FPT Education
FPT UNIVERSITY

Java

Spring Boot Concepts

Objectives

- What is Spring Boot?
- Spring Version History
- Key features of Spring Boot
- Developing Your First Spring Boot Application

Spring Boot

What is Spring Boot ?

- Spring Boot is a powerful framework built on top of the Spring Framework that simplifies the development of Spring applications. It aims to reduce boilerplate code, configuration, and setup time, allowing developers to focus on building the core functionality of their applications.

Spring Version History

Spring Boot Version	Release Date	Spring framework version
1.0.0.RELEASE	Apr 2014	4.0.3
1.5.x	*Jan 2017 - Aug 2019*	*4.3.6 - 4.3.25*
2.0.0.RELEASE	Mar 2018	5.0.4
2.7.x	*May 2022 - Aug 2023*	*5.3.20 - 5.3.29*
3.0.0	Nov 2022	6.0.2
3.0.x		*6.0.2 - 6.0.11*
3.1.x	*May 2023 - Now*	
3.2.0	*Nov 23, 2023*	6.1
3.3.5	**Oct 24, 2024**	**6.1.14**
3.4.0	**Nov 21, 2024**	

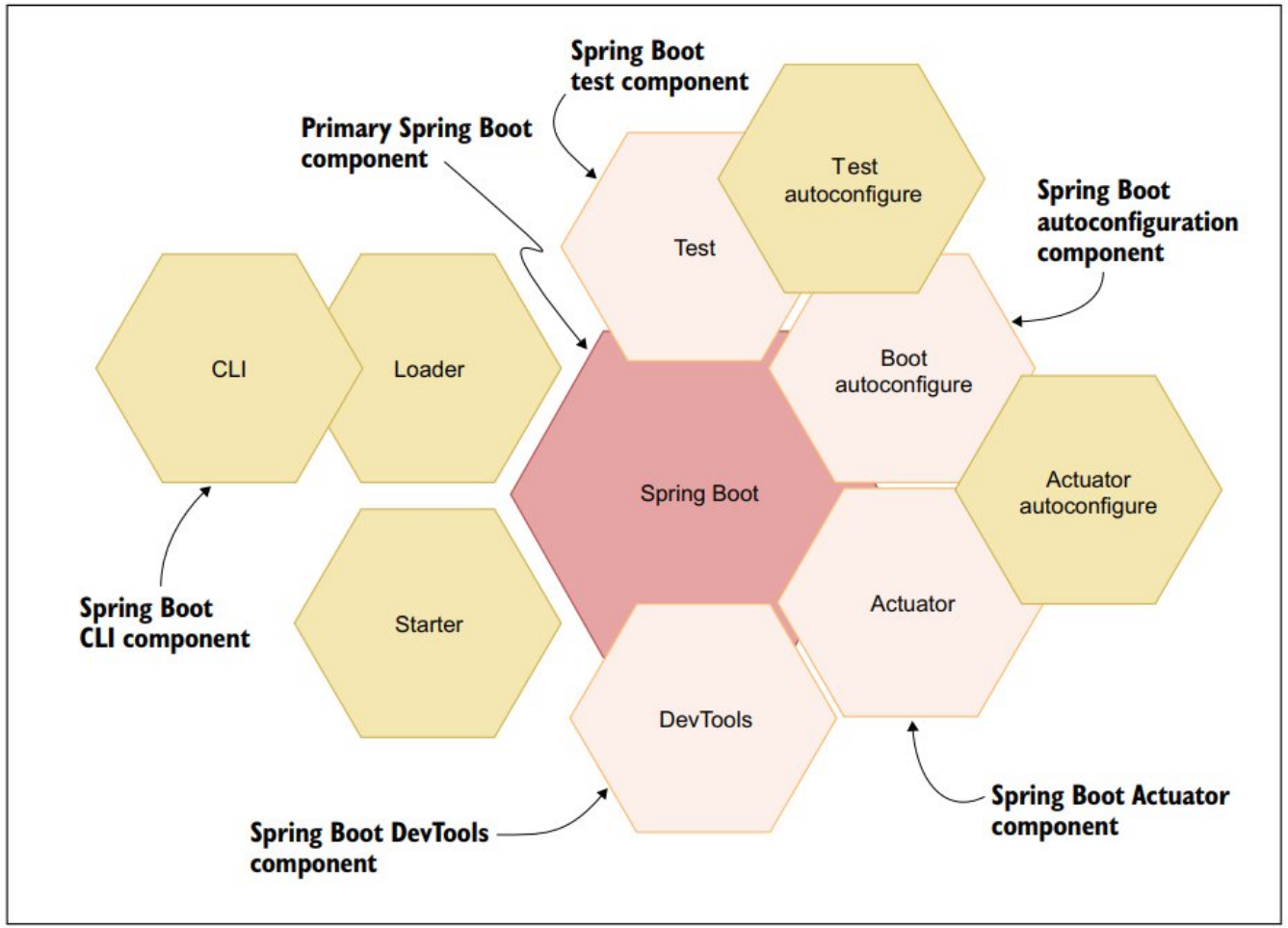
Spring Features

- **Autoconfiguration:** Spring Boot automatically configures your application based on the dependencies you include in your project.
- **Data Access:** Seamless integration with various data sources like relational databases, NoSQL databases, and JPA.
- **Stand-alone Applications:** Spring Boot allows you to create standalone applications that can be run directly without the need for external servers.
- **Embedded Servers:** Spring Boot includes embedded servers like Tomcat or Netty, so you can run your application without deploying it to a separate server.
- **Starter POMs:** Spring Boot provides starter POMs that simplify dependency management. You can easily include the necessary dependencies for specific functionalities, such as web development, database access, or security.

Benefits of Using Spring Boot

- **Rapid Application Development:** Quickly create and deploy applications.
- **Simplified Configuration:** Reduce boilerplate configuration.
- **Embedded Servers:** No need for external servers.
- **Production-Ready Features:** Built-in features for monitoring, metrics, and security.
- **Strong Community and Ecosystem:** Large community and extensive documentation.
- **Simplified deployment:** Package your applications as standalone JAR files or WAR files for easy deployment.
- **Cloud-native support:** Seamless integration with cloud platforms like AWS, Azure, and GCP.

Spring Boot Components



Spring Boot Components

- **Spring Boot:** This is the core of the framework. It provides the SpringApplication class (for creating standalone applications), support for embedded web servers (like Tomcat), and externalized configurations (like database connection details). Think of it as the foundation upon which the other components build.
- **Spring Boot CLI Component:** This is a developer-friendly command-line tool (CLI) that compiles and runs Groovy code. It can also watch for file changes and restart the application automatically, which makes prototyping Spring applications quicker and easier.
- **Spring Boot Test Component:** This enables the use of annotations and methods to write test cases for the Spring Boot application.

Spring Boot Components

- **Spring Boot DevTools Component:** This provides an additional toolkit aimed at improving the development experience. Features include automatic restart upon code changes and LiveReload to refresh the browser when HTML changes.
- **Spring Boot Autoconfiguration Component:** Autoconfiguration automatically configures the bare minimum components needed for a Spring application based on classpath JAR files and properties. It makes educated guesses to configure Spring beans based on the presence of dependencies.
- **Spring Boot Actuator Component:** This provides actuator endpoints to interact with, monitor, and audit a Spring Boot application. These endpoints offer health checks, thread dumps, and other useful metrics to manage the application in production.

