

# **THỰC HÀNH LẬP TRÌNH ỨNG DỤNG JAVA**

Ấn bản 2023

# MỤC LỤC

<b>MỤC LỤC .....</b>	<b>I</b>
<b>MỤC LỤC HÌNH ẢNH .....</b>	<b>II</b>
<b>HƯỚNG DẪN .....</b>	<b>V</b>
<b>BÀI 1 LÀM QUEN VỚI SPRING FRAMEWORK.....</b>	<b>VII</b>
1.1 Cơ bản về Spring Framework và các cài đặt cần thiết.....	VII
1.1.1 Yêu cầu.....	VII
1.1.2 Các bước cài đặt Java JDK .....	VIII
1.1.3 Cài đặt Laragon (Quản lý CSDL MySQL).....	X
1.1.4 Cài đặt IntelliJ IDEA.....	XIII
1.1.5 Download Template Spring từ spring.io.....	XVI
1.1.6 Chạy trang web đầu tiên .....	XXVII
TÓM TẮT .....	XXXII
<b>BÀI 2 VIẾT CHƯƠNG TRÌNH QUẢN LÝ SÁCH .....</b>	<b>33</b>
2.1 Xây dựng trang Book List .....	34
2.1.1 Tạo view layout trong ứng dụng .....	37
2.2 Xây dựng trang thêm sách .....	40
2.2.1 Chính sửa BookController.java.....	40
2.2.2 Thêm View hiển thị.....	42
2.3 Xây dựng trang sửa sách .....	47
2.4 Thực hiện chức năng xóa sách .....	50
TÓM TẮT .....	51
CÂU HỎI ÔN TẬP .....	52

## MỤC LỤC HÌNH ẢNH

Hình 1. Cài đặt Java JDK X64 Installer .....	VIII
Hình 2. Install Java SE to path .....	IX
Hình 3. Successfully installed Java JDK.....	IX
Hình 4. Download Edition Laragon - full.....	X
Hình 5. Setup Laragon path .....	XI
Hình 6. Setup Laragon runs extremely .....	XI
Hình 7. Ready to Install Laragon .....	XII
Hình 8. Completing the Laragon Setup .....	XII
Hình 9. Download IntelliJ IDEA Community Edition.....	XIII
Hình 10.Next Setup IntelliJ IDEA .....	XIV
Hình 11. Installation Options IntelliJ IDEA.....	XIV
Hình 12. Completing IntelliJ IDEA Community .....	XV
Hình 13. IntelliJ IDEA use agreement.....	XV
Hình 14. Giao diện bắt đầu của IntelliJ IDEAA .....	XVI
Hình 15. Open Spring Framework tại spring.io.....	XVII
Hình 16. Khởi động Quikstart you Project .....	XVII
Hình 17. Lựa chọn cấu hình cho spring initializr .....	XVIII
Hình 18. thêm các Dependencies trong spring.io.....	XIX
Hình 19.Giải nén và mở Spring Teamplate bằng IntelliJ IDEA.....	XX
Hình 20. Bước cài đặt các Dependencies cần thiết.....	XXI
Hình 21. Lựa chọn Project Structure.....	XXII

Hình 22. Lựa chọn phiên bản JDK .....	XXII
Hình 23. Lựa chọn phiên bản JDK - 19 .....	XXIII
Hình 24. Bổ sung đoạn code cấu hình cho application.properties .....	XXIV
Hình 25. Build project chương trình.....	XXIV
Hình 26. Run chương trình tại file DemoApplication .....	XXV
Hình 27. Giao diện mặc định đăng nhập .....	XXVI
Hình 28. Tạo package controller .....	XXVII
Hình 29.Tạo file HomeController.java trong package controller .....	XXVII
Hình 30.Chỉnh sửa nội dung HomeController.java .....	XXVIII
Hình 31. Tạo thư mục home .....	XXIX
Hình 32. Hiển thị ngay phía dưới thư mục templates .....	XXIX
Hình 33. Tạo file index.html.....	XXX
Hình 34. Chỉnh sửa file index.html .....	XXXI
Hình 35. Giao diện trang chủ .....	XXXI
Hình 1. Thêm 1 Package với tên entity .....	34
Hình 2. Thêm 1 file Book.java và thêm các thuộc tính cho book .....	34
Hình 3. Thêm 1 file BookController.java trong packege controller .....	35
Hình 4. Tạo file AppConfig.java .....	36
Hình 5. Tải thư viện Bootstrap .....	37
Hình 6. Thêm thư viện Bootstrap vào dự án .....	37
Hình 7. Tạo file layout.html .....	38
Hình 8. Tạo view hiển thị danh sách.....	39
Hình 9. Kết quả, trang Book List.....	40
Hình 10. Chỉnh sửa BookController.java cho chức năng Add .....	40

Hình 11. Tạo giao diện View hiển thị, tạo file add.html .....	42
Hình 12. Chính sửa đường dẫn các các menu ở trang layout.html.....	43
Hình 13. Chính sửa nội dung trang chủ tại index.html.....	44
Hình 14. Giao diện trang thêm book .....	45
Hình 15. Trang danh sách Book sau khi thêm Book mới .....	46
Hình 16. Thêm file edit.html và bổ sung code.....	47
Hình 17. Thêm hàm editbook() vào BookController .....	48
Hình 18. Giao diện Book List với nút Edit .....	49
Hình 19. Giao diện trang chỉnh sửa Book .....	49
Hình 20. Kết quả sau khi chỉnh sửa thành công, trả về Book List.....	50
Hình 21. Thêm hàm tên deleteBook() bên trong file BookController.java ...	50
Hình 22.Giao diện delete Book .....	51

# HƯỚNG DẪN

## MÔ TẢ MÔN HỌC

Học phần được xây dựng với các nội dung: kiến thức tổng quan về lập trình ứng dụng Java; cách thức phát triển ứng dụng web dùng Spring Framework

## NỘI DUNG MÔN HỌC

- **Bài 1:** Làm quen với Spring Framework
- **Bài 2:** Viết trang Web quản lý sách
- ...

## KIẾN THỨC TIỀN ĐỀ

Môn học thực hành lập trình web đòi hỏi sinh viên có nền tảng về kỹ thuật lập trình, lập trình hướng đối tượng, tiếp xúc nhiều với internet.

## YÊU CẦU MÔN HỌC

Người học cần đi học đầy đủ, đọc các nội dung sẽ được học trước khi đến lớp, làm các bài tập về nhà và đảm bảo thời gian tự học ở nhà.

## CÁCH TIẾP NHẬN NỘI DUNG MÔN HỌC

Để học tốt môn này, sinh viên cần ôn tập các bài đã học, trả lời các câu hỏi và làm đầy đủ bài tập; đọc trước bài mới và tìm thêm các thông tin liên quan đến bài học.

Đối với mỗi bài học, sinh viên đọc trước mục tiêu và tóm tắt bài học, sau đó đọc nội dung bài học. Kết thúc mỗi ý của bài học, sinh viên trả lời câu hỏi ôn tập và kết thúc toàn bộ bài học, sinh viên làm các bài tập.

## PHƯƠNG PHÁP ĐÁNH GIÁ MÔN HỌC

Môn học được đánh giá như sau:

- Điểm thực hành (100%): Hình thức thi thực hành, phù hợp với quy chế đào tạo và tình hình thực tế tại nơi tổ chức học tập.

# BÀI 1 LÀM QUEN VỚI SPRING FRAMEWORK

Bài này giúp người học nắm được các nội dung sau:

- Spring Framework là gì?
- Cài đặt môi trường và công cụ để phát triển ứng dụng Web bằng Spring Framework
- Lập trình cơ bản ứng dụng Website với Spring Framework

## 1.1 Cơ bản về Spring Framework và các cài đặt cần thiết

### 1.1.1 Yêu cầu

Máy tính phải cài đặt:

- Java JDK (Java Development Kit)
- Cài đặt Laragon (Quản lý CSDL MySQL)
- Cài đặt IntelliJ IDEA.

Máy tính chạy trên nền tảng hệ điều hành:

- Windows 7
- Windows 8
- Windows 10
- Windows 11.

## 1.1.2 Các bước cài đặt Java JDK

Java JDK (Java Development Kit) là một bộ công cụ để phát triển các ứng dụng Java. Java JDK cung cấp một số công cụ để biên dịch và thực thi mã Java, cũng như các thư viện và tài liệu hỗ trợ.

Để cài đặt Java JDK:

Google: JAVA JDK download

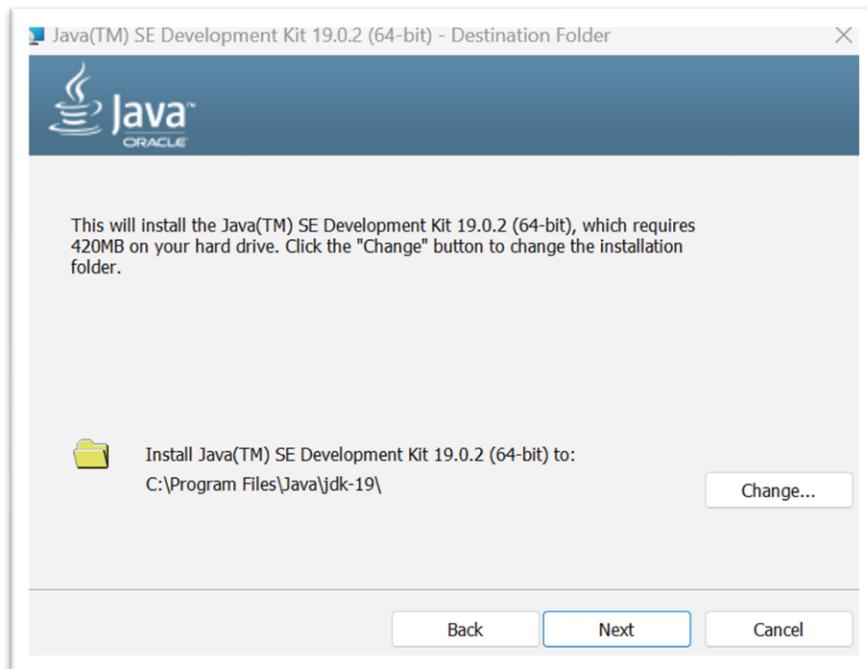
Hoặc truy cập đường dẫn sau:

<https://www.oracle.com/eg/java/technologies/downloads/>

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.		
Linux	macOS	Windows
Product/file description	File size	Download
x64 Compressed Archive	179.13 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.zip">https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.zip</a> ( sha256 )
x64 Installer	158.91 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.exe">https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.exe</a> ( sha256 )
x64 MSI Installer	157.76 MB	<a href="https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.msi">https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.msi</a> ( sha256 )

Hình 1. Cài đặt Java JDK X64 Installer

Cài đặt Java JDK như hướng dẫn, chọn Next..



Hình 2. Install Java SE to path

Quá trình cài đặt thành công sẽ như bên dưới.



Hình 3. Successfully installed Java JDK

### 1.1.3 Cài đặt Laragon (Quản lý CSDL MySQL)

Laragon cung cấp tính năng quản lý CSDL (Cơ sở dữ liệu) MySQL thông qua trình quản lý cơ sở dữ liệu phpMyAdmin.

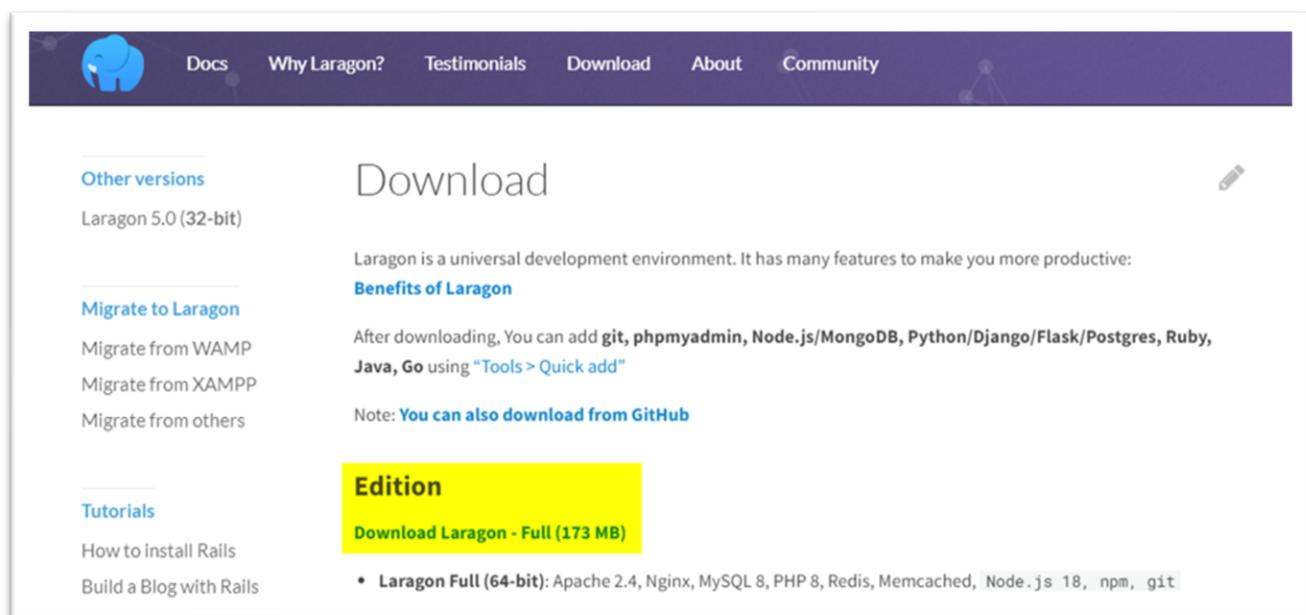
Khi bạn cài đặt và khởi chạy Laragon, nó cũng cài đặt một máy chủ MySQL đồng thời để bạn có thể tạo, quản lý và sử dụng các cơ sở dữ liệu MySQL.

Để cài đặt Laragon:

Google: download Laragon

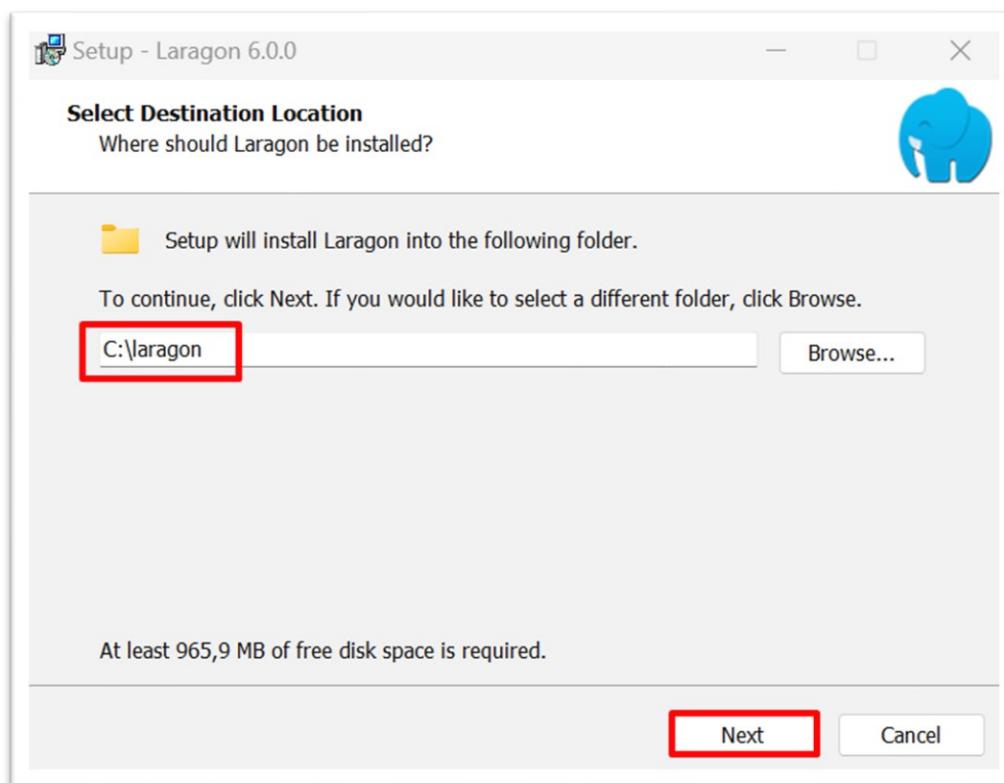
Hoặc truy cập đường dẫn sau: <https://laragon.org/download/index.html>

Tại trang Download, chọn phiên bản Download Laragon – Full (173 MB)



