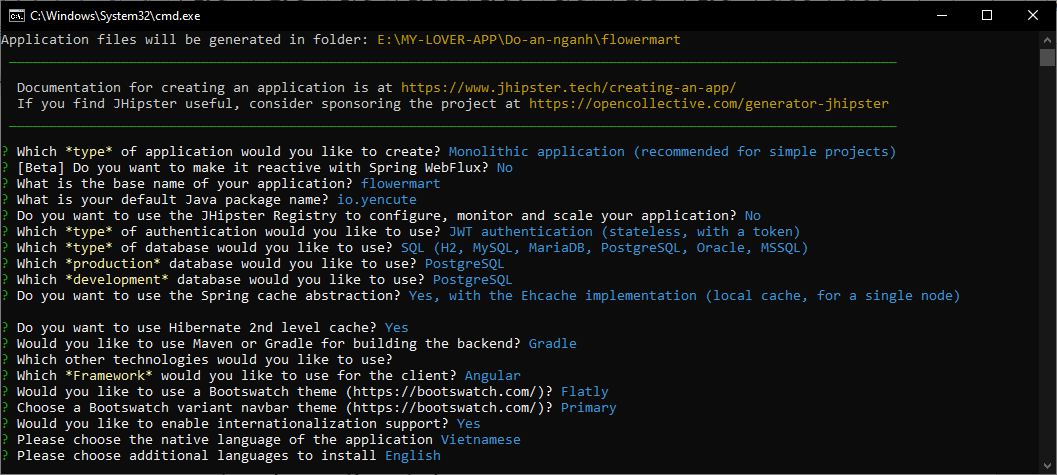
Hình 6 Tạo thư mục flowermart

1. Mở cmd trong thư mục vừa tạo và gõ “jhipster”



Hình 7 Khởi tạo dự án trong thư mục vừa tạo

1. Chọn các mục để tạo dự án

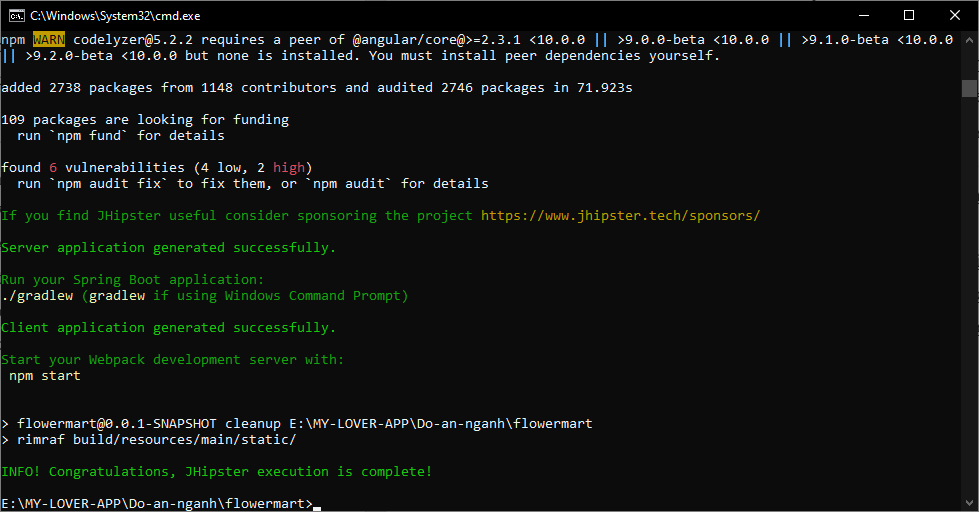


Hình 8 Các mục cần chọn để tạo dự án

Giải thích:

* + Ở đồ án này dùng kiến trúc nguyên khối nên chọn “Monolithic application”.
  + Không tạo reactive nên chọn “No” ở dòng tiếp theo
  + Đặt tên của dự án là “flowermart”
  + Đặt tên của Java package của dự án này là “io.yencute”
  + Không quản lý ứng dụng bằng Jhipster Registry nên chọn “No”
  + Chọn loại xác thực tài khoảng là “JWT authentication”
  + Chọn loại CSDL là “SQL”
  + Chọn hệ quản trị CSDL khi triển khai sản phẩm là “PostgreSQL”
  + Chọn hệ quản trị CSDL khi phát triển sản phẩm là “PostgreSQL”
  + Chọn Ehcache
  + Chấp nhận sử dụng Hibernate 2 level cache
  + Chọn công cụ tự động hóa quá trình build là “Gradle” vì Gradle tổng hợp toàn bộ các lỗi trong 1 lần đóng gói và nhiều tính năng khác thích hợp với ứng dụng cấp sản xuất.
  + Chọn “Angular” framework để lập trình giao diện
  + Chọn chủ để bootstrap là “Flatly”
  + Chọn màu thành navbar là “primary”
  + Chấp nhận dùng công cụ dịch tạo trang web đa ngôn ngữ (dùng i18n)
  + Chọn ngôn ngữ mặc định là “Tiếng Việt”.
  + Chọn ngôn ngữ tùy chọn là “Tiếng Anh”.
  + Chọn công cụ test nhanh là “Protractor”
  + Chọn “No” vì không cần sử dụng các trình tự động khác

1. Chờ jhipster khởi tạo dự án từ các mục đã chọn



Hình 9 Khởi tạo dự án thành công

1. Tạo database trong hệ quản trị cơ sở dữ liệu PostgreSQL

* Mở pgAdmin
* Tạo người dùng “flowermart\_admin” với password có quyền tạo database, như sau:

CREATE USER "flowermart\_admin" WITH PASSWORD 'root@123' CREATEDB;

* Tạo database “flowermart” và gán quyền cho user vừa tạo, như sau:

CREATE DATABASE flowermart

WITH

OWNER = flowermart\_admin

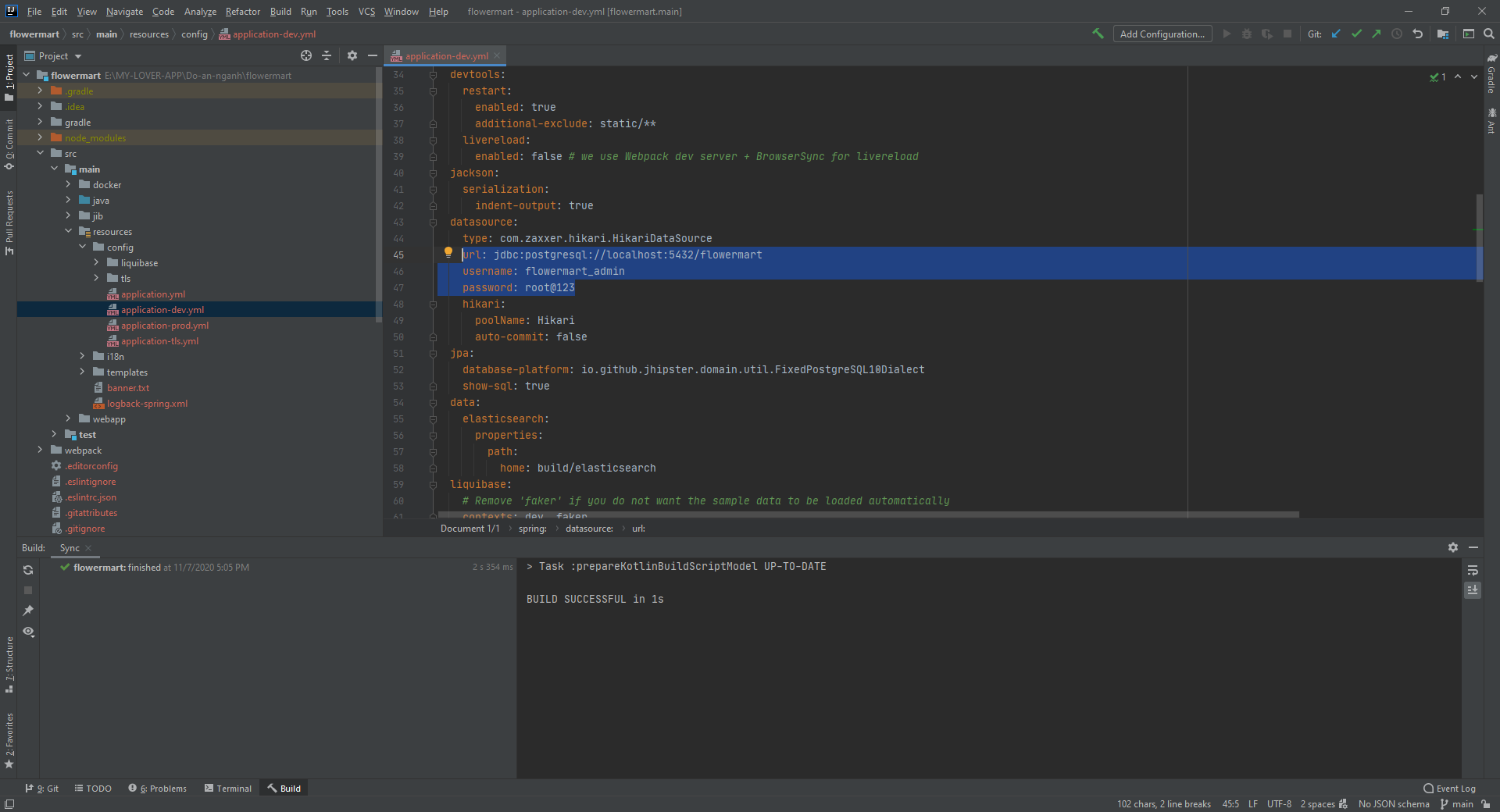
ENCODING = 'UTF8'