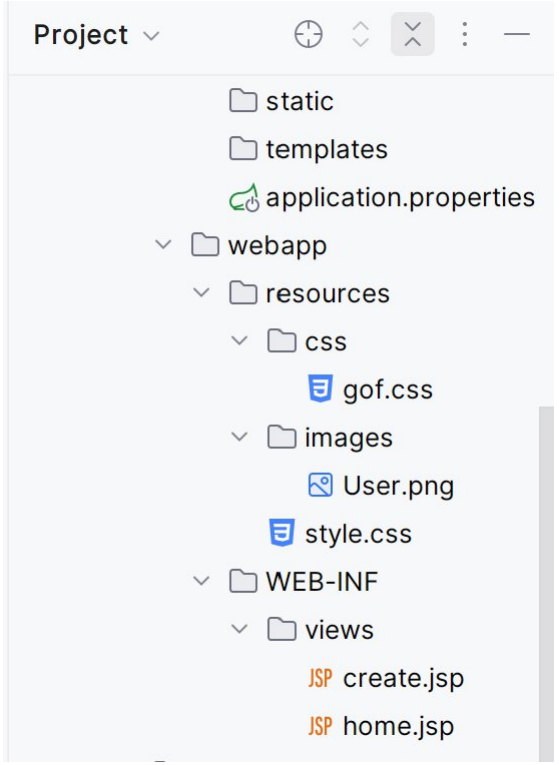
FPT Education
FPT UNIVERSITY

Java

Spring Boot Programming Demo

Key Points

- **Step 01 :** Create webapp in the src/main/webapp directory and its importance for JSP files.



Key Points

- **Step 02:** Config application.properties for JSP

```
spring.application.name=DemoLab01
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
```

- **Step 03 : Config the pom.xml**

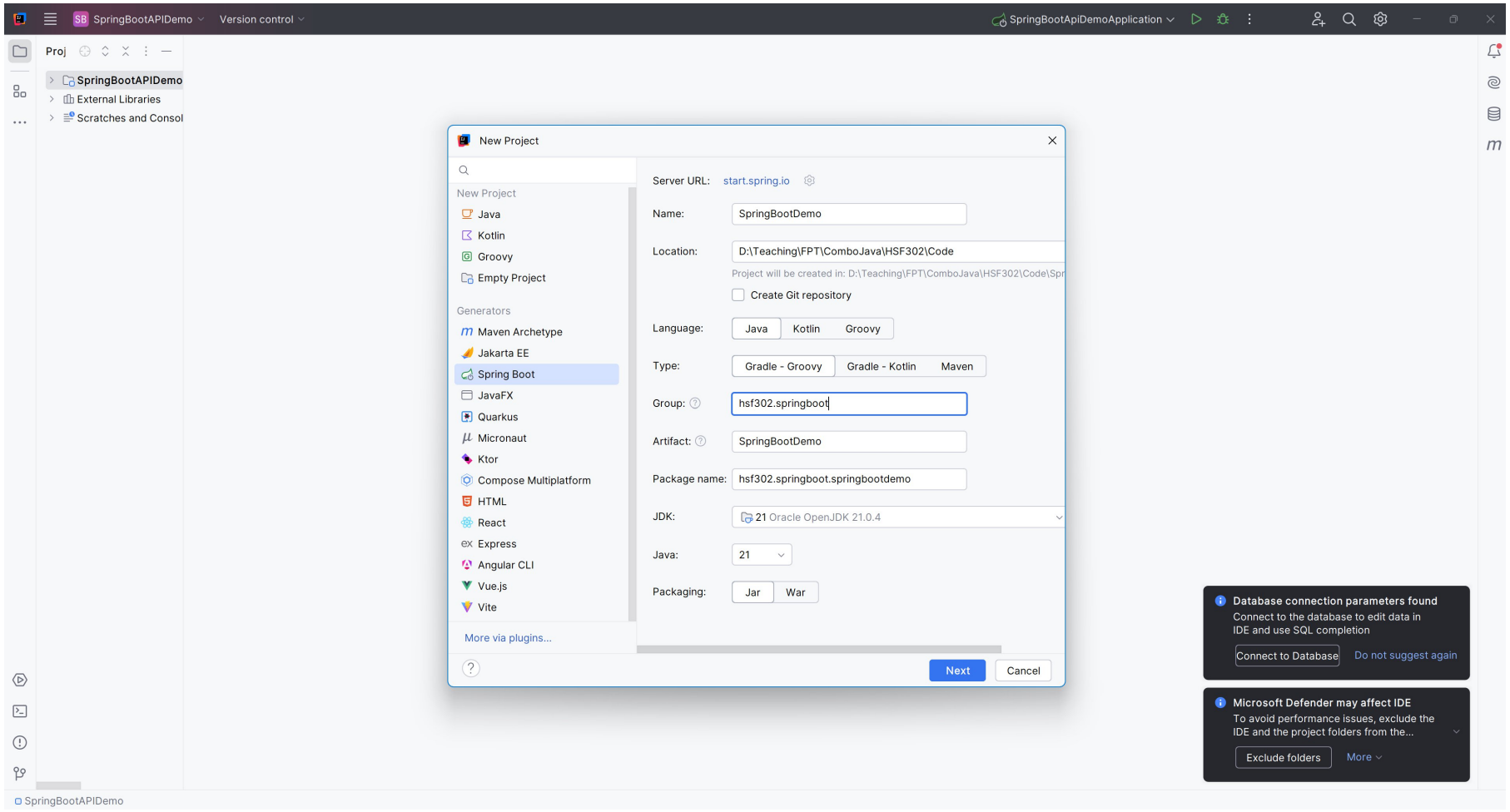
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.1</version>
</dependency>
```

Key Points

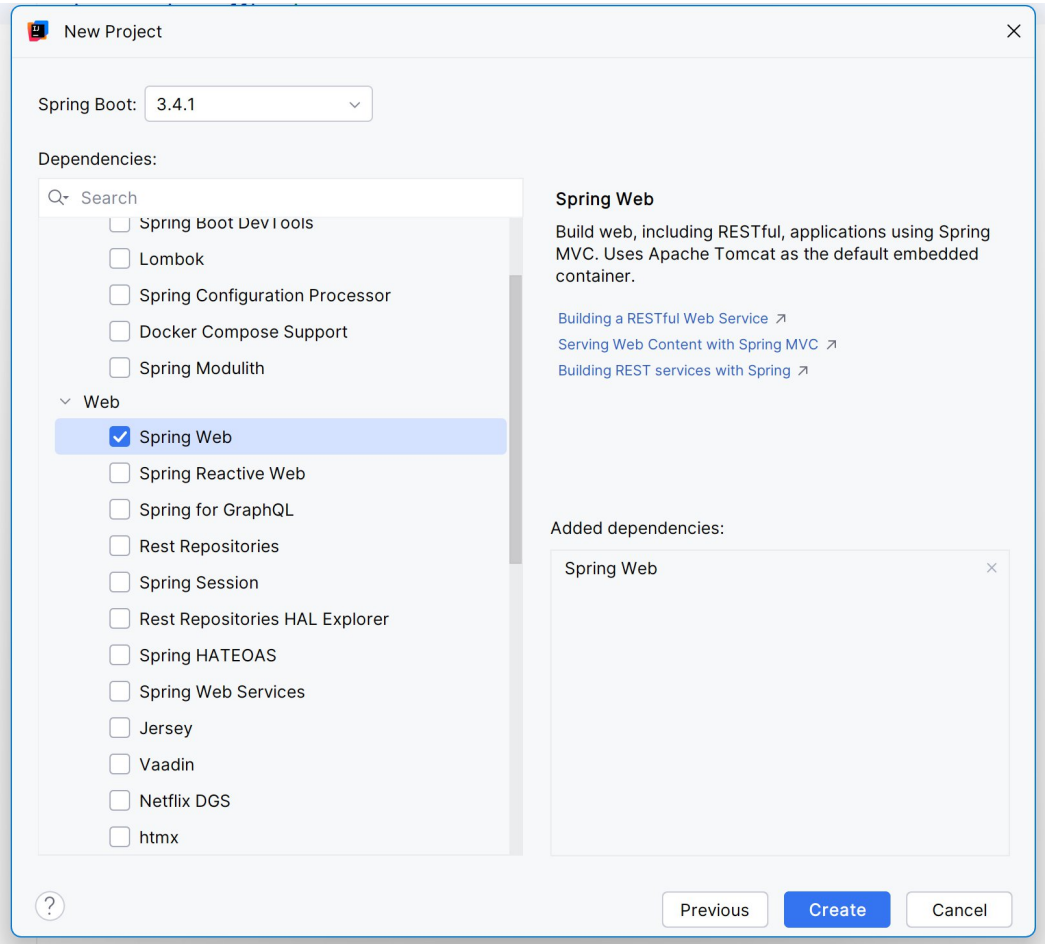
- **Step 03 : Config the pom.xml**

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.1</version>
</dependency>
```