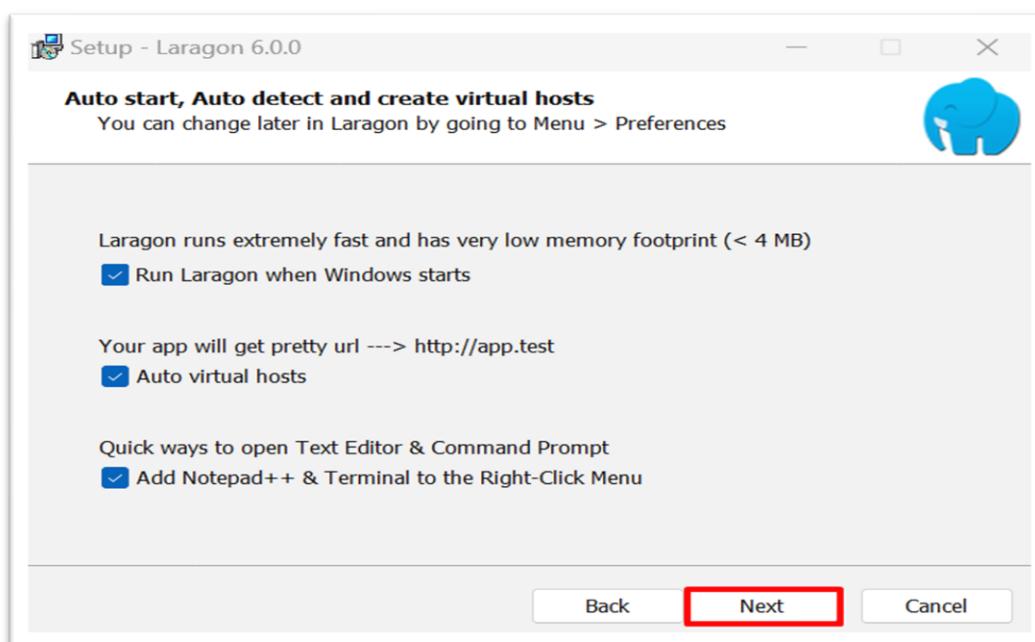
Hình 4. Download Edition Laragon - full

Tiếp tục, chọn nơi lưu trữ và nhấn Next

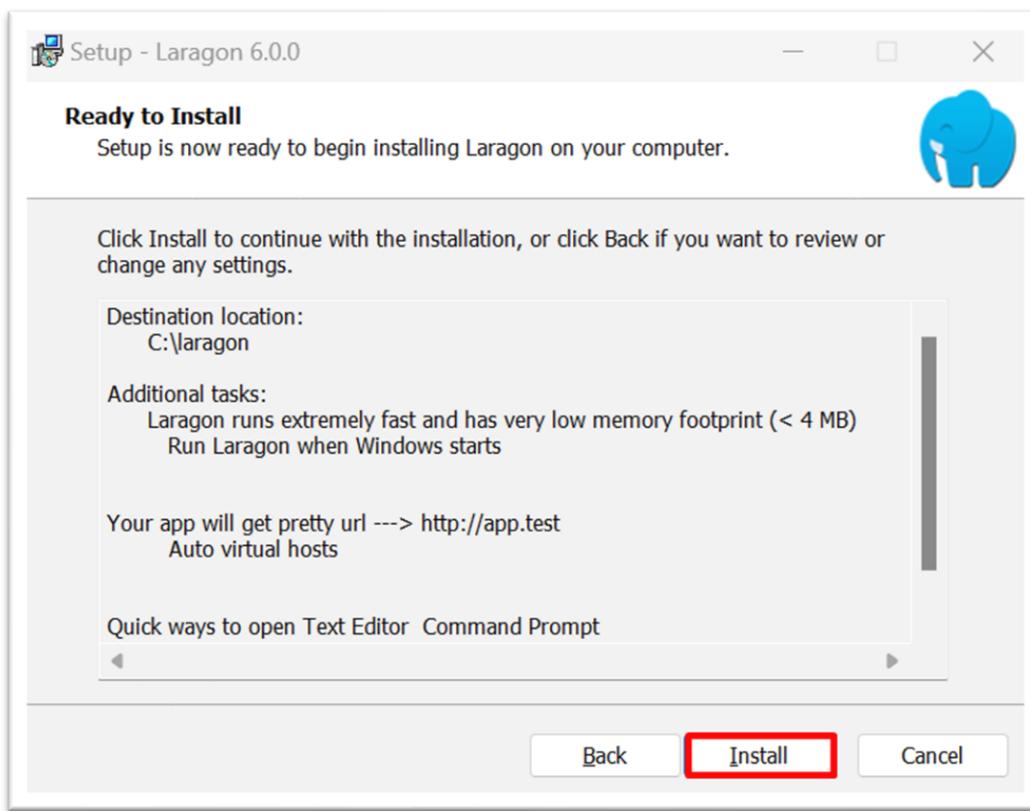


Hình 5. Setup Laragon path

Tích chọn 3 dấu tích và nhấn Next



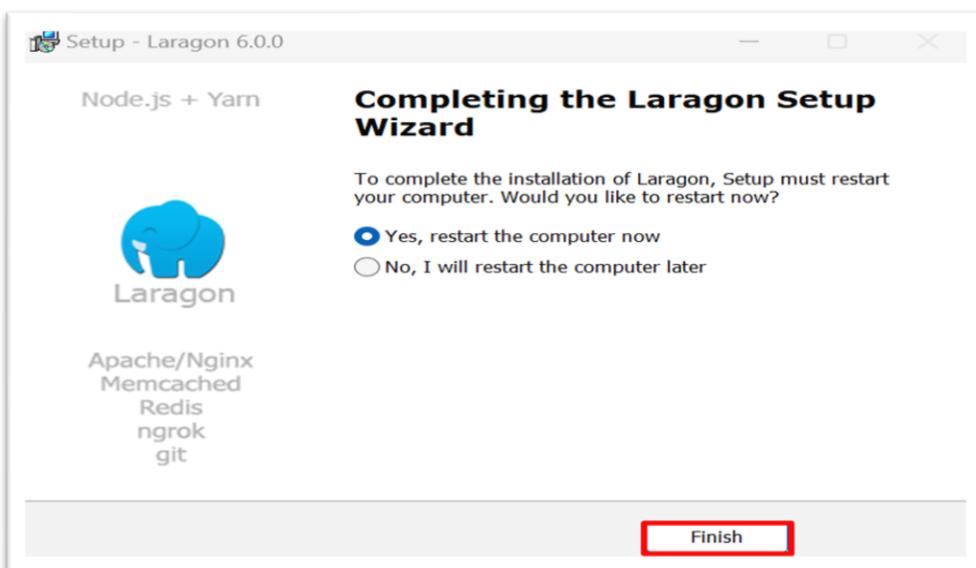
Hình 6. Setup Laragon runs extremely



Hình 7. Ready to Install Laragon

Quá trình cài đặt hoàn thành, máy tính sẽ khởi động lại.

Kiểm tra trên máy tính xuất hiện Laragon.



Hình 8. Completing the Laragon Setup

### 1.1.4 Cài đặt IntelliJ IDEA

IntelliJ IDEA là một IDE (Integrated Development Environment) cho lập trình viên phát triển các ứng dụng sử dụng ngôn ngữ lập trình Java, Scala, Kotlin và một số ngôn ngữ khác. IntelliJ IDEA được phát triển bởi JetBrains và được coi là một trong những IDE tốt nhất cho lập trình Java.

Ngoài ra, IntelliJ IDEA còn hỗ trợ tính năng đa nền tảng, cho phép lập trình viên phát triển ứng dụng trên nhiều hệ điều hành khác nhau bao gồm Windows, MacOS và Linux.

Để cài đặt Laragon:

Google: Download IntelliJ IDEA Community

Hoặc truy cập đường dẫn sau:

<https://www.jetbrains.com/idea/download/#section=windows>

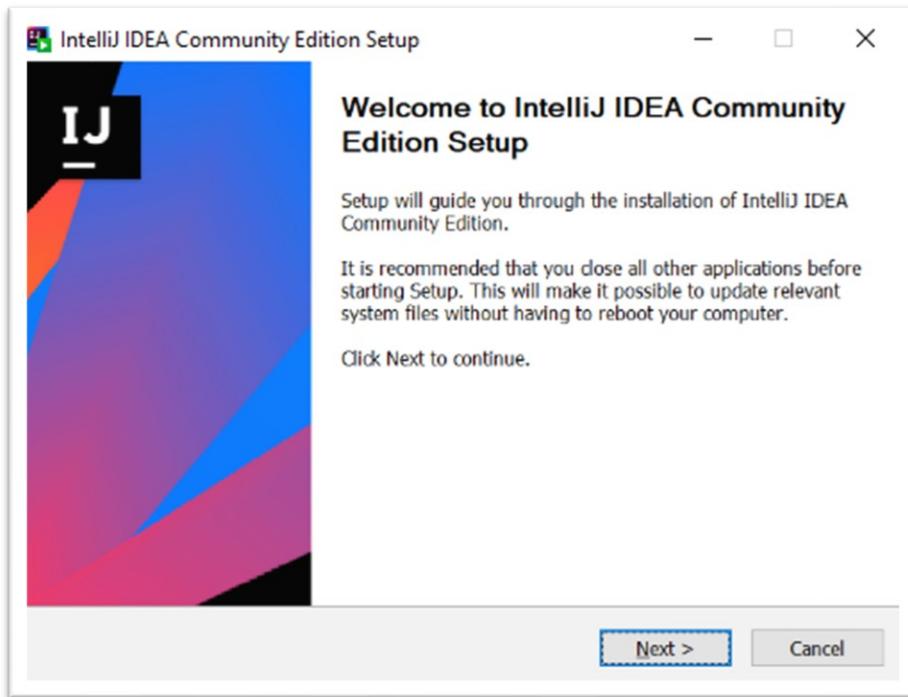
Tại trang download IntelliJ IDEA, chọn phiên bản **Community Edition** bên phải.

Phiên bản Community Edition này miễn phí cho người dùng.



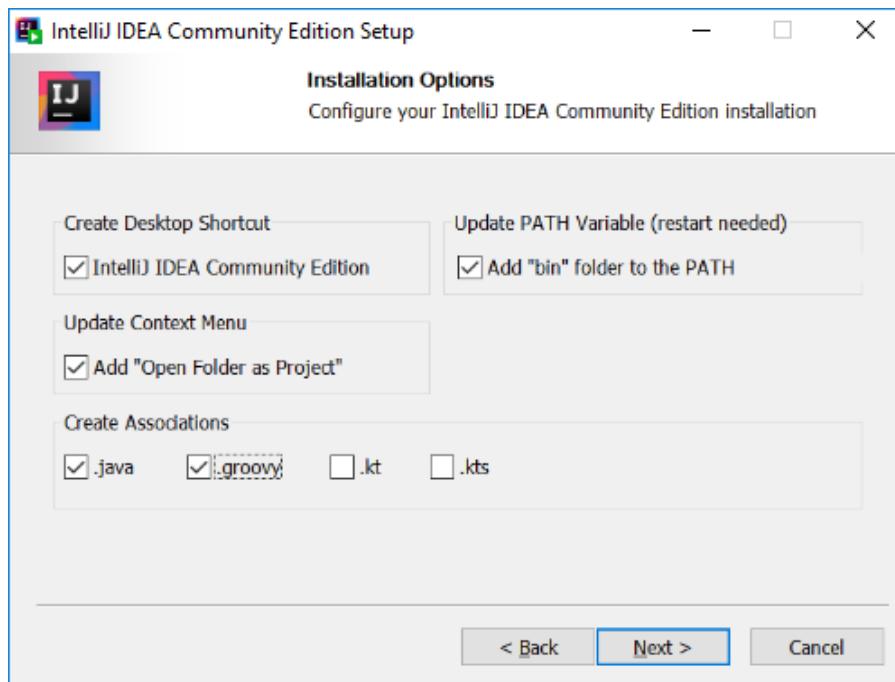
Hình 9. Download IntelliJ IDEA Community Edition

Sau khi tải về, nhấn cài đặt như hướng dẫn bên dưới. Tiếp theo, Nhấn **Next**



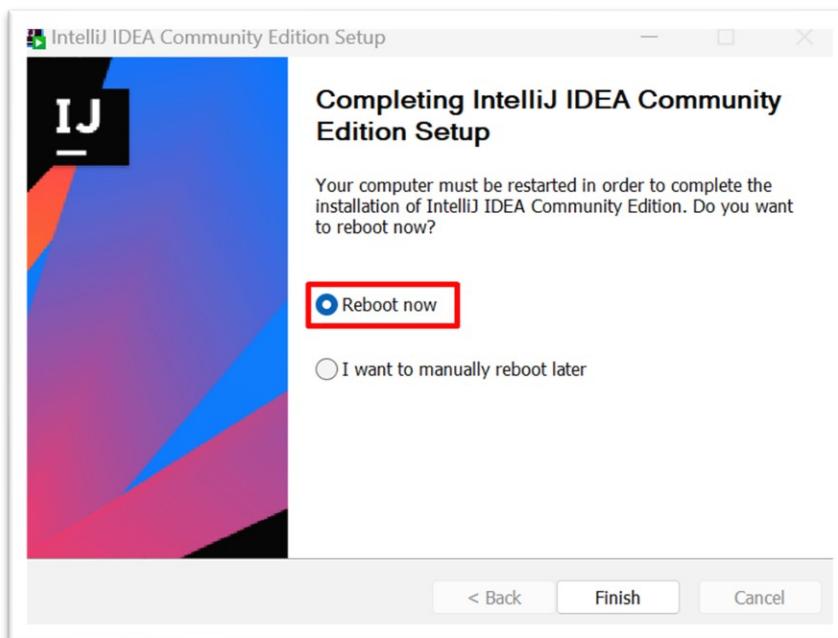
Hình 10. Next Setup IntelliJ IDEA

Ở bước này, tích chọn 4 lựa chọn như ảnh bên dưới, sau đó nhấn **Next**.



Hình 11. Installation Options IntelliJ IDEA

Tiếp theo, Tích chọn **Reboot now**, chọn **Finish** để hoàn tất cài đặt.

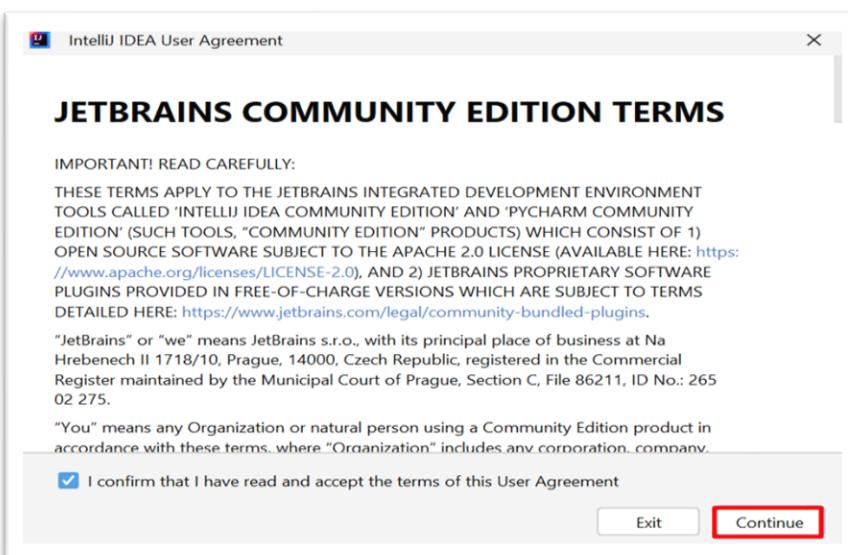


Hình 12. Completing IntelliJ IDEA Community

Sau khi hoàn tất cài đặt, tìm phần mềm trên máy và khởi động.

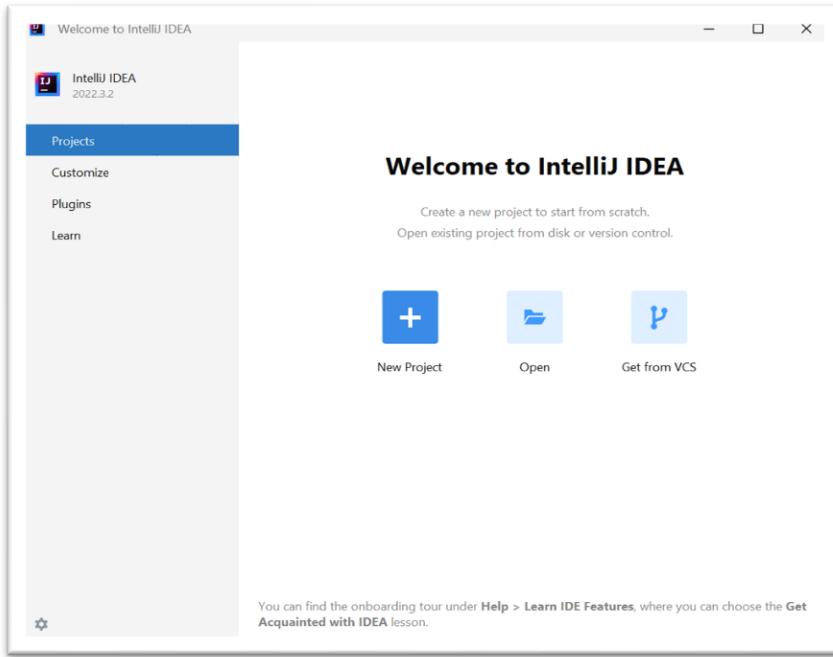


Chọn **Continue**,



Hình 13. IntelliJ IDEA use agreement

Giao diện bắt đầu của **IntelliJ IDEA**, chúng ta sẽ quay lại nó sau.