CONNECTION LIMIT = -1;

GRANT ALL ON DATABASE flowermart TO flowermart\_admin;

1. Cấu hình kết nối database vừa tạo và dự án

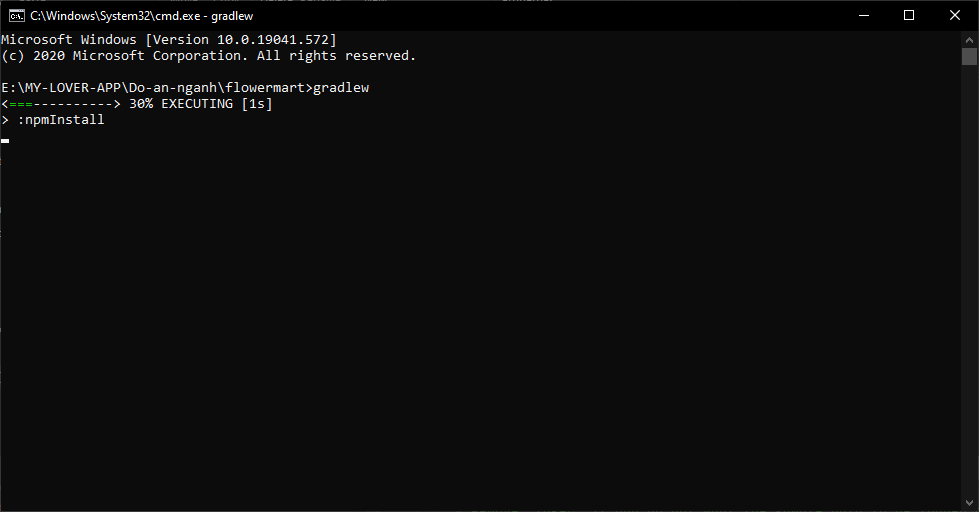
* Mở dự án vừa khởi tạo bằng Intelij IDEA 2020
* Mở file “src\main\resources\config\application-dev.yml” để chạy ứng dụng khi phát triển với database flowermart.
* Thay đổi các thông tin trong datasource thành tên User, password và dường dẫn tới database flowermart, như sau:



Hình 10 Cập nhật thông tin cấu hình kết nối database

1. Chạy dự án trong môi trường phát triển

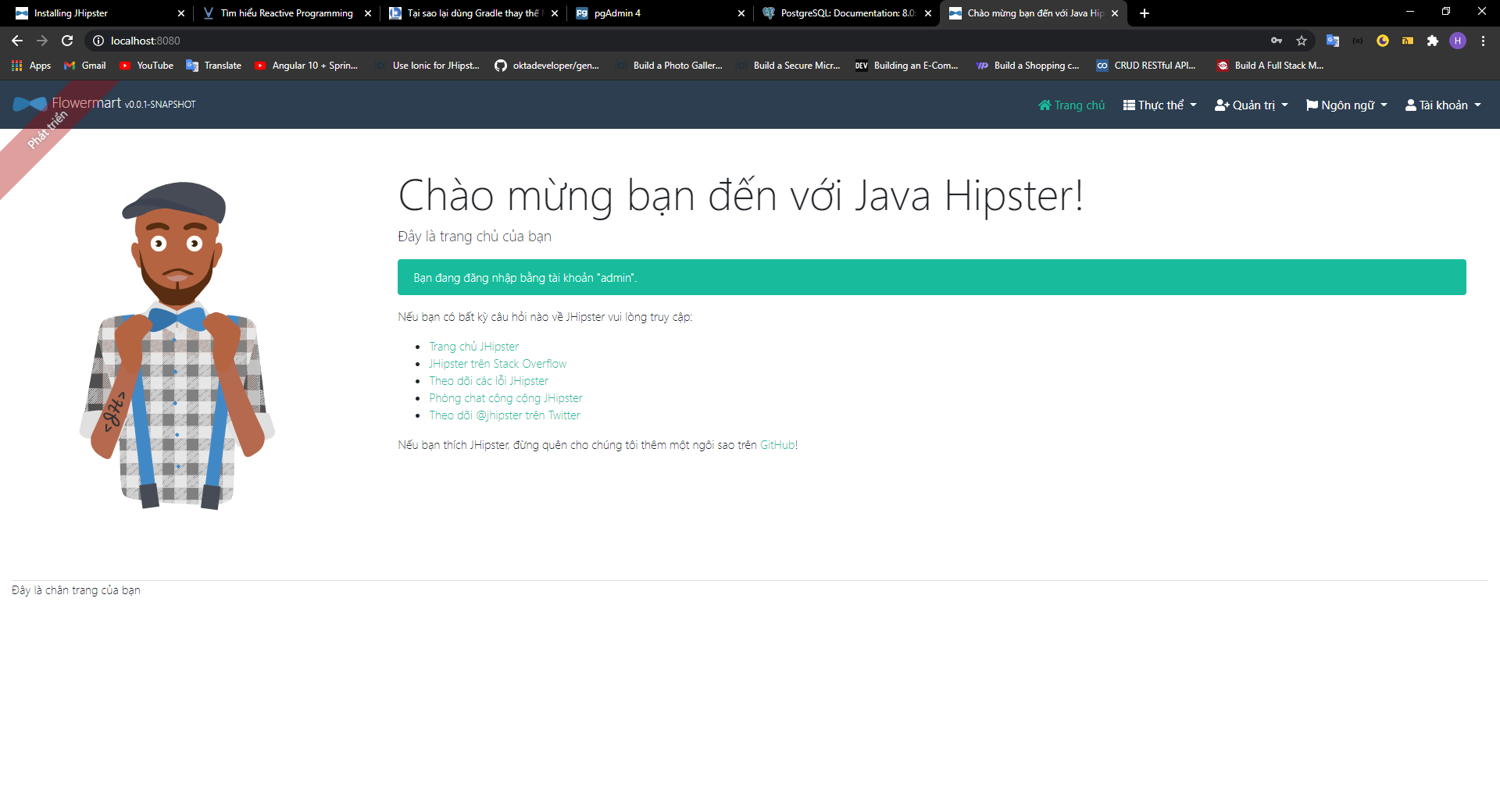
* Mở cmd trong dự án flowermart
* Gõ “gradlew” và nhấn enter



Hình 11 Chạy dự án trong môi trường phát triển

* Giao diện mặc định khi khởi tạo dự án

Khi khởi tạo dự án xong jhipster sẽ giải quyết sẳn cho chúng ta những vấn đề cơ bản như giao diện home, chức năng xác thực đăng nhập,…



Hình 12 Giao diện mặc định của dự án khi vừa khởi tạo

* + 1. Sử dụng JDL để tạo các chức năng cơ bản cho dự án

JDL – Jhipster Domain Language là một ngôn ngữ của Jhipster dùng để mô tả các ứng dụng, triển khai, thực thể và các mối quan hệ giữa các thực thể trong một tệp duy nhất (hoặc nhiều hơn) với cú pháp thân thiện với người dùng.