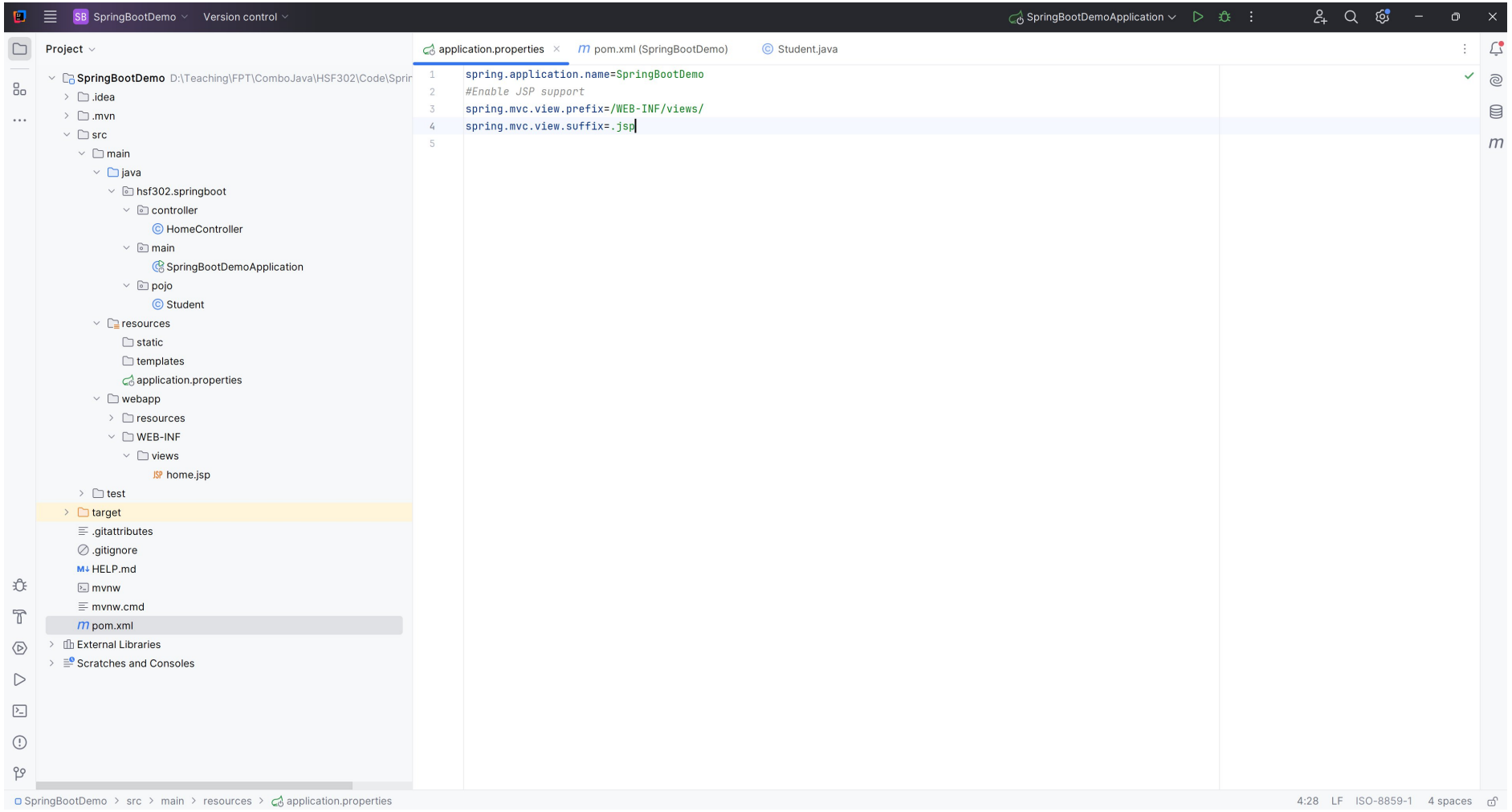
Open IntelliJ, File | New | Maven Project