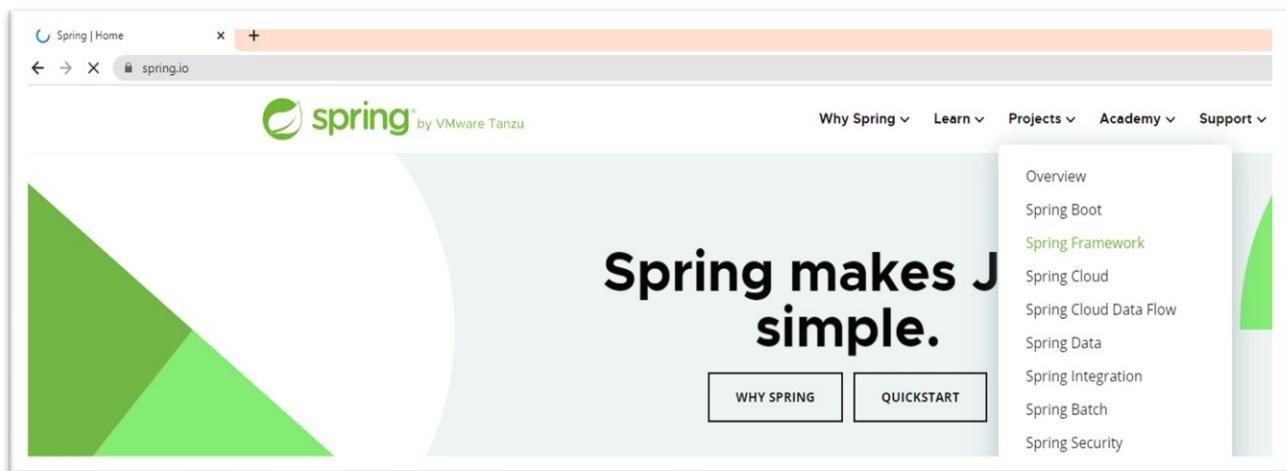
Hình 14. Giao diện bắt đầu của IntelliJ IDEA

## 1.1.5 Download Template Spring từ spring.io

Template Spring từ spring.io là một dự án mẫu (template project) sẵn có cho việc phát triển ứng dụng sử dụng framework Spring. Dự án mẫu này cung cấp một số bộ khung cơ bản cho việc phát triển ứng dụng, giúp giảm thời gian và công sức của lập trình viên khi bắt đầu phát triển một dự án mới.

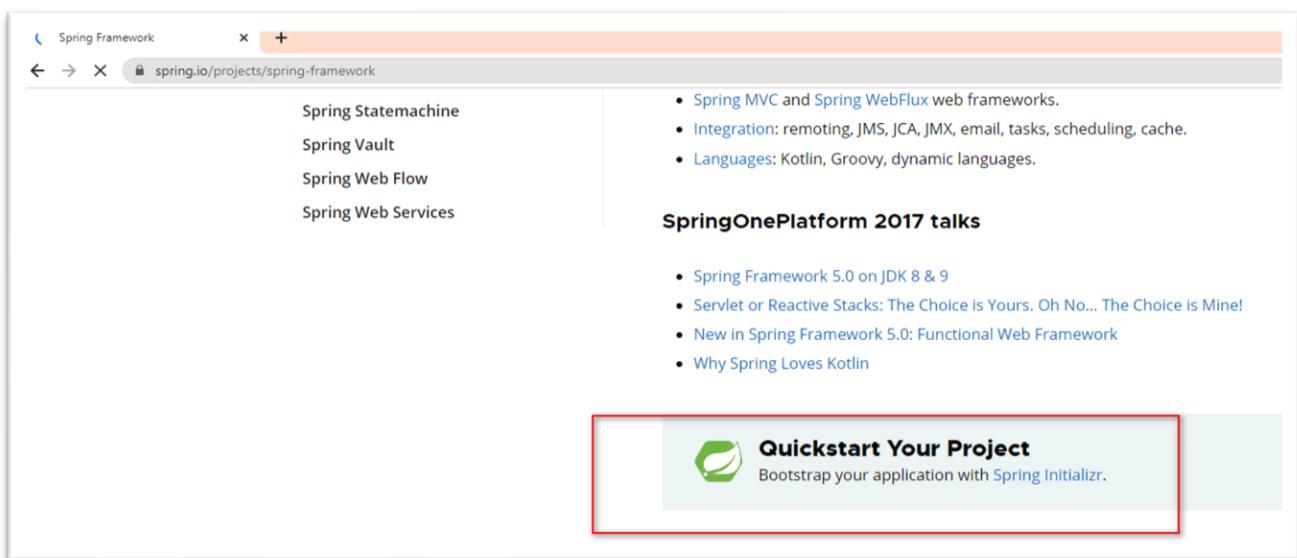
Template Spring bao gồm một số module cơ bản như Spring Boot, Spring Security, Spring Data, Spring MVC, và nhiều hơn nữa.

Truy cập trang web: <https://spring.io/> , tại mục **Project** chọn **Spring Framework**



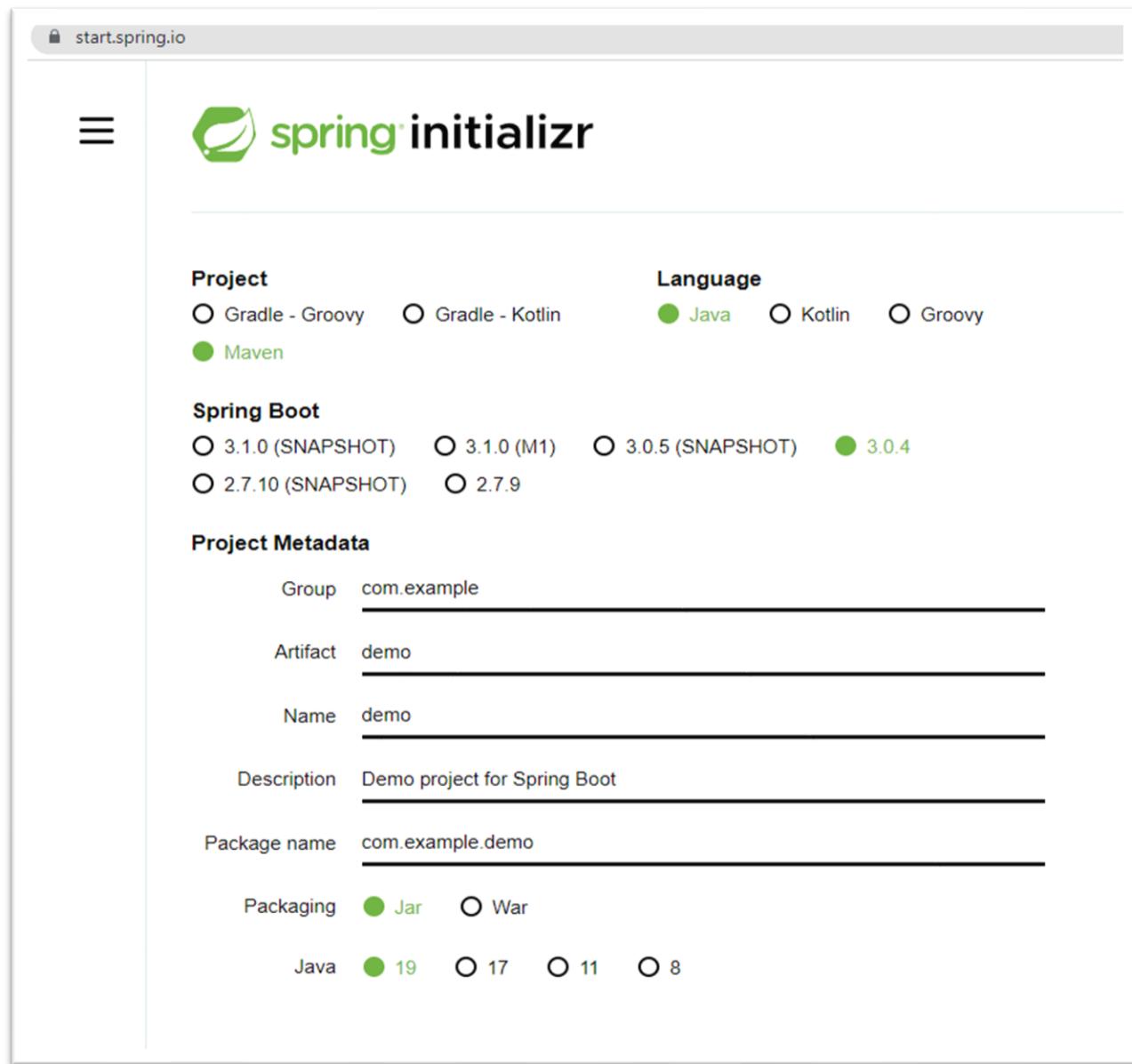
Hình 15. Open Spring Framework tại [spring.io](http://spring.io)

Tiếp theo, lướt xuống dưới chọn **Spring Initializr**.



Hình 16. Khởi động Quickstart your Project

Lựa chọn cấu hình, tích chọn các lựa chọn như ảnh bên dưới,



Hình 17. Lựa chọn cấu hình cho spring initializr

Tiếp theo nhấn Tiếp, nhấn **Add Dependencies...**



Cửa sổ mở ra, tìm và thêm các **Dependencies** như ảnh bên dưới.

The screenshot shows a list of dependencies in a Spring Boot DevTools interface. Each dependency has a green 'ADD' button and a red minus sign button to its right. The dependencies listed are:

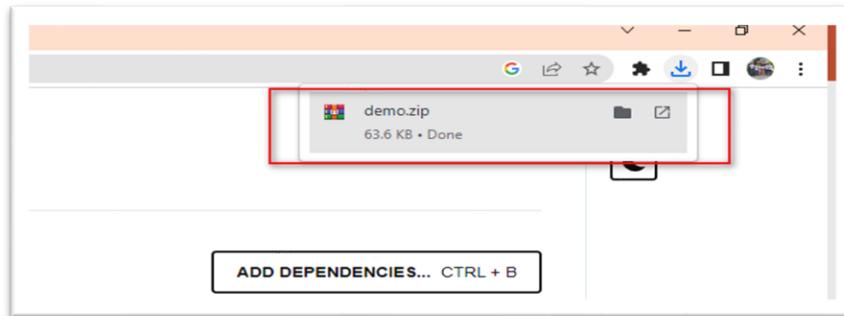
- Spring Boot DevTools** **DEVELOPER TOOLS**  
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.
- Lombok** **DEVELOPER TOOLS**  
Java annotation library which helps to reduce boilerplate code.
- Spring Web** **WEB**  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
- Thymeleaf** **TEMPLATE ENGINES**  
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.
- Spring Security** **SECURITY**  
Highly customizable authentication and access-control framework for Spring applications.
- Spring Data JPA** **SQL**  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
- MySQL Driver** **SQL**  
MySQL JDBC driver.

Hình 18. thêm các Dependencies trong spring.io

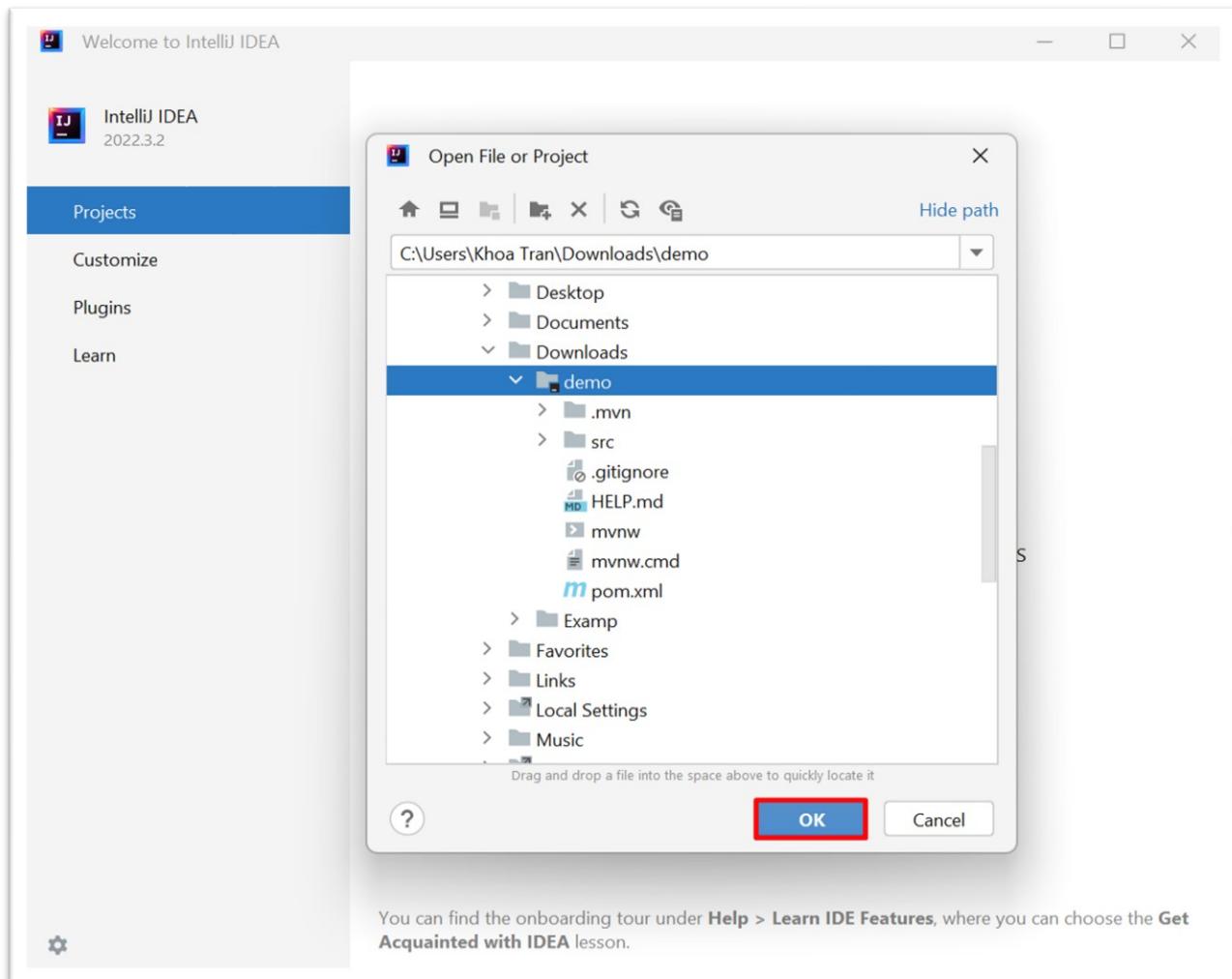
Sau đó, nhấn **GENERATE** để tải về



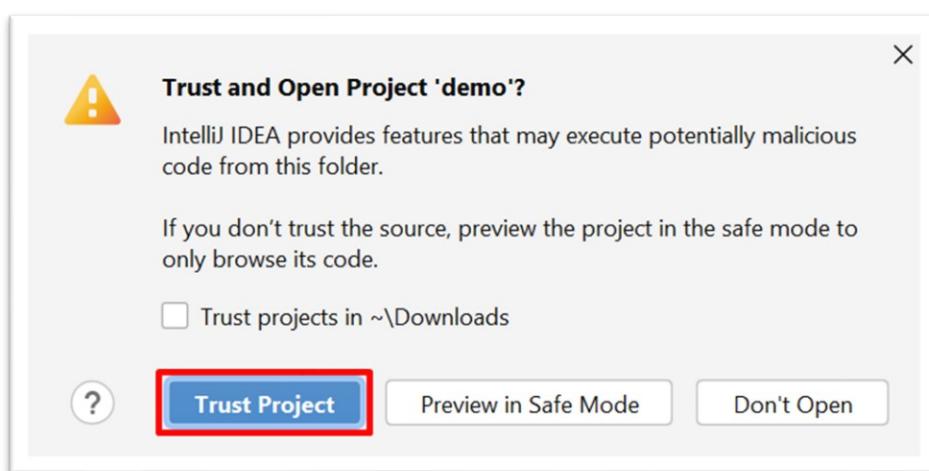
Một file **demo.zip** được tải về máy tính,



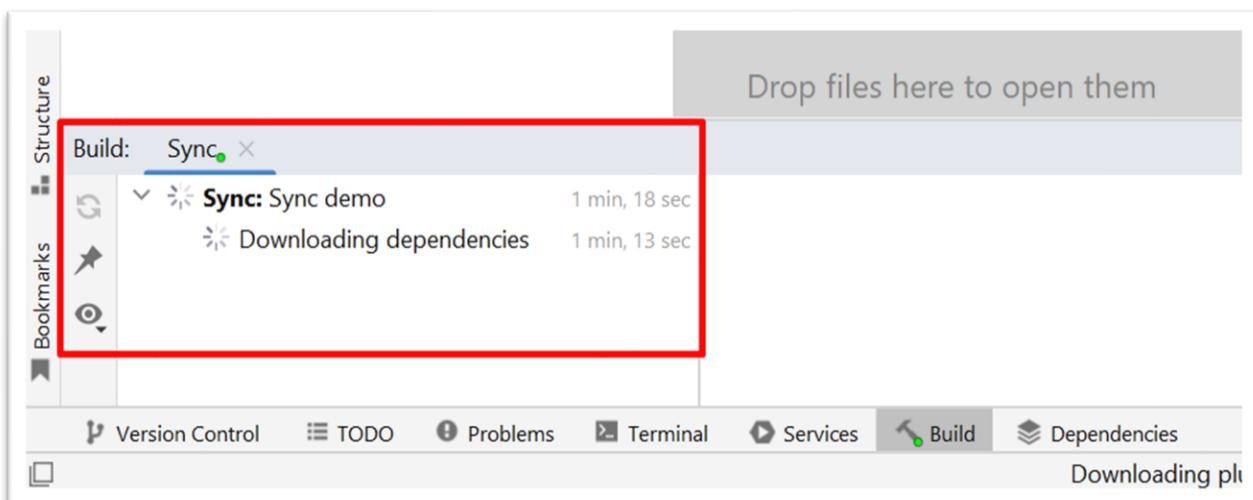
Giải nén thư mục **Demo.zip Template** đã tải về và mở Spring Teamplate bằng **IntelliJ IDEA**.