Đồ án này sử dụng JDL để tự động hóa việc tạo ra các chức năng thêm xóa sửa cơ bản. Ta tạo 1 file JDL với nội dung như sau:

enum Gender {

    MALE, FEMALE, OTHER

}

entity ContactInfo {

    email String required pattern(/^[^@\s]+@[^@\s]+\.[^@\s]+$/)

    phone String required

}

entity NewsAndEvent {

    title String required

    content TextBlob

    time Instant

}

entity Customer {

    gender Gender required

    phone String required

    addressLine1 String required

    addressLine2 String

    city String required

    country String required

}

enum Size {

    NORMAL, SPECIAL

}

entity Product {

    name String required

    description TextBlob

    price BigDecimal required min(0)

    size Size required

    image ImageBlob

}

entity Category {

    name String required

    description TextBlob required

}

enum OrderStatus {

    COMPLETED, PENDING, CANCELLED

}

entity Bill {

    placedDate Instant required

    status OrderStatus required

}

enum BillItemStatus {

    AVAILABLE, OUT\_OF\_STOCK

}

entity BillItem {

    quantity Integer required min(0)

    totalPrice BigDecimal required min(0)

    status BillItemStatus required

}

relationship OneToOne {

    Customer{*user*(*login*)} to User // thêm ràng buộc ngay đây

}

relationship ManyToMany {

    Product{*Category*(*name*)} to Category{*product*}

}

relationship OneToMany {

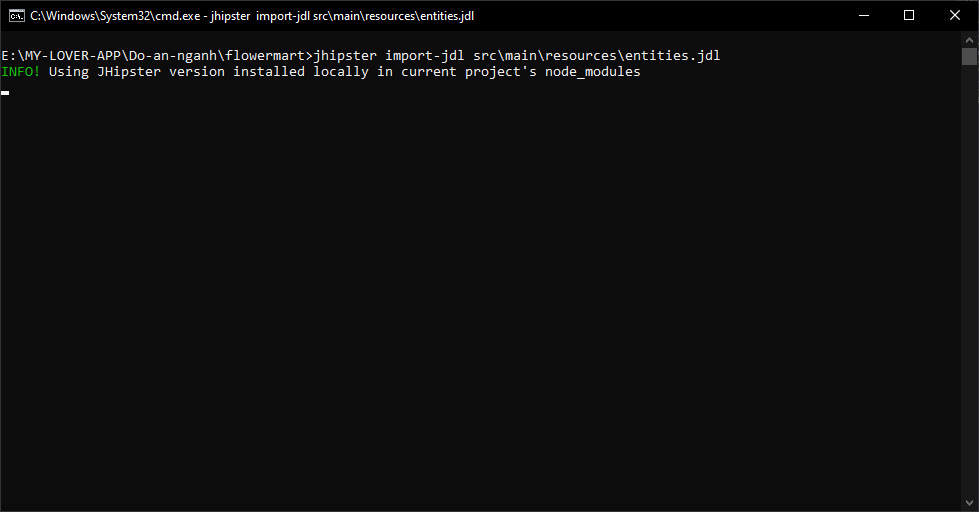
    Customer{*Bill*} to Bill{*Customer*}

    Bill{*billItem*} to BillItem{*bill*}

    Product{*billItem*} to BillItem{*product*(*name*)}

}

Hình File JDL



Hình Nhập file JDL vừa tạo vào dự án

* + 1. Cài đặt môi trường kiểm thử
    2. Một số hình ảnh của phẩn mềm

# Chương 4: KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

## 4.1. Kết quả đạt được

## 4.2. Hạn chế

* Tiếp cận nhiều công nghệ mới cùng lúc nên có thể còn những hạn chế, chưa thể phát huy tối đa hiệu quả mà các công nghệ đó mang lại.

## 4.3. Hướng phát triển

# TÀI LIỆU THAM KHẢO

[1].TypeScript – Overview, <https://www.tutorialspoint.com/typescript/typescript_overview.htm>,

[2]. Stackjava(19/12/2017), Tổng quan về Spring Framework, <https://viblo.asia/p/tong-quan-ve-spring-framework-YWOZryEyKQ0>

[3], HoangManh(20/03/2019), Giới thiệu về JHipster Stack, <https://viblo.asia/p/gioi-thieu-ve-jhipster-stack-6J3ZgmdL5mB>, https://www.jhipster.tech/

[4]. Tài liệu của SpringBoot, <https://docs.spring.io/spring-data/rest/docs/current/reference/html/#repository-resources.collection-resource>

[5]. Diễn đàn trao đổi code Stackoverflow <https://stackoverflow.com/>