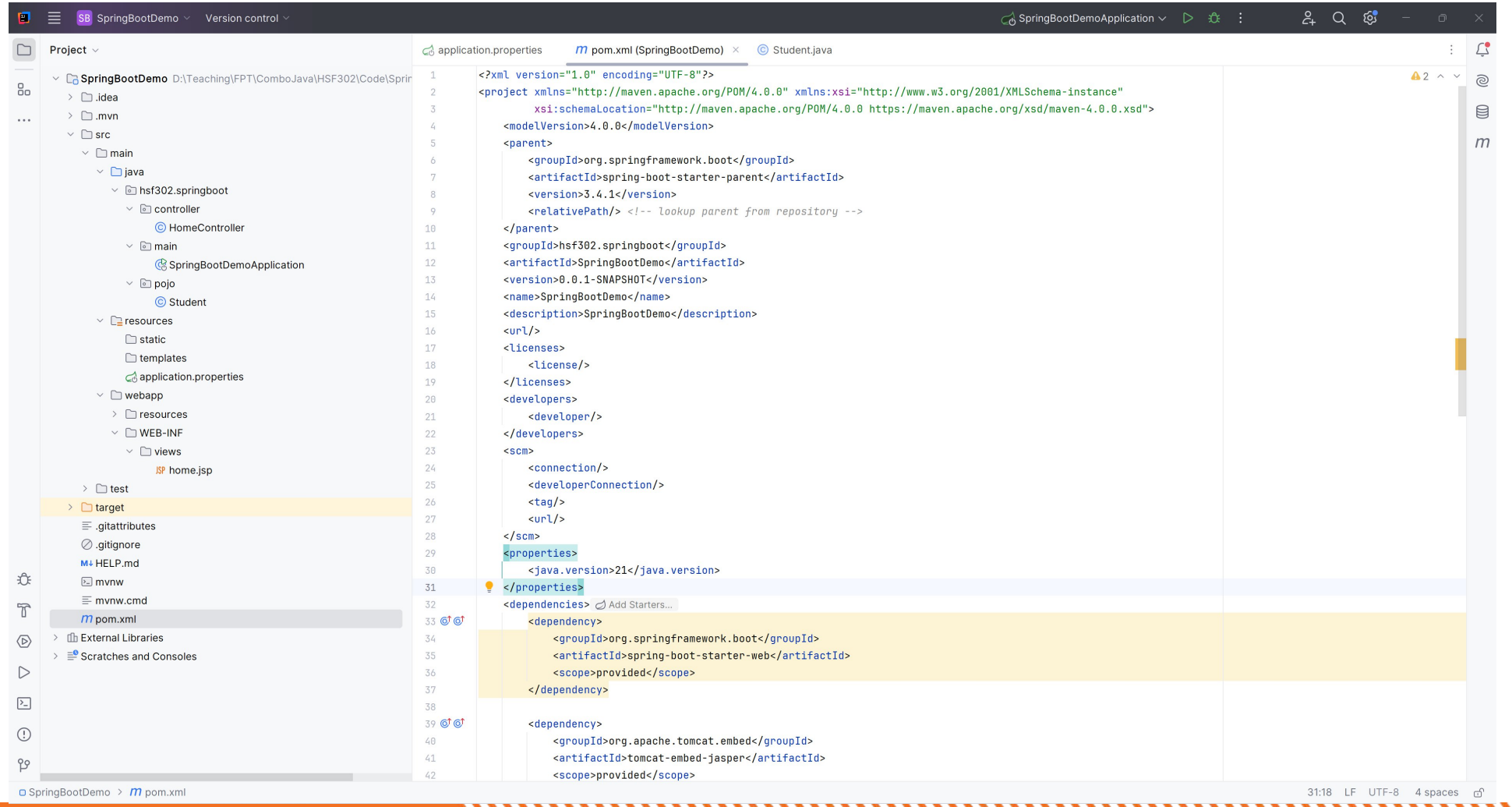
Add Dependences



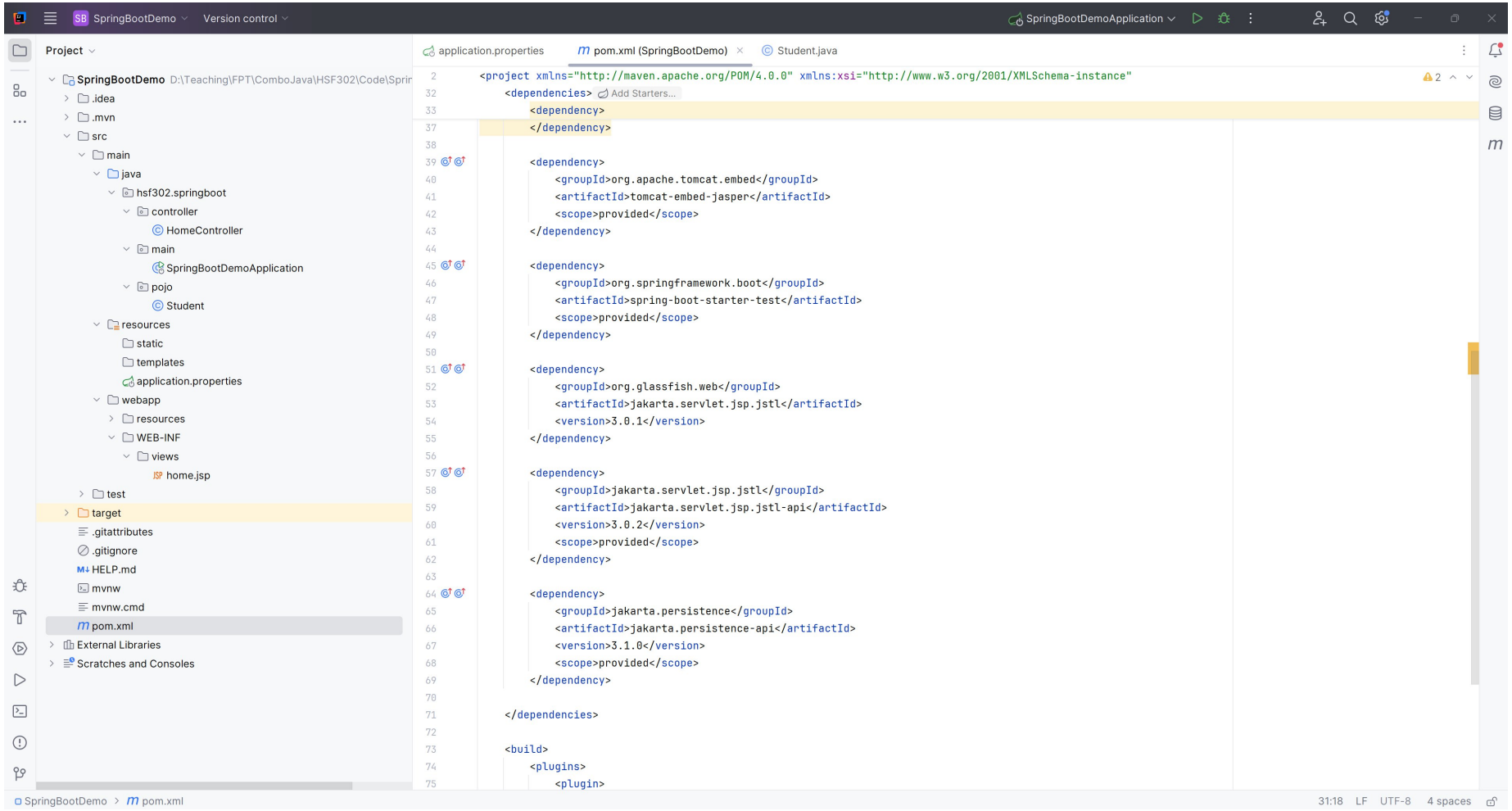
Create the Structure Project



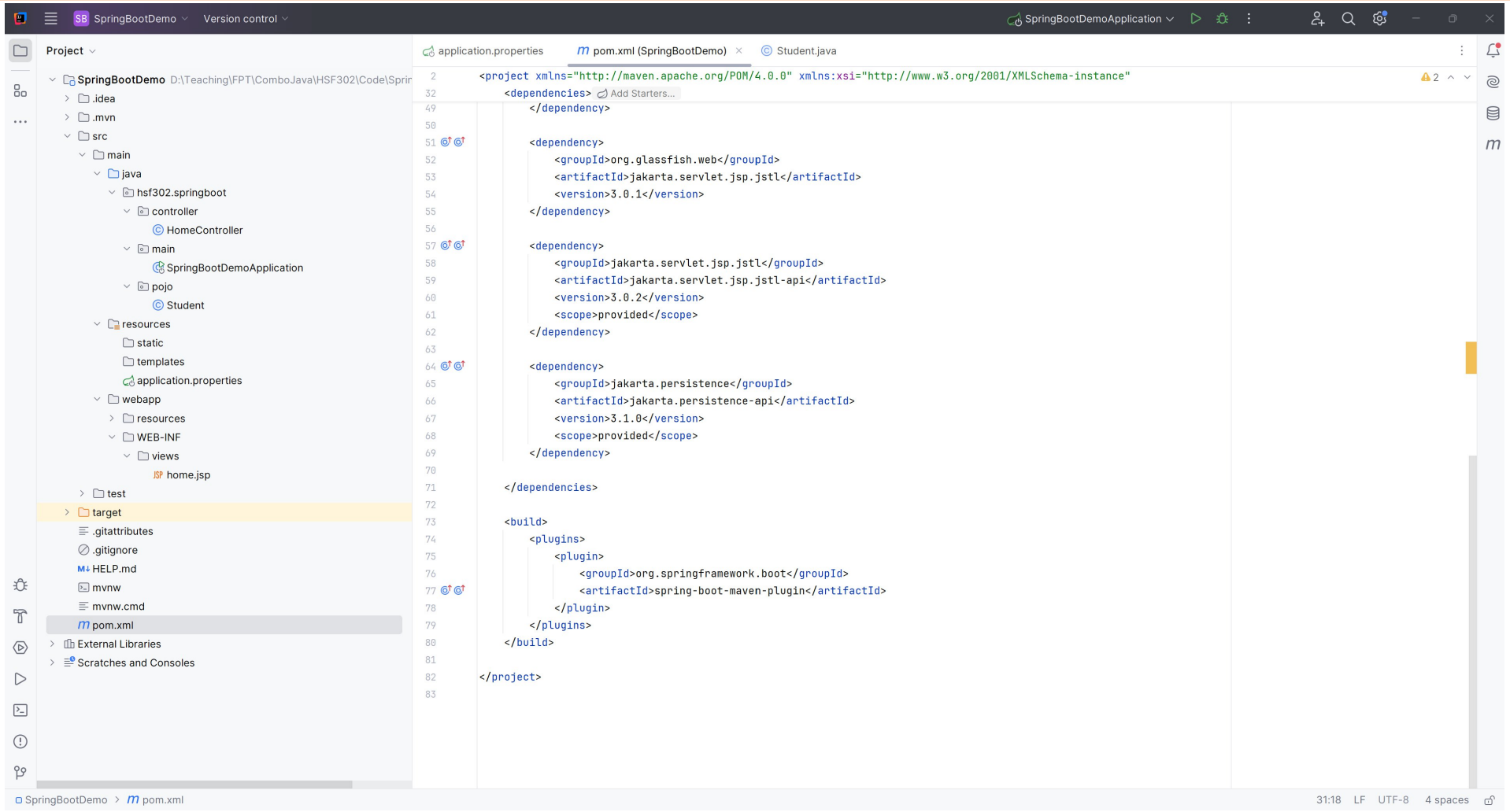
Edit pom.xml


SpringBootDemo
Version control
SpringBootDemoApplication
Project
application.properties
pom.xml (SpringBootDemo)
Student.java
SpringBootDemo D:\Teaching\FPT\ComboJava\HSF302\Code\Sprir
.idea
.mvn
src
main
java
hsf302.springboot
controller
HomeController
main
SpringBootDemoApplication
pojo
Student
resources
static
templates
application.properties
webapp
resources
WEB-INF
views
home.jsp
test
target
.gitattributes
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>3.4.1</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>hsf302.springboot</groupId>
<artifactId>SpringBootDemo</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>SpringBootDemo</name>
<description>SpringBootDemo</description>
<url/>
<licenses>
<license/>
</licenses>
<developers>
<developer/>
</developers>
<scm>
<connection/>
<developerConnection/>
<tag/>
<url/>
</scm>
<properties>
<java.version>21</java.version>
</properties>
<dependencies> Add Starters...
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
<scope>provided</scope>
SpringBootDemo
pom.xml
31:18
LF
UTF-8
4 spaces