Hình 19. Giải nén và mở Spring Teamplate bằng IntelliJ IDEA.

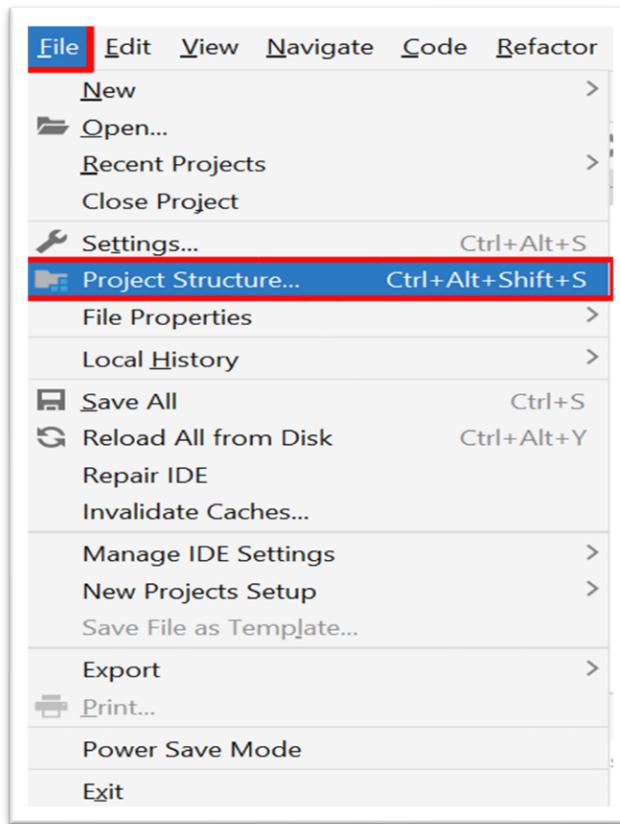


Tiến trình tải các **Dependencies** cần thiết. Bạn đợi cho các cài đặt hoàn tất, bao gồm sẽ cài đặt đường dẫn tới thư mục JDK đã cài ở bước trước.



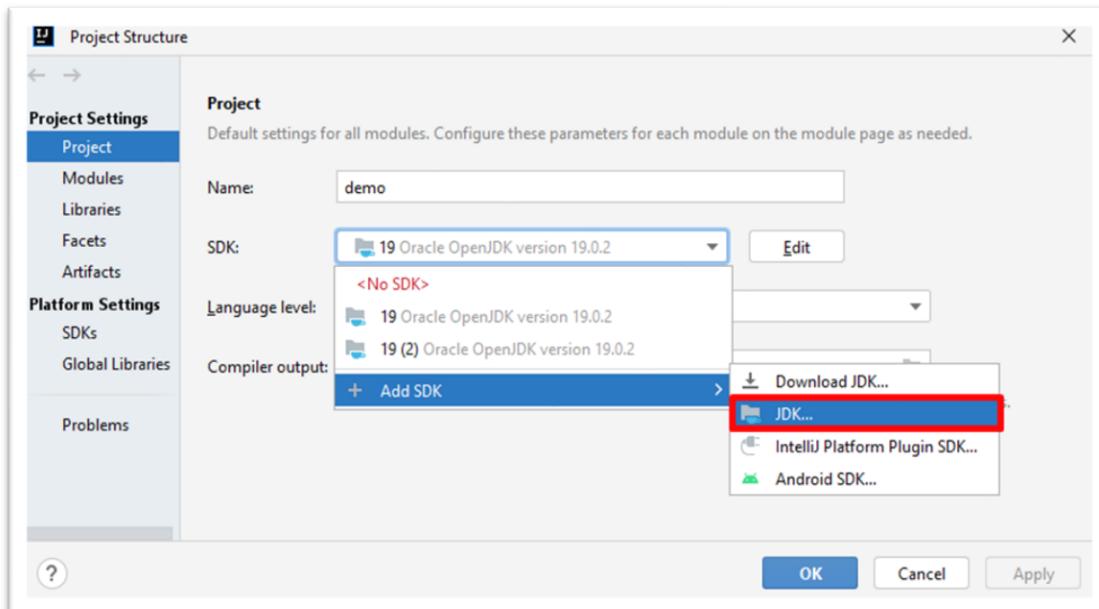
Hình 20. Bước cài đặt các Dependencies cần thiết

Tại thẻ **File** → chọn **Project Structure...** Để tạo cấu trúc dự án.



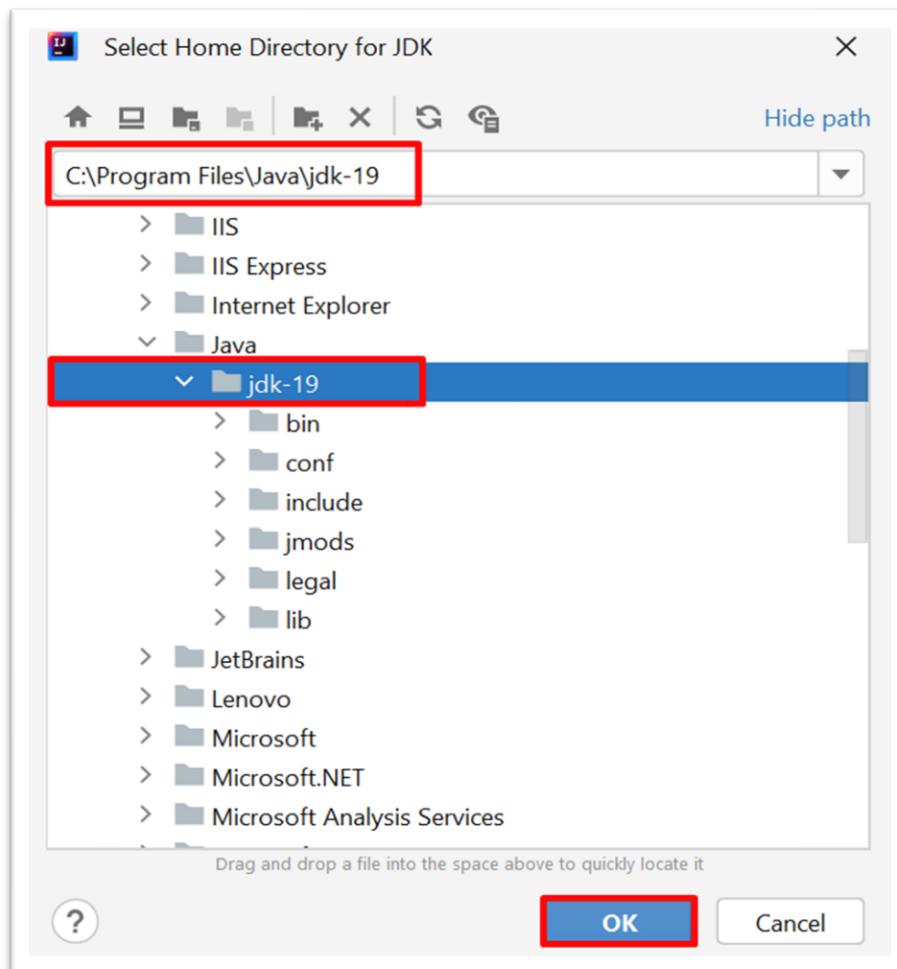
Hình 21. Lựa chọn Project Structure

Tiếp theo, làm theo hướng dẫn bên dưới để Add đến đường dẫn chứa file JDK.



Hình 22. Lựa chọn phiên bản JDK

Cửa sổ xuất hiện, chọn thư mục **jdk-19** theo đường dẫn **C:\Program Files\Java\jdk-19**.



Hình 23. Lựa chọn phiên bản JDK - 19

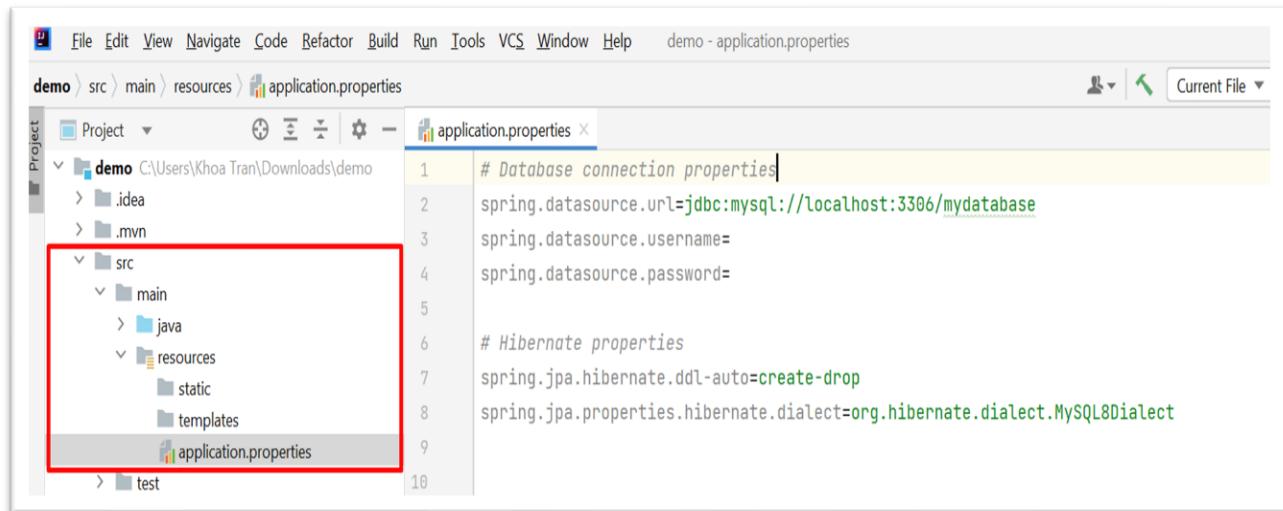
Thêm nội dung vào file **application.properties**

**Application.properties** có thể được đặt trong thư mục **src/main/resources** của ứng dụng web Java. Nó được sử dụng để cấu hình các thông số như cơ sở dữ liệu, cổng kết nối, tên đăng nhập và mật khẩu.

```
# Database connection properties
spring.datasource.url=jdbc:mysql://localhost:3306/mydatabase
spring.datasource.username=
spring.datasource.password=

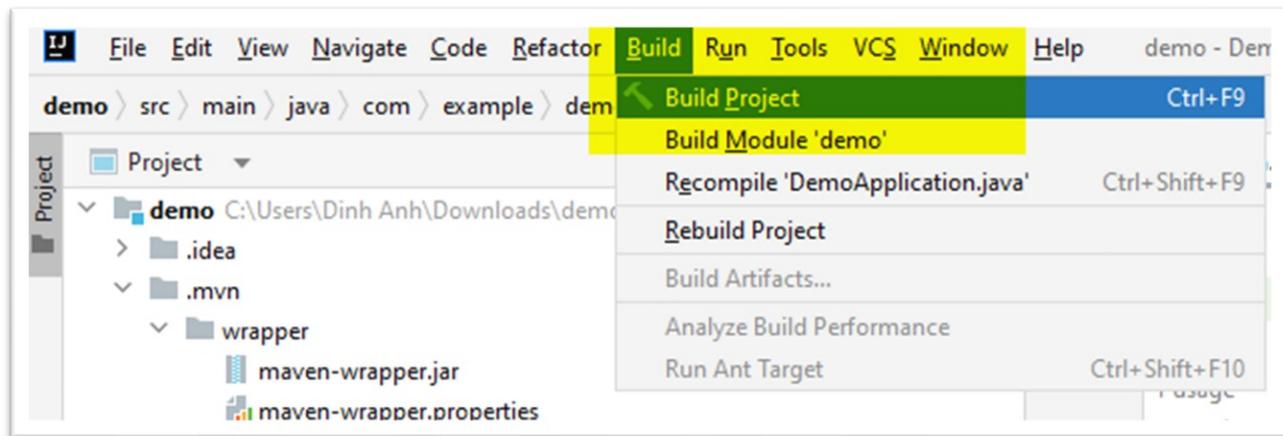
# Hibernate properties
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
```

Tìm đến file **application.properties** tại đường dẫn **src/main/resources**



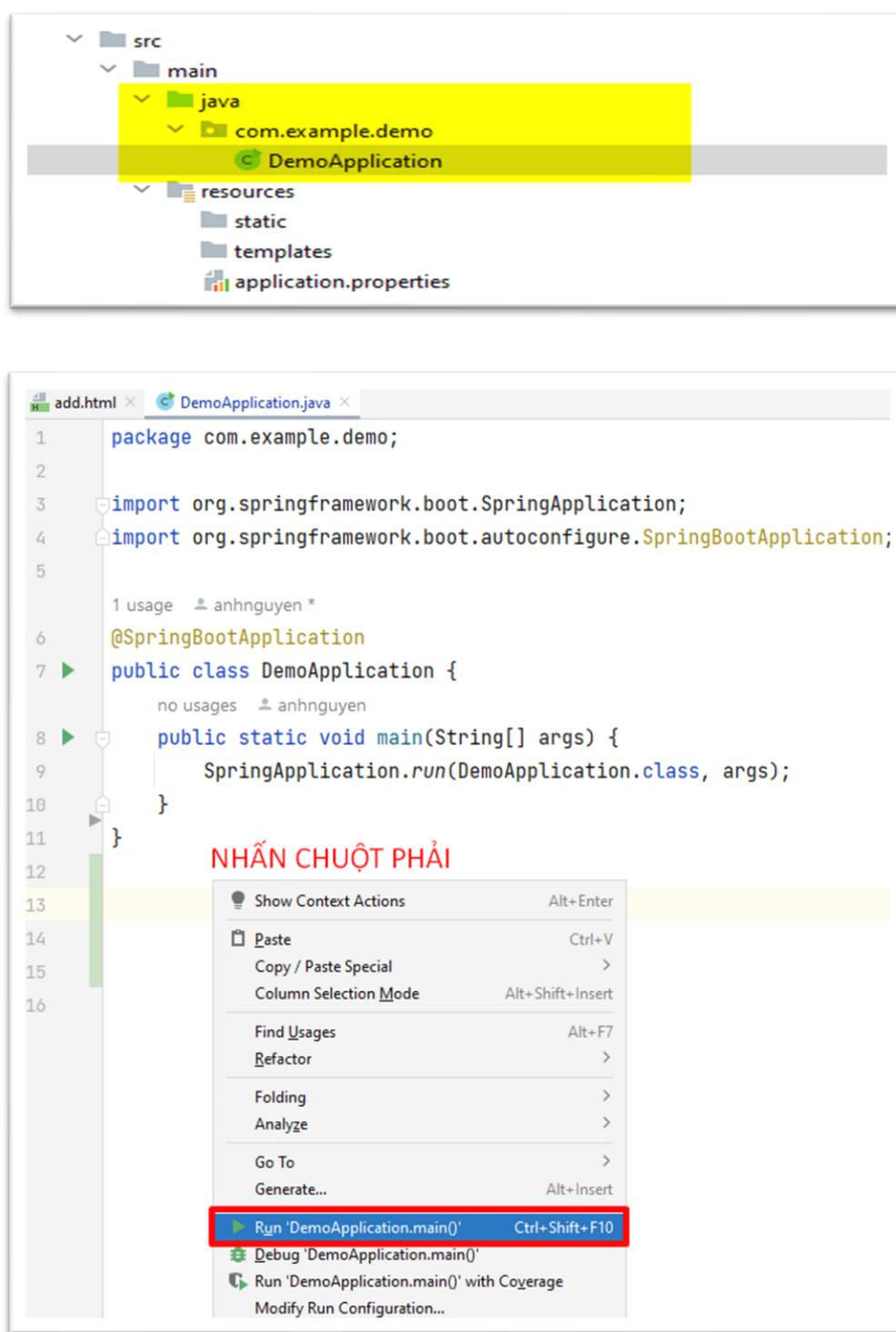
Hình 24. Bổ sung đoạn code cấu hình cho application.properties

Tiếp theo, **Build Project** như ảnh bên dưới.