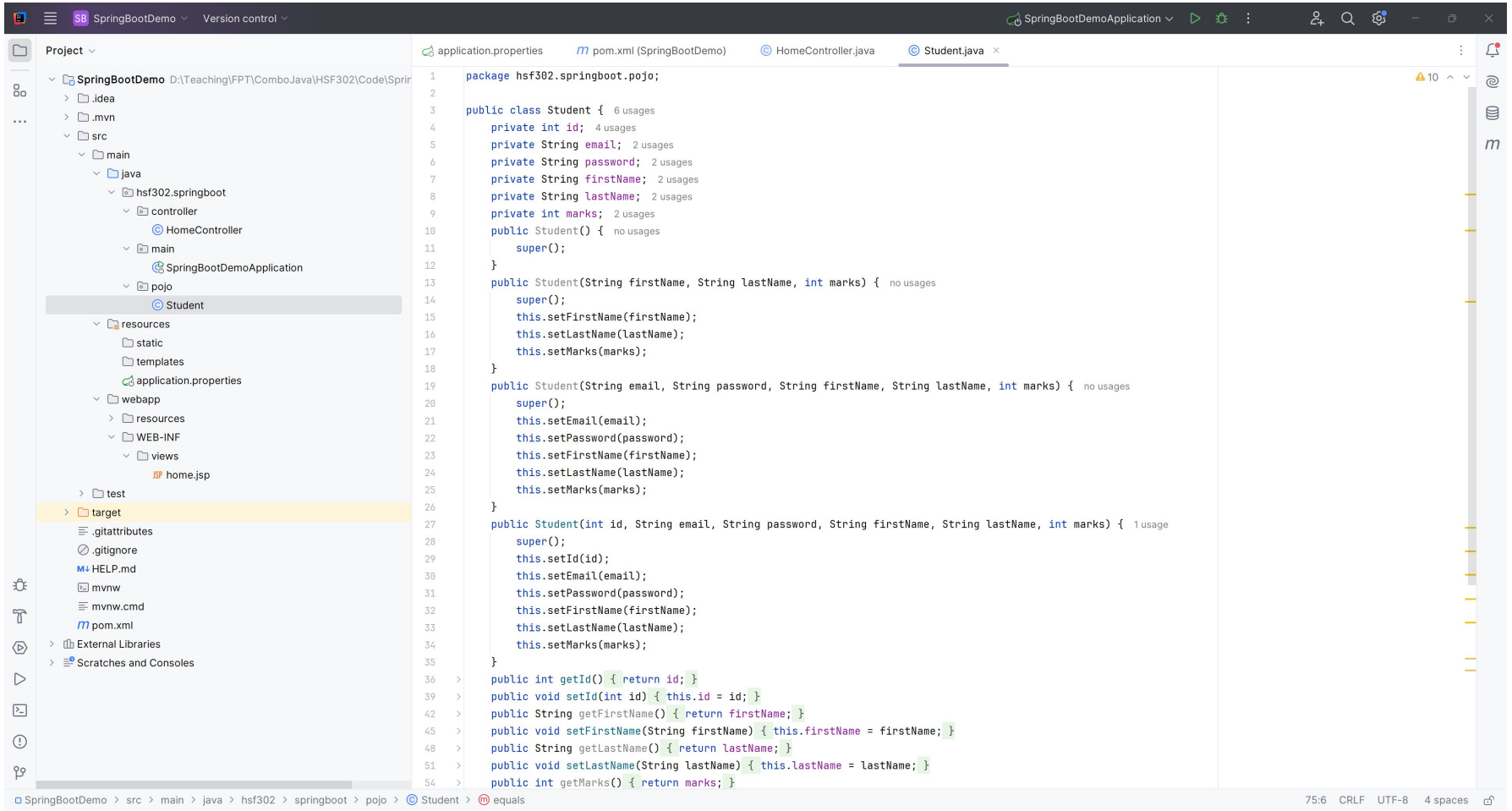
Edit pom.xml


SpringBootDemo
Version control
SpringBootDemoApplication
Project
application.properties
pom.xml (SpringBootDemo)
Student.java
SpringBootDemo D:\Teaching\FPT\ComboJava\HSF302\Code\Sprir
.idea
.mvn
src
main
java
hsf302.springboot
controller
HomeController
main
SpringBootDemoApplication
pojo
Student
resources
static
templates
application.properties
webapp
resources
WEB-INF
views
home.jsp
test
target
.gitattributes
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
<dependencies> Add Starters...
<dependency>
</dependency>
<dependency>
<groupId>org.apache.tomcat.embed</groupId>
<artifactId>tomcat-embed-jasper</artifactId>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>org.glassfish.web</groupId>
<artifactId>jakarta.servlet.jsp.jstl</artifactId>
<version>3.0.1</version>
</dependency>
<dependency>
<groupId>jakarta.servlet.jsp.jstl</groupId>
<artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
<version>3.0.2</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>jakarta.persistence</groupId>
<artifactId>jakarta.persistence-api</artifactId>
<version>3.1.0</version>
<scope>provided</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
SpringBootDemo
pom.xml
31:18
LF
UTF-8
4 spaces

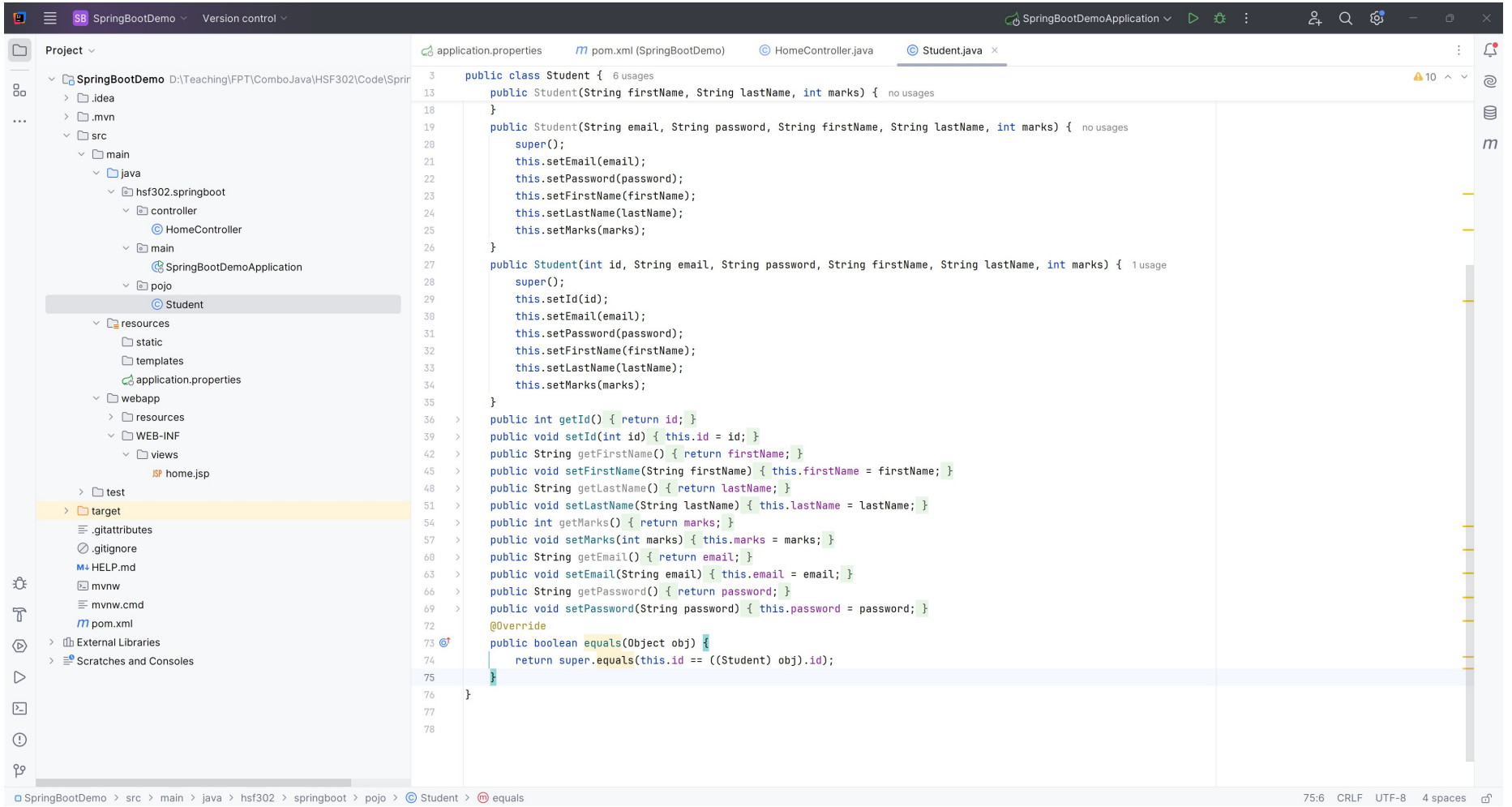
Edit pom.xml


SpringBootDemo
Version control
SpringBootDemoApplication
Project
application.properties
pom.xml (SpringBootDemo)
Student.java
SpringBootDemo D:\Teaching\FPT\ComboJava\HSF302\Code\Sprir
.idea
.mvn
src
main
java
hsf302.springboot
controller
HomeController
main
SpringBootDemoApplication
pojo
Student
resources
static
templates
application.properties
webapp
resources
WEB-INF
views
home.jsp
test
target
.gitattributes
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
<dependencies> Add Starters...
</dependency>
<dependency>
<groupId>org.glassfish.web</groupId>
<artifactId>jakarta.servlet.jsp.jstl</artifactId>
<version>3.0.1</version>
</dependency>
<dependency>
<groupId>jakarta.servlet.jsp.jstl</groupId>
<artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
<version>3.0.2</version>
<scope>provided</scope>
</dependency>
<dependency>
<groupId>jakarta.persistence</groupId>
<artifactId>jakarta.persistence-api</artifactId>
<version>3.1.0</version>
<scope>provided</scope>
</dependency>
</dependencies>
<build>
<plugins>
<plugin>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
</plugins>
</build>
</project>
SpringBootDemo > pom.xml
31:18 LF UTF-8 4 spaces

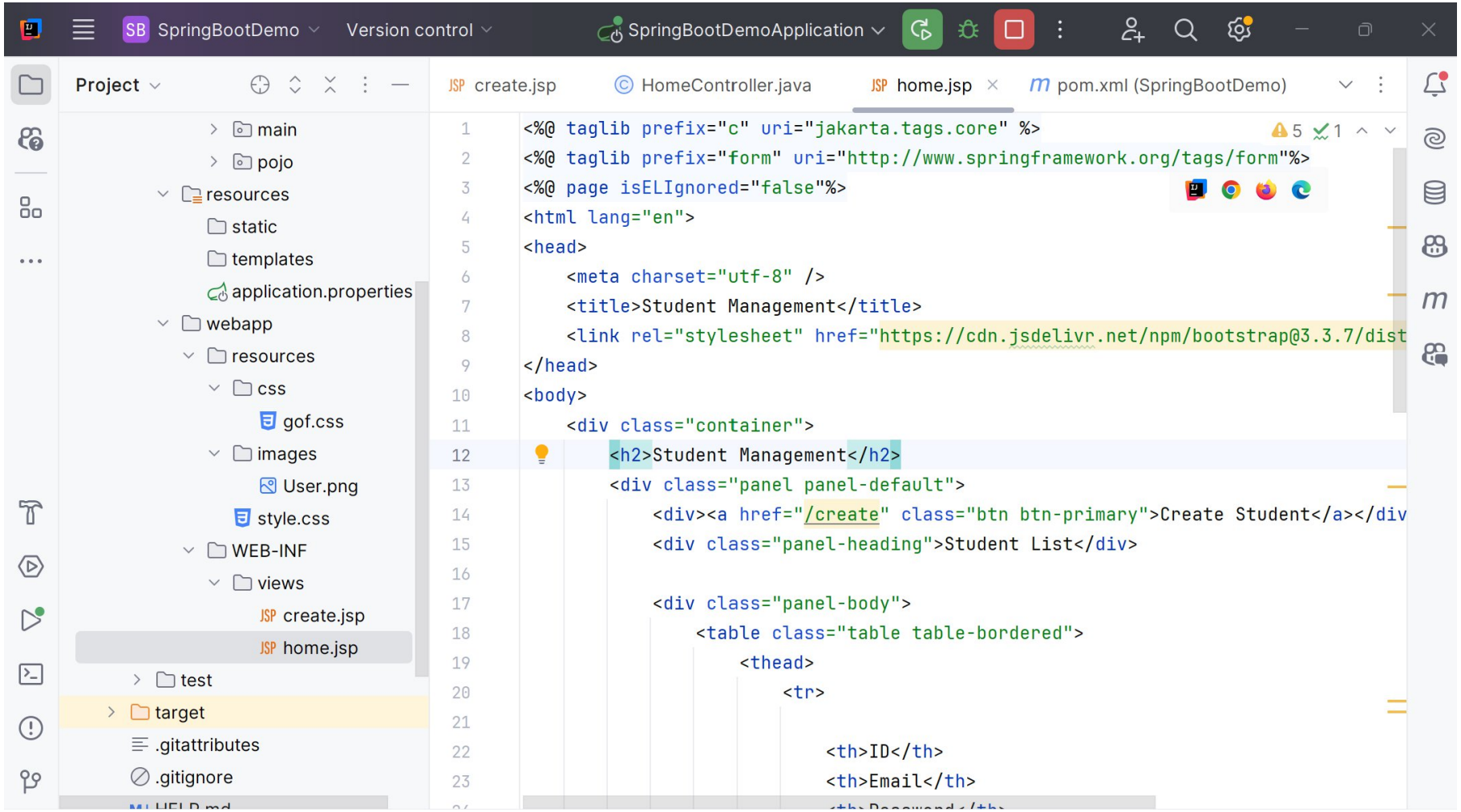
Create the Student.java in pojo



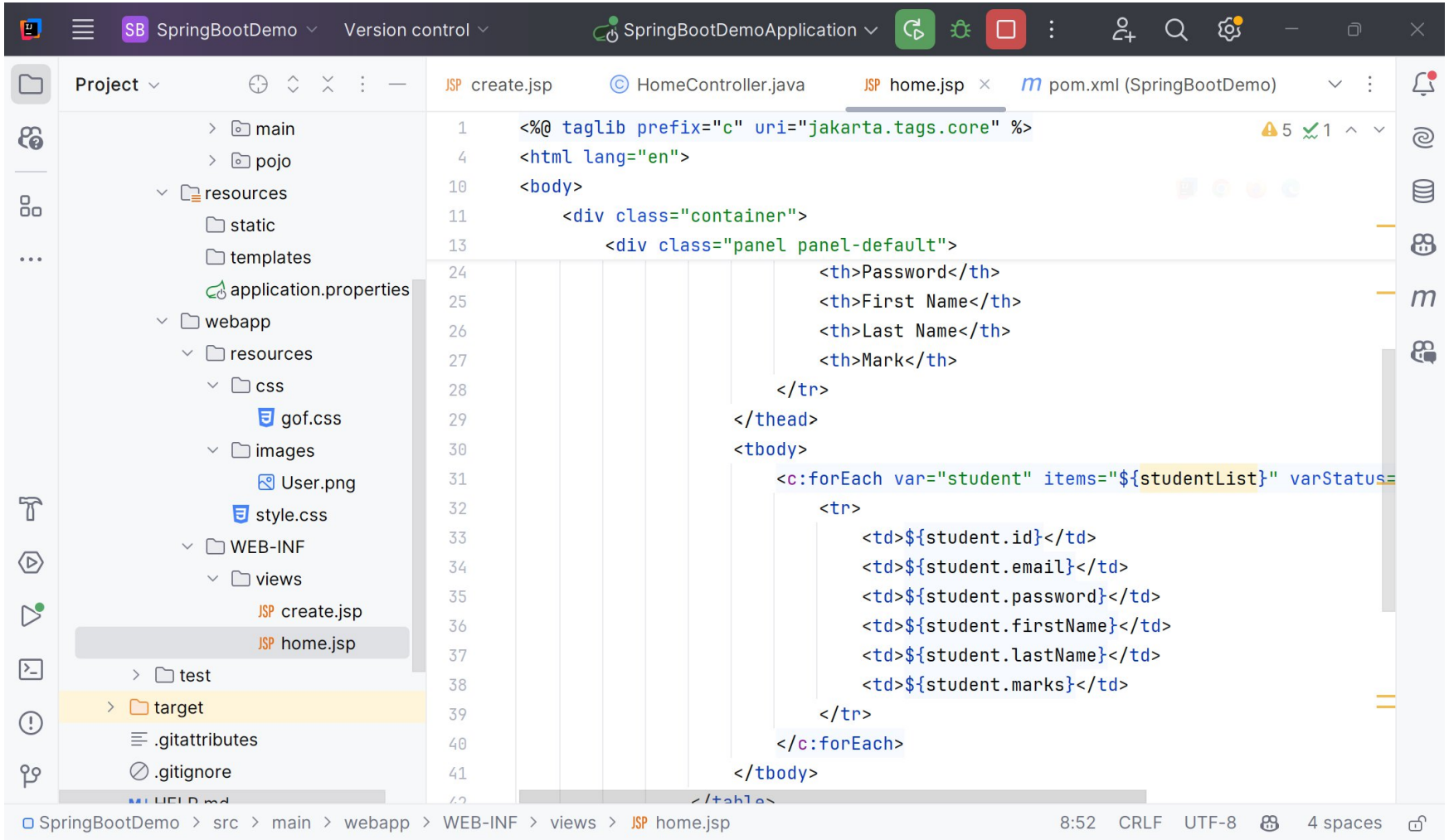
Create the Student.java in pojo