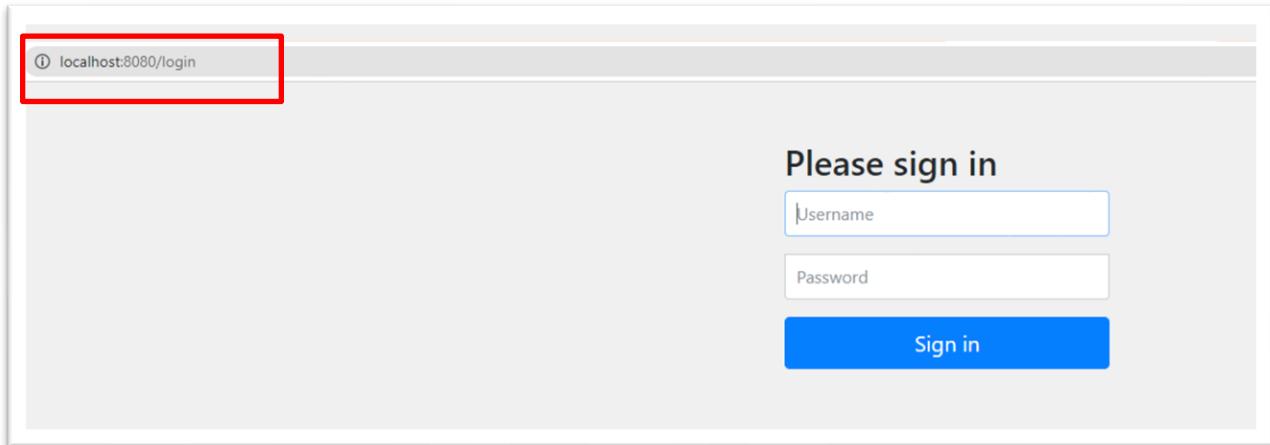
Hình 25. Build project chương trình

Tiến hành Run chương trình tại file **DemoApplication.java** theo ảnh hướng dẫn bên dưới.



Hình 26. Run chương trình tại file `DemoApplication`

Sau đó, thực hiện mở trình duyệt trên máy tính, truy cập vào đường dẫn mặc định: <http://localhost:8080/>, kết quả xuất hiện giao diện đăng nhập.



Hình 27. Giao diện mặc định đăng nhập

Trang mặc định, được cấu hình khởi chạy từ **Spring Security**

Tắt **Spring Security** để thực hiện các demo phía dưới

Chỉnh sửa file **src/application.properties**, thêm phần **#Spring Security**

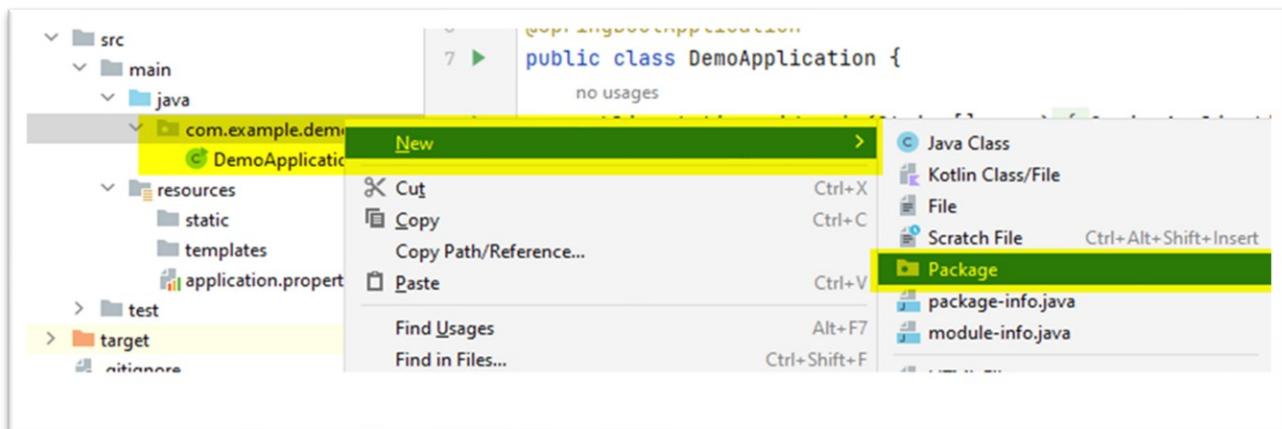
```
# Database connection properties
spring.datasource.url=jdbc:mysql://localhost:3306/mydatabase
spring.datasource.username=
spring.datasource.password=

# Hibernate properties
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

#Spring Security
spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration
```

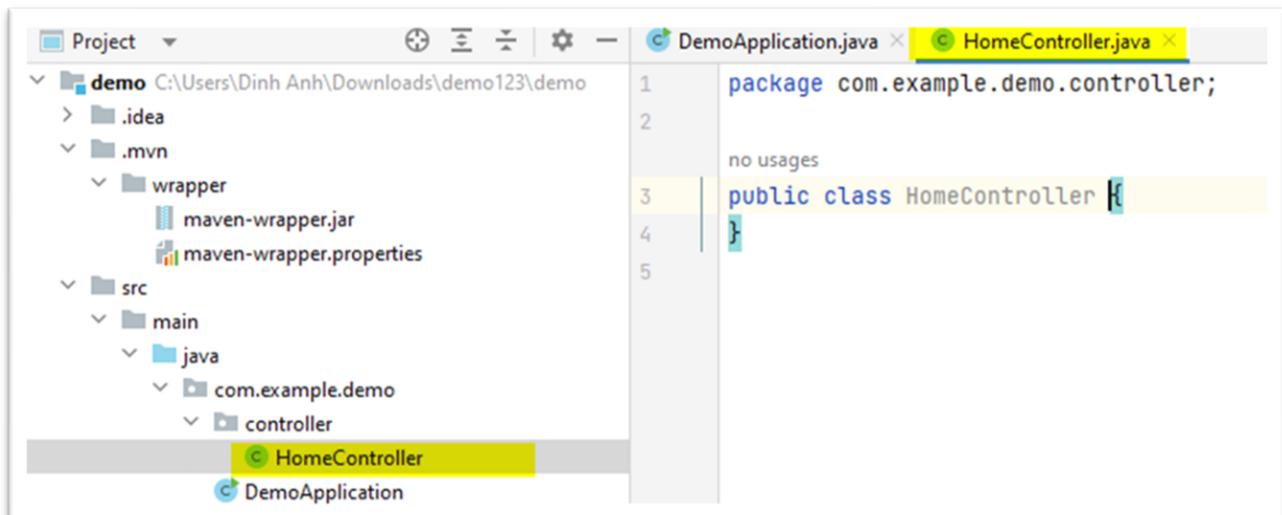
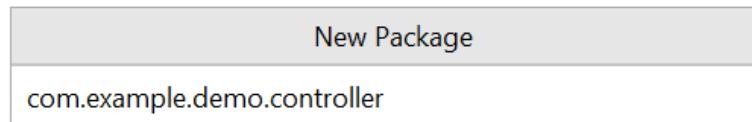
### 1.1.6 Chạy trang web đầu tiên

Tạo package controller tại **src/main/java/com.example.demo**



Hình 28. Tạo package controller

Tạo file **HomeController.java** đặt tại  
**src/main/java/com.example.demo/controller**



Hình 29.Tạo file HomeController.java trong package controller

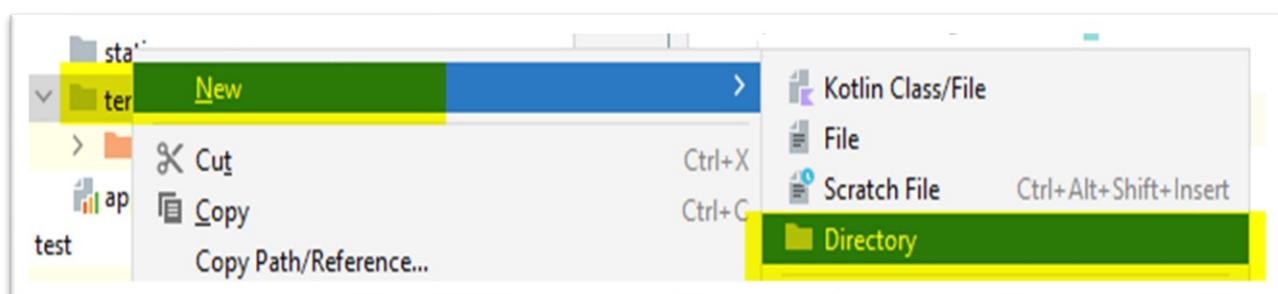
## XXVIII HƯỚNG DẪN

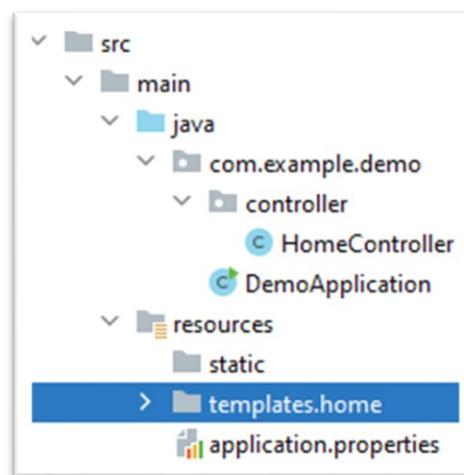
Chỉnh sửa nội dung **HomeController.java** như ảnh bên dưới,

```
1 package com.example.demo.controller;
2
3 import ...
4
5
6 no usages anhnguyen
7 @Controller
8 @RequestMapping("/")
9 public class HomeController {
10
11     @GetMapping("/")
12     public String home() { return "home/index"; }
13
14     @GetMapping("/contact")
15     public String contact() { return "home/contact"; }
16
17 }
```

Hình 30.Chỉnh sửa nội dung HomeController.java

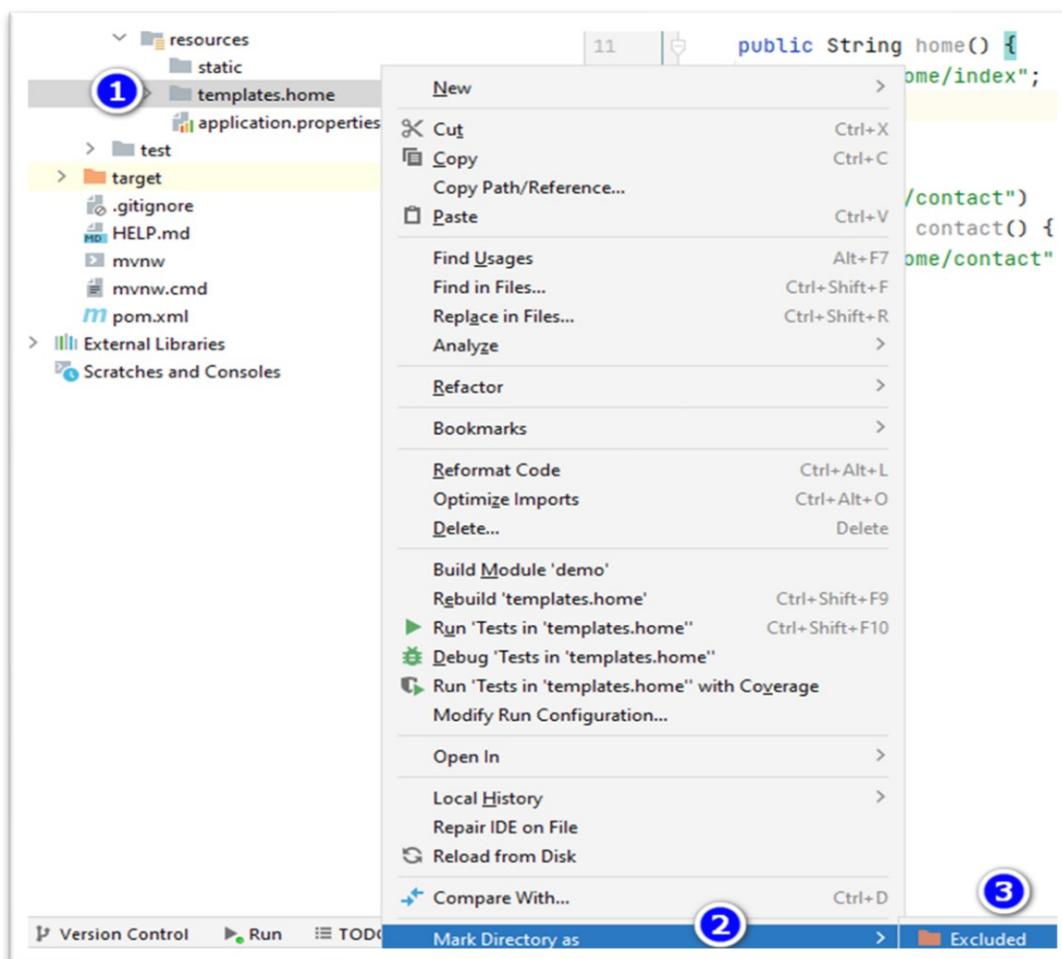
Tạo thư mục **home** đặt tại **src/resources/templates**





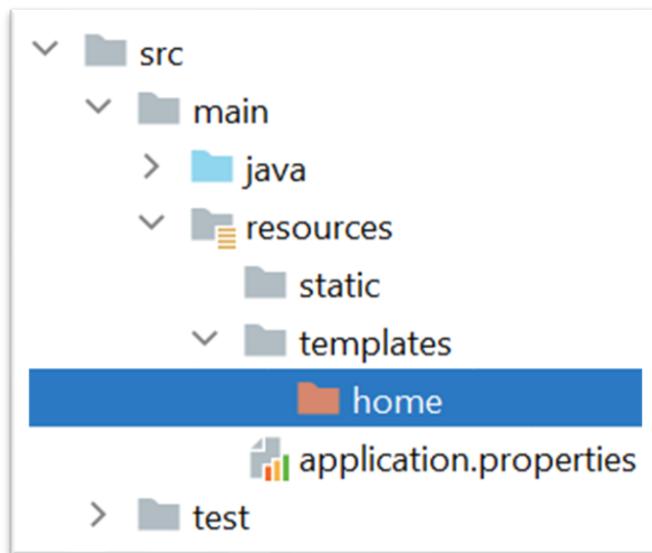
Hình 31. Tạo thư mục home

Để thư mục **home** hiển thị ngay phía dưới thư mục **templates** như ảnh bên dưới.

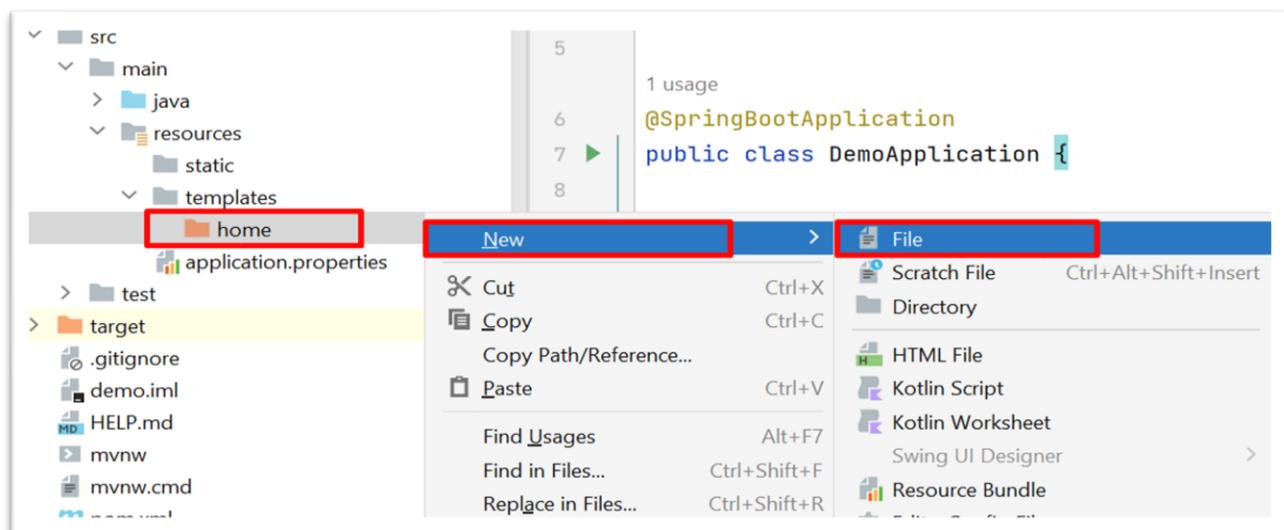


Hình 32. Hiển thị ngay phía dưới thư mục templates

Kết quả như ảnh bên dưới



Tạo file **index.html** đặt tại **src/resources/templates/home**



Hình 33. Tạo file index.html

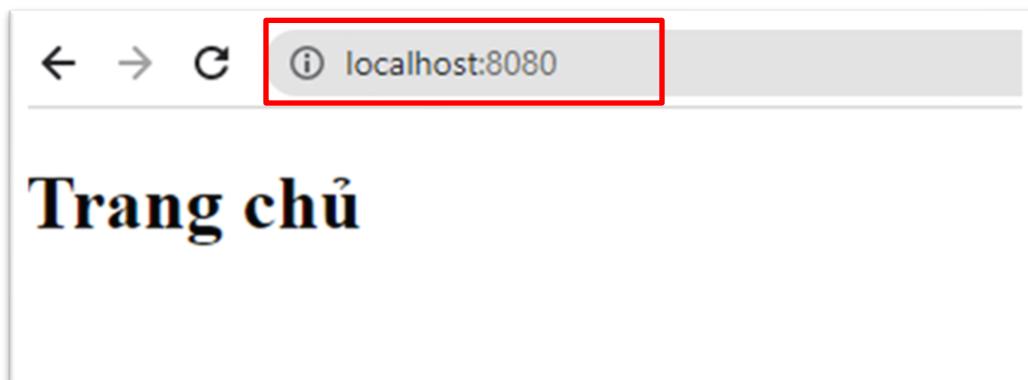
Bổ sung đoạn code như bên dưới vào file **index.html**

The screenshot shows a Maven project structure in the left panel. The `src/main/java/com/example/demo/controller` package contains `HomeController` and `DemoApplication`. The `src/main/resources/templates/home/index.html` file is selected and shown in the right panel. The code in `index.html` is:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
    <h1>Trang chủ</h1>
</body>
</html>
```

Hình 34. Chỉnh sửa file `index.html`

Chạy ứng dụng web lên với địa chỉ **localhost:8080** và xem kết quả,



Hình 35. Giao diện trang chủ

## TÓM TẮT

*Spring MVC là một framework được sử dụng để phát triển các ứng dụng web Java. Nó cung cấp các tính năng để xử lý các yêu cầu HTTP, định tuyến các URL đến các đối tượng xử lý và quản lý các biểu mẫu HTML và các tham số đầu vào khác được gửi từ trình duyệt web. Spring MVC được xây dựng trên Spring Framework, một framework phổ biến cho phát triển ứng dụng Java.*

*Với Spring MVC, các lập trình viên có thể phát triển các ứng dụng web Java một cách nhanh chóng và hiệu quả, bằng cách tận dụng các tính năng mạnh mẽ của framework để giảm thiểu công việc lập trình lặp đi lặp lại. Spring MVC cũng hỗ trợ tích hợp với các công nghệ khác như Hibernate, JPA và Thymeleaf để đơn giản hóa việc xây dựng ứng dụng web.*

*Với Spring MVC, các lập trình viên có thể xây dựng các ứng dụng web đa dạng, bao gồm các trang web tĩnh, trang web động, ứng dụng web phức tạp với nhiều chức năng, ứng dụng web thương mại điện tử và nhiều hơn nữa. Tóm lại, Spring MVC là một trong những lựa chọn phổ biến cho phát triển ứng dụng web Java.*

## CÂU HỎI ÔN TẬP

**Câu 1:** Spring MVC là gì? Tại sao nó lại quan trọng trong phát triển web Java?

**Câu 2:** Các tệp cấu hình nào được sử dụng để cấu hình ứng dụng Spring MVC?

# BÀI 2 VIẾT CHƯƠNG TRÌNH QUẢN LÝ SÁCH

**Bài này giúp người học nắm được các nội dung sau:**

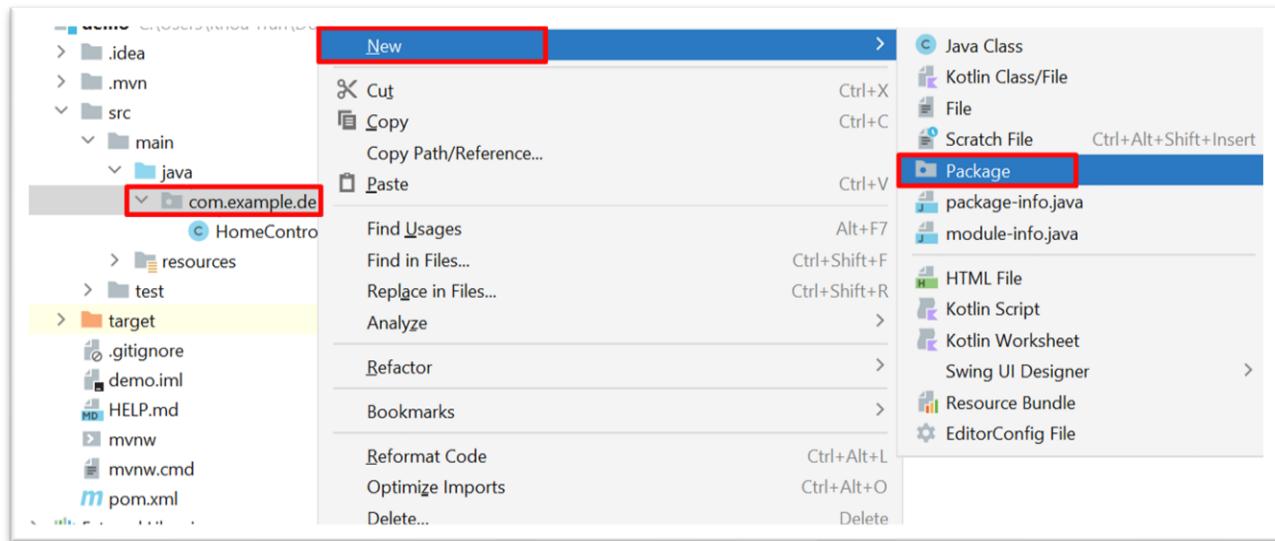
- Lập trình cơ bản ứng dụng Website với JAVA SPRING FRAMEWORK
- Hiểu được cơ chế làm việc của Model, View, Controller
- Xây dựng ứng dụng website quản lý sách với 4 chức năng cơ bản: thêm, xóa, sửa, hiển thị từ một danh sách.

**Mô tả chức năng:**

- ✓ **Thêm sách:** Cho phép người dùng nhập thông tin của một quyển sách gồm tiêu đề, tác giả, giá, danh mục của sách. Thông tin này sẽ được lưu vào một biến toàn cục trong ứng dụng
- ✓ **Xóa sách:** Cho phép người dùng chọn một quyển sách trong danh sách hiển thị và xóa khỏi danh sách.
- ✓ **Sửa sách:** Cho phép người dùng chọn một quyển sách trong danh sách hiển thị và cập nhật thông tin của nó.
- ✓ **Hiển thị sách:** Hiển thị danh sách tất cả các quyển sách có trong danh sách, bao gồm tiêu đề, tác giả, giá và danh mục của sách.

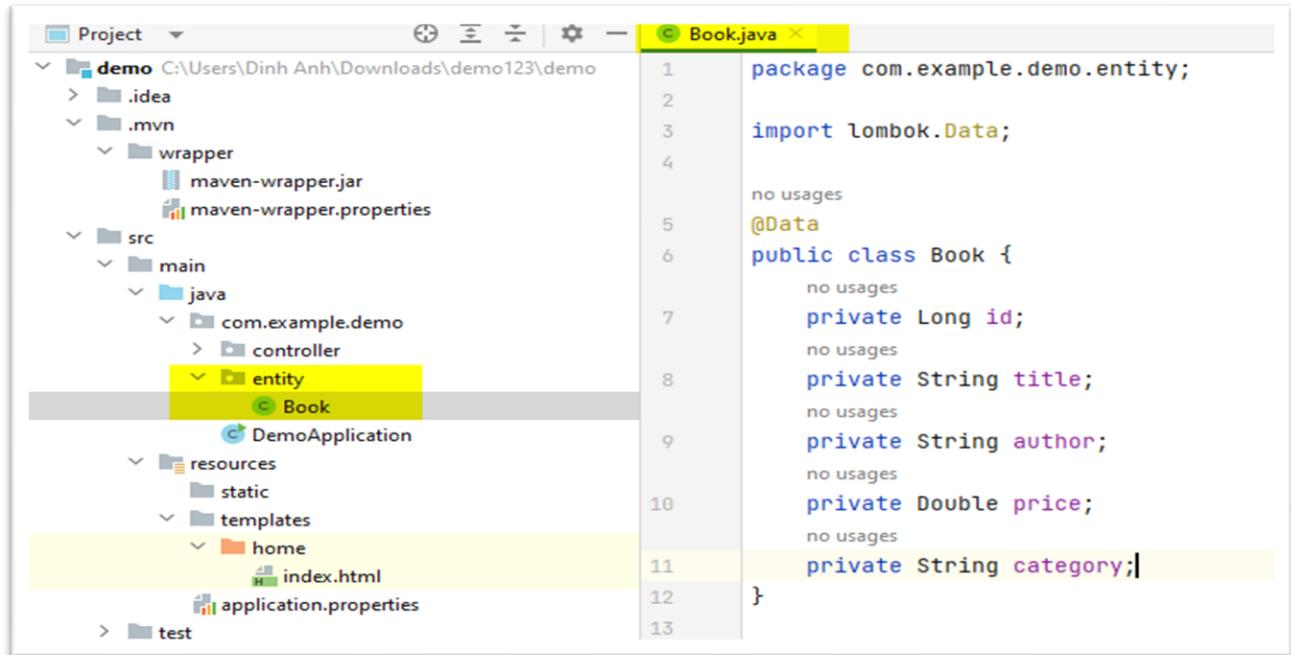
## 2.1 Xây dựng trang Book List

Tạo thêm 1 Package với tên **entity** tại **com.example.demo**



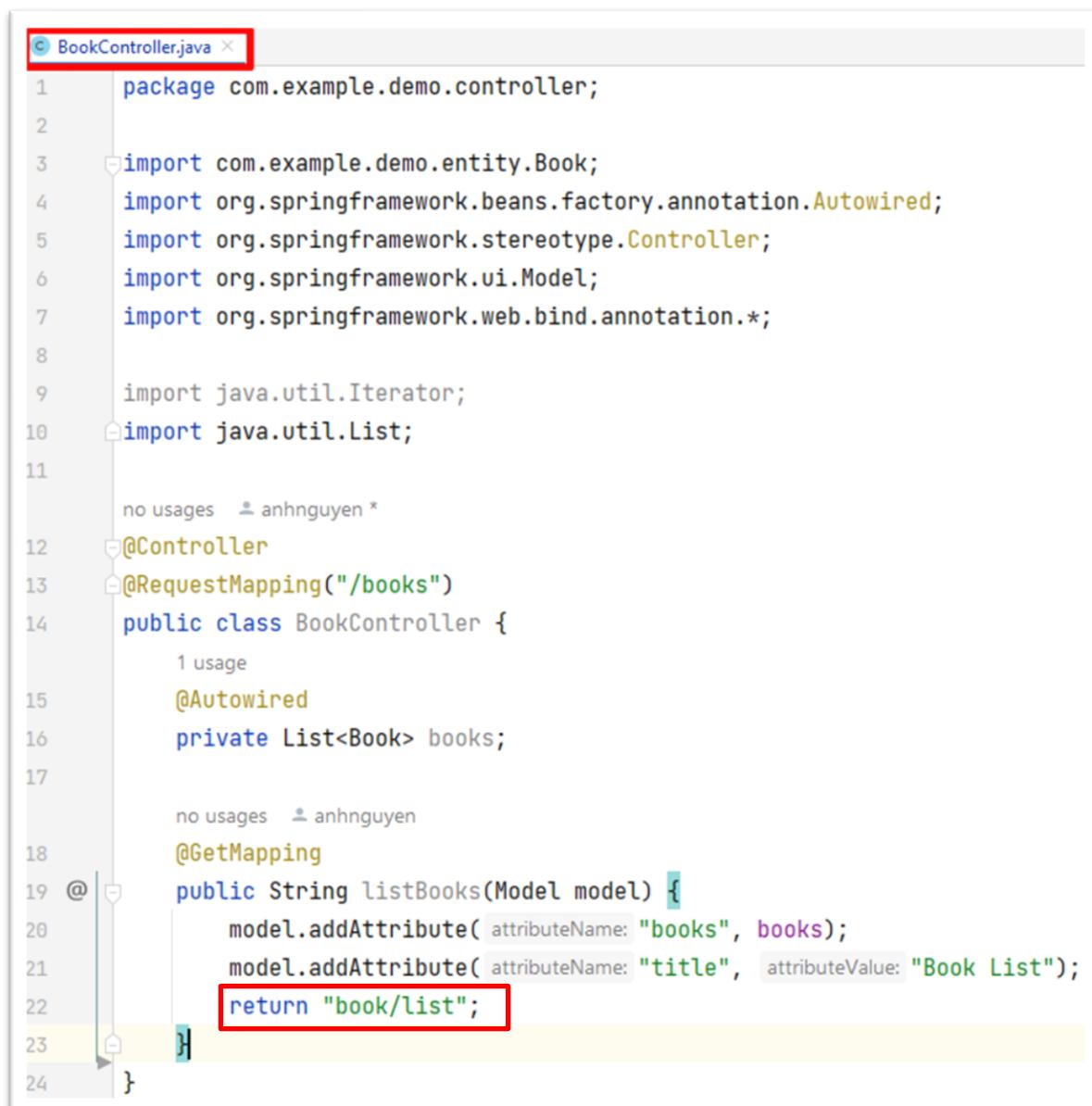
Hình 36. Thêm 1 Package với tên entity

Thêm 1 file **Book.java** trong package vừa tạo, và tạo danh sách các biến cho phép truy cập trong toàn ứng dụng.



Hình 37. Thêm 1 file Book.java và thêm các thuộc tính cho book

Tạo thêm 1 file **BookController.java** trong package **controller** theo đường dẫn sau: **src\main\java\com.example.demo\controller** , và bổ sung những dòng code bên dưới vào như ảnh bên dưới.



```
BookController.java
1 package com.example.demo.controller;
2
3 import com.example.demo.entity.Book;
4 import org.springframework.beans.factory.annotation.Autowired;
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.*;
8
9 import java.util.Iterator;
10 import java.util.List;
11
12 no usages  ▲ anhnguyen *
13 @Controller
14 @RequestMapping("/books")
15 public class BookController {
16     1 usage
17     @Autowired
18     private List<Book> books;
19
20     no usages  ▲ anhnguyen
21     @GetMapping
22     public String listBooks(Model model) {
23         model.addAttribute(attributeName: "books", books);
24         model.addAttribute(attributeName: "title", attributeValue: "Book List");
25         return "book/list";
26     }
27 }
```

Hình 38. Thêm 1 file BookController.java trong package controller

Thực hiện tạo 1 file  **AppConfig.java** tại đường dẫn

**src/main/java/com.example.demo**, Bổ sung đoạn code như ảnh bên dưới vào.

The screenshot shows the IntelliJ IDEA interface with the project structure on the left and the code editor on the right. The project structure includes a 'demo' folder containing '.idea', '.mvn', 'src' (with 'main' and 'test' subfolders), 'target', and 'External Libraries'. The 'src/main/java/com.example.demo/controller' package contains 'BookController' and 'HomeController' classes. The 'src/main/java/com.example.demo/entity' package contains 'AppConfig' and 'DemoApplication' classes. The 'src/main/resources' folder contains 'application.properties'. The code editor displays the 'AppConfig.java' file with the following content:

```

@Configuration
public class AppConfig {
    @Bean
    public List<Book> getBooks() {
        List<Book> books = new ArrayList<>();
        Book book1 = new Book();
        book1.setId(1L);
        book1.setTitle("Lập trình Web Spring MVC");
        book1.setAuthor("Ánh Nguyễn");
        book1.setPrice(10.99);
        book1.setCategory("Công nghệ thông tin");
        books.add(book1);

        Book book2 = new Book();
        book2.setId(2L);
        book2.setTitle("Lập trình Ứng dụng Spring");
        book2.setAuthor("Huy Cường");
        book2.setPrice(12.99);
        book2.setCategory("Công nghệ thông tin");
        books.add(book2);

        Book book3 = new Book();
        book3.setId(3L);
        book3.setTitle("Lập trình Ứng dụng Java");
        book3.setAuthor("Xuân Nhân");
        book3.setPrice(15.99);
        book3.setCategory("Công nghệ thông tin");
        books.add(book3);

        Book book4 = new Book();
        book4.setId(4L);
        book4.setTitle("Lập trình Ứng dụng Java");
        book4.setAuthor("Đặng Khoa");
        book4.setPrice(15.99);
        book4.setCategory("Công nghệ thông tin");
        books.add(book4);

        return books;
    }
}

```

Hình 39. Tạo file AppConfig.java

## 2.1.1 Tạo view layout trong ứng dụng

Tải thư viện Bootstrap từ <https://getbootstrap.com/>

### Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.3.0-alpha1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

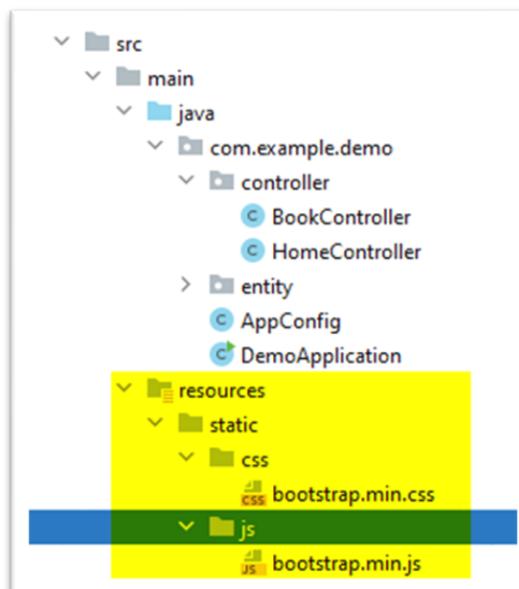
[Download](#)

Hình 40. Tải thư viện Bootstrap

Tìm đến file vừa Download về từ trang Bootstrap, giải nén ra → chứa 2 thư mục lần lượt là **css** và **js**.

Tạo thư mục **CSS** tại **src/resources/static/css** và bỏ file bootstrap .css

Tạo thư mục **JS** tại **src/resources/static/css** và bỏ file bootstrap .js



Hình 41. Thêm thư viện Bootstrap vào dự án

Tạo 1 file **layout.html** đặt tại đường dẫn **src/resources/templates**



```
layout.html
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3  <head>
4      <meta charset="UTF-8">
5      <title>My App</title>
6      <link th:fragment="link-css" rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
7  </head>
8  <body>
9      <header th:fragment="header">
10         <nav class="navbar navbar-expand-lg navbar-light bg-light">
11             <div class="container-fluid">
12                 <a class="navbar-brand" href="#">HUTECH</a>
13                 <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
14                     data-bs-target="#navbarSupportedContent"
15                     aria-controls="navbarSupportedContent"
16                     aria-expanded="false" aria-label="Toggle navigation">
17                     <span class="navbar-toggler-icon"></span>
18                 </button>
19                 <div class="collapse navbar-collapse" id="navbarSupportedContent">
20                     <ul class="navbar-nav me-auto mb-2 mb-lg-0">
21                         <li class="nav-item">
22                             <a class="nav-link active" aria-current="page" href="/">Home</a>
23                         </li>
24                         <li class="nav-item"><a class="nav-link" href="#">List Book</a></li>
25                         <li class="nav-item"><a class="nav-link" href="#">Add Book</a></li>
26                     </ul>
27                 </div>
28             </div>
29         </nav>
30     </header>
31     <div th:insert="~{::${content}}">
32         <!-- NỘI DUNG TRANG CON-->
33     </div>
34     <footer th:fragment="footer">
35         <script th:src="@{/js/bootstrap.min.js}"></script>
36     </footer>
37     </body>
38 </html>
```

Hình 42. Tạo file layout.html

Tạo view hiển thị danh sách, tại thư mục **templates** tạo thêm 1 thư mục tên là **book**. Tạo thêm **file list.html** trong thư mục **book**.



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>My Book List</title>
    <th:block th:replace="layout :: link-css"></th:block>
</head>
<body>
    <th:block th:replace="layout :: header"></th:block>
    <div class="container">
        <h1 th:text=${title}></h1>
        <table class="table">
            <tr>
                <th>ID</th>
                <th>Title</th>
                <th>Author</th>
                <th>Price</th>
                <th>Category</th>
                <th>Action</th>
            </tr>
            <tr th:each="book : ${books}">
                <td th:text="${book.id}"></td>
                <td th:text="${book.title}"></td>
                <td th:text="${book.author}"></td>
                <td th:text="${book.price}"></td>
                <td th:text="${book.category}"></td>
                <td>
                    <a th:href="@{/books/edit/{id}(id=${book.id})}">Edit</a> |
                    <a th:href="@{/books/delete/{id}(id=${book.id})}"
                       onclick="return confirm('Are you sure?')">Delete</a>
                </td>
            </tr>
        </table>
    </div>
    <th:block th:replace="layout :: footer"></th:block>
</body>
</html>
```

Hình 43. Tạo view hiển thị danh sách

Build và chạy ứng dụng xem kết quả, truy cập địa chỉ như bên dưới.

ID	Title	Author	Price	Category	Action
1	Lập trình Web Spring MVC	Ánh Nguyễn	10.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
2	Lập trình Ứng dụng Spring	Huy Cường	12.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
3	Lập trình Ứng dụng Java	Xuân Nhân	15.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
4	Lập trình Ứng dụng Java	Đăng Khoa	15.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>

Hình 44. Kết quả, trang Book List

## 2.2 Xây dựng trang thêm sách

### 2.2.1 Chính sửa BookController.java

Tìm đến thư mục **controller** và chỉnh sửa file **BookController.java** như ảnh bên dưới.

```

no usages  ▲ anhnguyen
24   @GetMapping("/add")
25     public String addBookForm(Model model) {
26       model.addAttribute(attributeName: "book", new Book());
27       return "book/add";
28     }
no usages  ▲ anhnguyen
29   @PostMapping("/add")
30     public String addBook(@ModelAttribute("book") Book book) {
31       books.add(book);
32       return "redirect:/books";
33     }
34
35 }
```

Hình 45. Chính sửa BookController.java cho chức năng Add

**@GetMapping("/add")** là một **annotation** trong Spring Framework được sử dụng để ánh xạ một phương thức xử lý request HTTP GET đến một địa chỉ URL, cụ thể là **/add**.

Trong đó, khi request HTTP GET được gửi đến địa chỉ URL **/add**, phương thức **addBookForm** được gọi để xử lý request và trả về tên view "**book/add**".

Tương tự, **@PostMapping("/add")** là một annotation trong Spring Framework được sử dụng để ánh xạ một phương thức xử lý request HTTP POST đến một địa chỉ URL, cụ thể là **/add**.

## 2.2.2 Thêm View hiển thị

Tạo file **add.html** đặt tại **src/resources/templates/book**



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <th:block th:replace="layout :: link-css"></th:block>
    <title>Add Book</title>
</head>
<body>
    <th:block th:replace="layout :: header"></th:block>
    <div class="container">
        <h1>Add Book</h1>
        <form th:action="@{/books/add}" th:object="${book}" method="post">
            <div class="col-6 mb-3">
                <label class="form-label" for="title">Id:</label>
                <input class="form-control" type="text" th:field="*{id}" id="id">
            </div>
            <div class="col-6 mb-3">
                <label class="form-label" for="title">Title:</label>
                <input class="form-control" type="text" th:field="*{title}" id="title">
            </div>
            <div class="col-6 mb-3">
                <label class="form-label" for="author">Author:</label>
                <input class="form-control" type="text" th:field="*{author}" id="author">
            </div>
            <div class="col-6 mb-3">
                <label class="form-label" for="price">Price:</label>
                <input class="form-control" type="text" th:field="*{price}" id="price">
            </div>
            <div class="col-6 mb-3">
                <label class="form-label" for="category">Category:</label>
                <input class="form-control" type="text" th:field="*{category}" id="category">
            </div>
            <input type="submit" class="btn btn-primary" value="Save">
        </form>
        <br>
        <a th:href="@{/books}">Back to List</a>
    </div>
    <th:block th:replace="layout :: footer"></th:block>
</body>
</html>
```

Hình 46. Tạo giao diện View hiển thị, tạo file add.html

Chỉnh sửa đường dẫn các các menu ở trang **layout.html** , tại đường dẫn **src/main/resources/templates**

Chỉnh sửa thành **href="/books"**

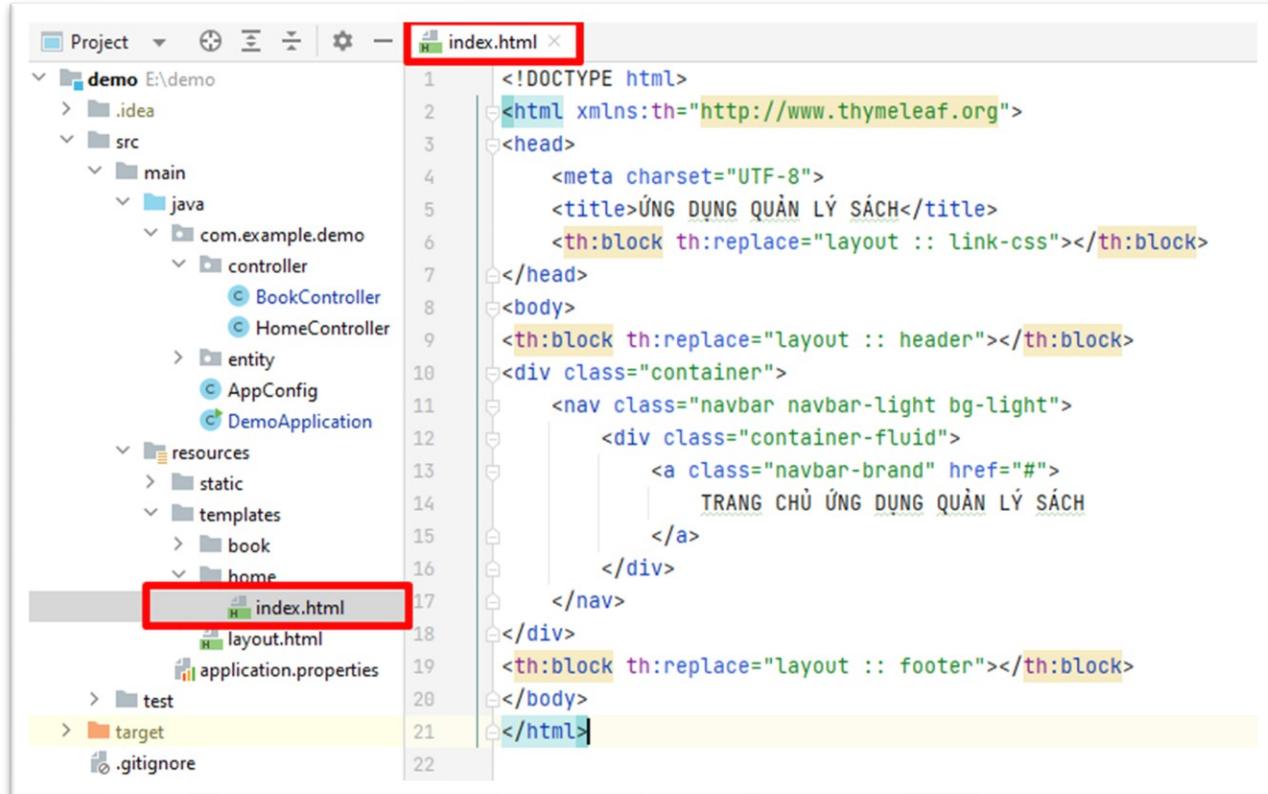
Chỉnh sửa thành **href="/books/add"**



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>My App</title>
    <link th:fragment="link-css" rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
</head>
<body>
    <header th:fragment="header">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <div class="container-fluid">
                <a class="navbar-brand" href="#">HUTECH</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
                    data-bs-target="#navbarSupportedContent"
                    aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                        <li class="nav-item">
                            <a class="nav-link active" aria-current="page" href="/">Home</a>
                        </li>
                        <li class="nav-item"><a class="nav-link" href="/books">List Book</a></li>
                        <li class="nav-item"><a class="nav-link" href="/books/add">Add Book</a></li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div th:insert="~{::${content}}">
        <!-- NỘI DUNG TRANG CON-->
    </div>
    <footer th:fragment="footer">
        <script th:src="@{/js/bootstrap.min.js}"></script>
    </footer>
</body>
</html>
```

Hình 47. Chỉnh sửa đường dẫn các các menu ở trang layout.html

Chỉnh sửa nội dung trang chủ **index.html** tại đường dẫn  
**src/resources/templates/home/index.html**



The screenshot shows a Java-based application structure in an IDE. The project tree on the left includes a demo module with .idea, src, test, target, and .gitignore files. The src directory contains main/java/com/example/demo/controller (BookController, HomeController), entity (AppConfig, DemoApplication), and resources/templates/book/home (index.html, layout.html). The index.html file is selected and shown in the code editor on the right. The code uses Thymeleaf syntax to define a header, body, and footer for the page.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>ỨNG DỤNG QUẢN LÝ SÁCH</title>
    <th:block th:replace="layout :: link-css"></th:block>
</head>
<body>
<th:block th:replace="layout :: header"></th:block>
<div class="container">
    <nav class="navbar navbar-light bg-light">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">
                TRANG CHỦ ỨNG DỤNG QUẢN LÝ SÁCH
            </a>
        </div>
    </nav>
</div>
<th:block th:replace="layout :: footer"></th:block>
</body>
</html>
```

Hình 48. Chỉnh sửa nội dung trang chủ tại index.html

Build chương trình tại địa chỉ **localhost:8080/books/add** và chạy ứng dụng xem kết quả.

The screenshot shows a web browser window with the URL `localhost:8080/books/add` highlighted in red in the address bar. The page title is "Add Book". The form fields are as follows:

- Id:** An empty input field.
- Title:** An empty input field.
- Author:** An empty input field.
- Price:** An empty input field.
- Category:** An empty input field.

At the bottom of the form are two buttons: a blue **Save** button and a link **Back to List**.

Hình 49. Giao diện trang thêm book

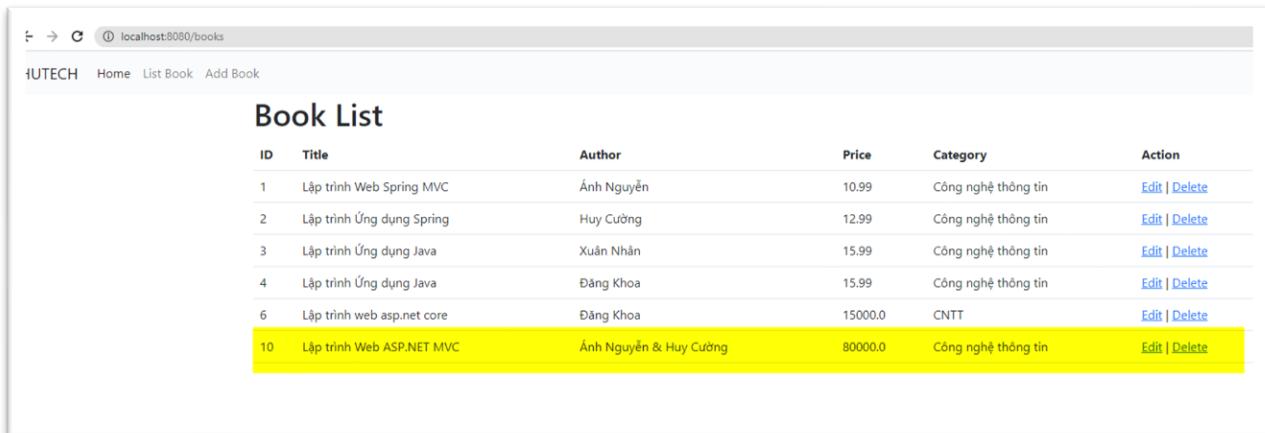
Nhập thông tin đầy đủ để thêm một cuốn sách, nhấn nút Save và xem kết quả.

The screenshot shows the same "Add Book" form as above, but with the following data entered into the fields:

- Id:** 10
- Title:** Lập trình Web ASP.NET MVC
- Author:** Ánh Nguyễn & Huy Cường
- Price:** 80000
- Category:** Công nghệ thông tin

At the bottom of the form are two buttons: a blue **Save** button and a link **Back to List**.

Kết quả nhận được là một cuốn sách vừa thêm vào cuối danh sách. Như vậy bạn đã hoàn thành chức năng thêm sách mới.

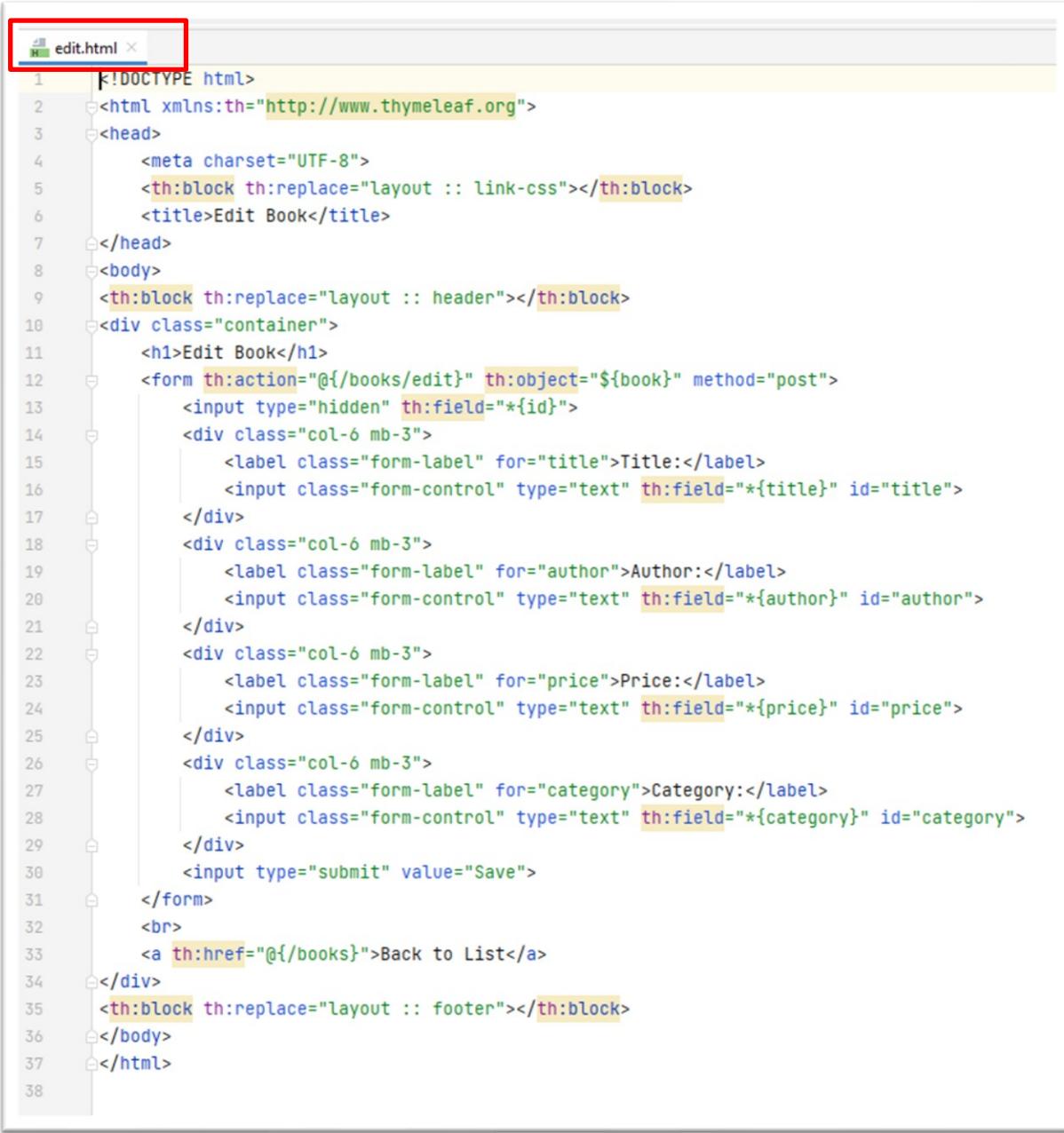


ID	Title	Author	Price	Category	Action
1	Lập trình Web Spring MVC	Ánh Nguyễn	10.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
2	Lập trình Ứng dụng Spring	Huy Cường	12.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
3	Lập trình Ứng dụng Java	Xuân Nhân	15.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
4	Lập trình Ứng dụng Java	Đặng Khoa	15.99	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>
5	Lập trình web asp.net core	Đặng Khoa	15000.0	CNTT	<a href="#">Edit</a>   <a href="#">Delete</a>
10	Lập trình Web ASP.NET MVC	Ánh Nguyễn & Huy Cường	80000.0	Công nghệ thông tin	<a href="#">Edit</a>   <a href="#">Delete</a>

Hình 50. Trang danh sách Book sau khi thêm Book mới

## 2.3 Xây dựng trang sửa sách

Thêm 1 file **edit.html** đặt tại **src/resources/templates/book**



```
edit.html
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8">
5     <th:block th:replace="layout :: link-css"></th:block>
6     <title>Edit Book</title>
7 </head>
8 <body>
9     <th:block th:replace="layout :: header"></th:block>
10 <div class="container">
11     <h1>Edit Book</h1>
12     <form th:action="@{/books/edit}" th:object="${book}" method="post">
13         <input type="hidden" th:field="*{id}">
14         <div class="col-6 mb-3">
15             <label class="form-label" for="title">Title:</label>
16             <input class="form-control" type="text" th:field="*{title}" id="title">
17         </div>
18         <div class="col-6 mb-3">
19             <label class="form-label" for="author">Author:</label>
20             <input class="form-control" type="text" th:field="*{author}" id="author">
21         </div>
22         <div class="col-6 mb-3">
23             <label class="form-label" for="price">Price:</label>
24             <input class="form-control" type="text" th:field="*{price}" id="price">
25         </div>
26         <div class="col-6 mb-3">
27             <label class="form-label" for="category">Category:</label>
28             <input class="form-control" type="text" th:field="*{category}" id="category">
29         </div>
30         <input type="submit" value="Save">
31     </form>
32     <br>
33     <a th:href="@{/books}">Back to List</a>
34 </div>
35     <th:block th:replace="layout :: footer"></th:block>
36 </body>
37 </html>
```

Hình 51. Thêm file edit.html và bổ sung code

Chỉnh sửa file **BookController.java**

Thêm 1 hàm: **editbookForm()**, với 1 annotation

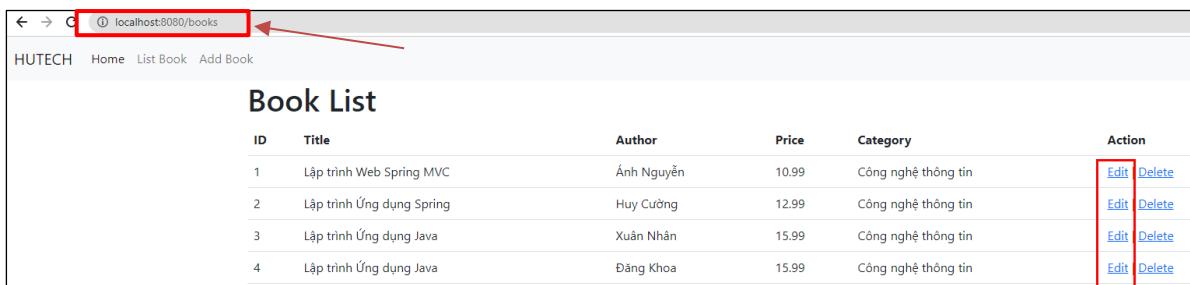
**@PostMapping("/edit/{id}")**

Tiếp, Thêm 1 hàm **editbook()**, với 1 annotation **@PostMapping("/edit")**

```
no usages  ± anhnguyen
35      @GetMapping("/edit/{id}")
36      public String editBookForm(@PathVariable("id") Long id, Model model) {
37          Book editBook = null;
38          for (Book book : books) {
39              if (book.getId().equals(id)) {
40                  editBook = book;
41              }
42          }
43          if (editBook != null) {
44              model.addAttribute( attributeName: "book", editBook);
45              return "book/edit";
46          }else {
47              return "not-found";
48          }
49      }
50
51      no usages  ± anhnguyen
52      @PostMapping("/edit")
53      public String editBook(@ModelAttribute("book") Book updatedBook) {
54          for (int i = 0; i < books.size(); i++) {
55              Book book = books.get(i);
56              if (book.getId() == updatedBook.getId()) {
57                  books.set(i, updatedBook);
58              }
59          }
60          return "redirect:/books";
61      }
```

Hình 52. Thêm hàm `editbook()` vào `BookController`

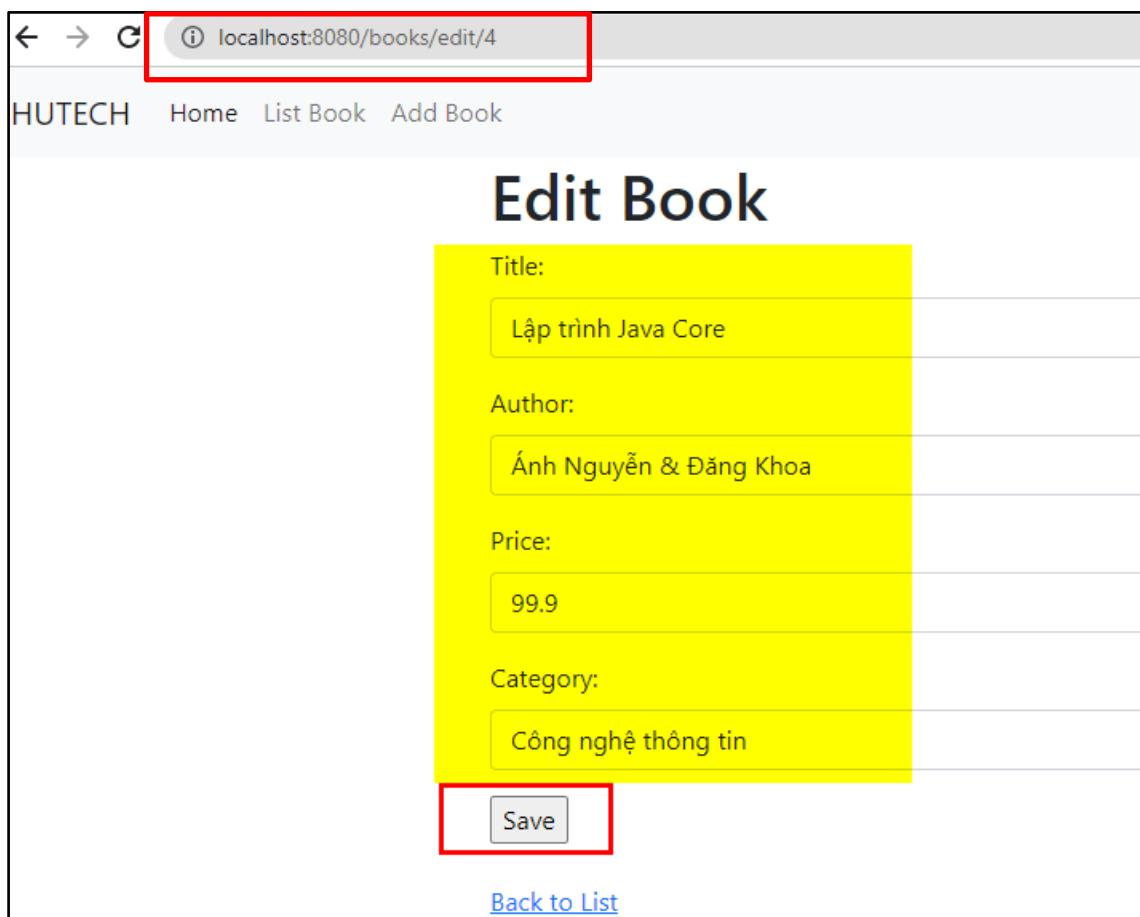
## Chạy ứng dụng với trang Book List



ID	Title	Author	Price	Category	Action
1	Lập trình Web Spring MVC	Ánh Nguyễn	10.99	Công nghệ thông tin	<a href="#">Edit</a> <a href="#">Delete</a>
2	Lập trình Ứng dụng Spring	Huy Cường	12.99	Công nghệ thông tin	<a href="#">Edit</a> <a href="#">Delete</a>
3	Lập trình Ứng dụng Java	Xuân Nhân	15.99	Công nghệ thông tin	<a href="#">Edit</a> <a href="#">Delete</a>
4	Lập trình Ứng dụng Java	Đặng Khoa	15.99	Công nghệ thông tin	<a href="#">Edit</a> <a href="#">Delete</a>

Hình 53. Giao diện Book List với nút Edit

Chọn 1 cuốn sách muốn **Edit**, xem đường dẫn như bên dưới, chỉnh sửa thông tin và kiểm tra kết quả sau đó nhấn nút **save**, như vậy bạn đã hoàn thành chức năng Edit Book.



The screenshot shows the 'Edit Book' form. The URL in the address bar is 'localhost:8080/books/edit/4'. The form fields are:

- Title: Lập trình Java Core
- Author: Ánh Nguyễn & Đặng Khoa
- Price: 99.9
- Category: Công nghệ thông tin

A red box highlights the 'Save' button at the bottom of the form.

Hình 54. Giao diện trang chỉnh sửa Book

Kết quả sau khi hoàn tất chỉnh sửa,

ID	Title	Author	Price	Category	Action
1	Lập trình Web Spring MVC	Ánh Nguyễn	10.99	Công nghệ thông tin	<a href="#">Edit   Delete</a>
2	Lập trình Ứng dụng Spring	Huy Cường	12.99	Công nghệ thông tin	<a href="#">Edit   Delete</a>
3	Lập trình Ứng dụng Java	Xuân Nhàn	15.99	Công nghệ thông tin	<a href="#">Edit   Delete</a>
4	Lập trình Java Core	Ánh Nguyễn & Đăng Khoa	99.9	Công nghệ thông tin	<a href="#">Edit   Delete</a>

Hình 55. Kết quả sau khi chỉnh sửa thành công, trả về Book List

## 2.4 Thực hiện chức năng xóa sách

Thêm 1 hàm tên **deleteBook()** bên trong file **BookController.java**.

Chỉnh sửa file **BookController.java** như ảnh bên dưới với

**@PostMapping("delete/{id}")**, **Id** của cuốn sách cần xoá sẽ được truyền vào hàm **deleteBook()**.

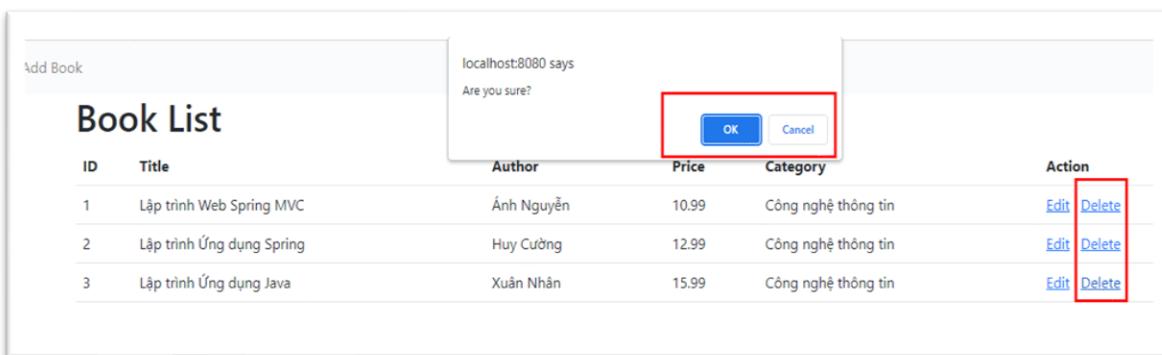
```

no usages  ± anhnguyen
63
64      @GetMapping("/delete/{id}")
65      public String deleteBook(@PathVariable("id") Long id) {
66          Iterator<Book> iterator = books.iterator();
67          while (iterator.hasNext()) {
68              Book book = iterator.next();
69              if (book.getId() == id) {
70                  iterator.remove();
71                  break;
72              }
73          }
74      }
    return "redirect:/books";
}

```

Hình 56. Thêm hàm tên **deleteBook()** bên trong file **BookController.java**

Build chạy ứng dụng và kiểm tra kết quả,  
Tại giao diện Book List, chọn 1 cuốn sách muốn xoá. Một thông báo dạng Alert xuất hiện xác nhận thao tác xoá.



Hình 57.Giao diện delete Book

## TÓM TẮT

*Chức năng thêm, xoá và sửa (CRUD) là những tính năng quan trọng trong ứng dụng Java Web Spring MVC.*

*Chức năng thêm mới đối tượng thường bao gồm việc tạo một form để người dùng nhập thông tin và lưu trữ dữ liệu vào cơ sở dữ liệu. Trong quá trình thêm mới đối tượng, cần xác thực dữ liệu đầu vào và hiển thị thông báo lỗi cho người dùng nếu cần thiết.*

*Chức năng xoá đối tượng yêu cầu xác nhận từ người dùng trước khi xóa dữ liệu khỏi cơ sở dữ liệu. Trong quá trình xóa đối tượng, cần xử lý các tình huống đặc biệt như trường hợp đối tượng không tồn tại.*

*Chức năng sửa đổi đối tượng yêu cầu phải hiển thị thông tin của đối tượng cần sửa đổi cho người dùng và cho phép họ cập nhật các thuộc tính của đối tượng đó. Trong quá trình sửa đổi đối tượng, cần xác thực dữ liệu đầu vào và hiển thị thông báo lỗi cho người dùng nếu cần thiết.*

## CÂU HỎI ÔN TẬP

**Câu 1:** Cách thực hiện chức năng thêm mới đối tượng trong Spring MVC?

**Câu 2:** Làm thế nào để xóa một đối tượng trong Spring MVC?

**Câu 3:** Các bước để thực hiện chức năng sửa đổi đối tượng trong Spring MVC?