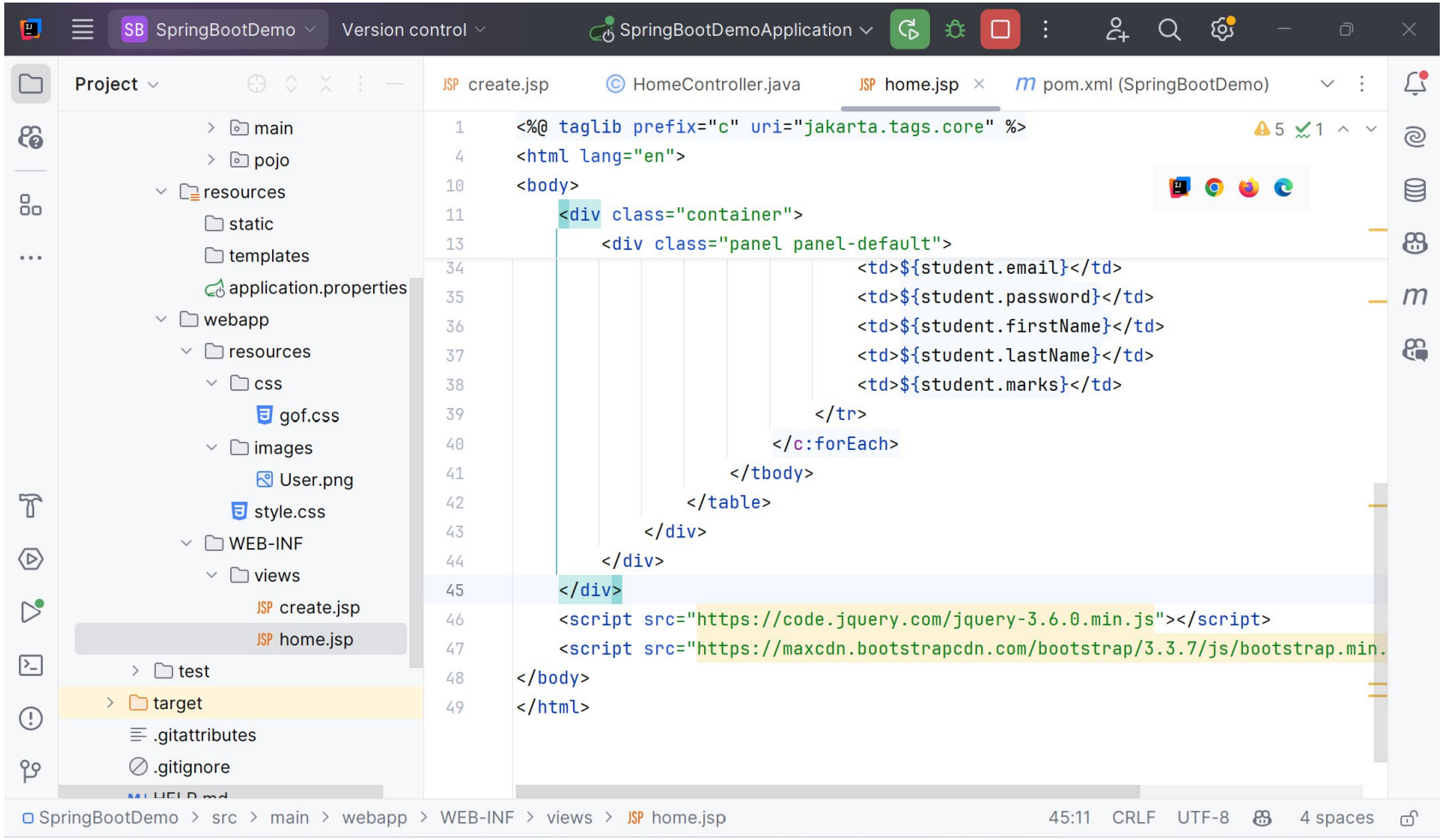
Create views : home.jsp



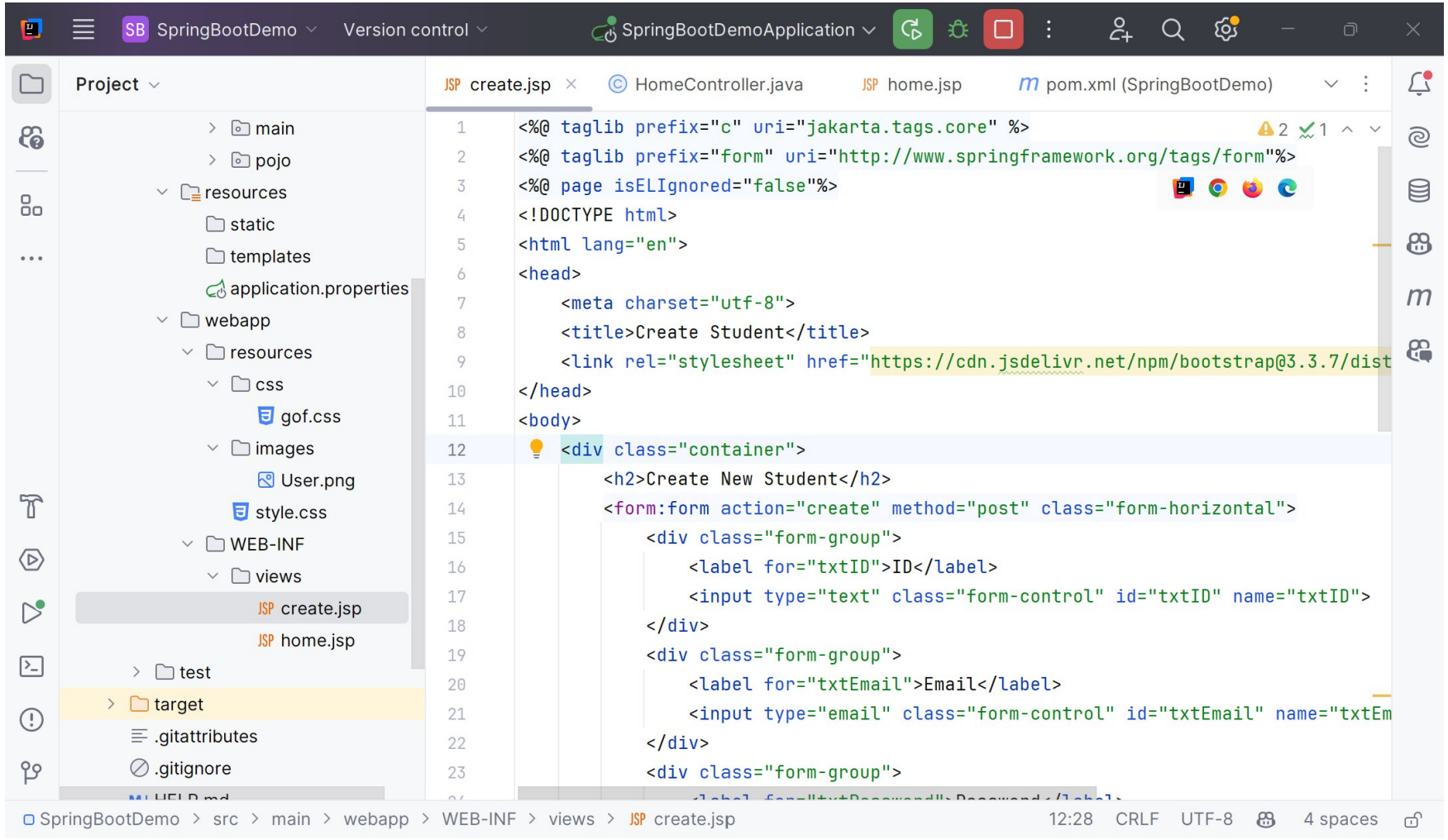
Create views : home.jsp



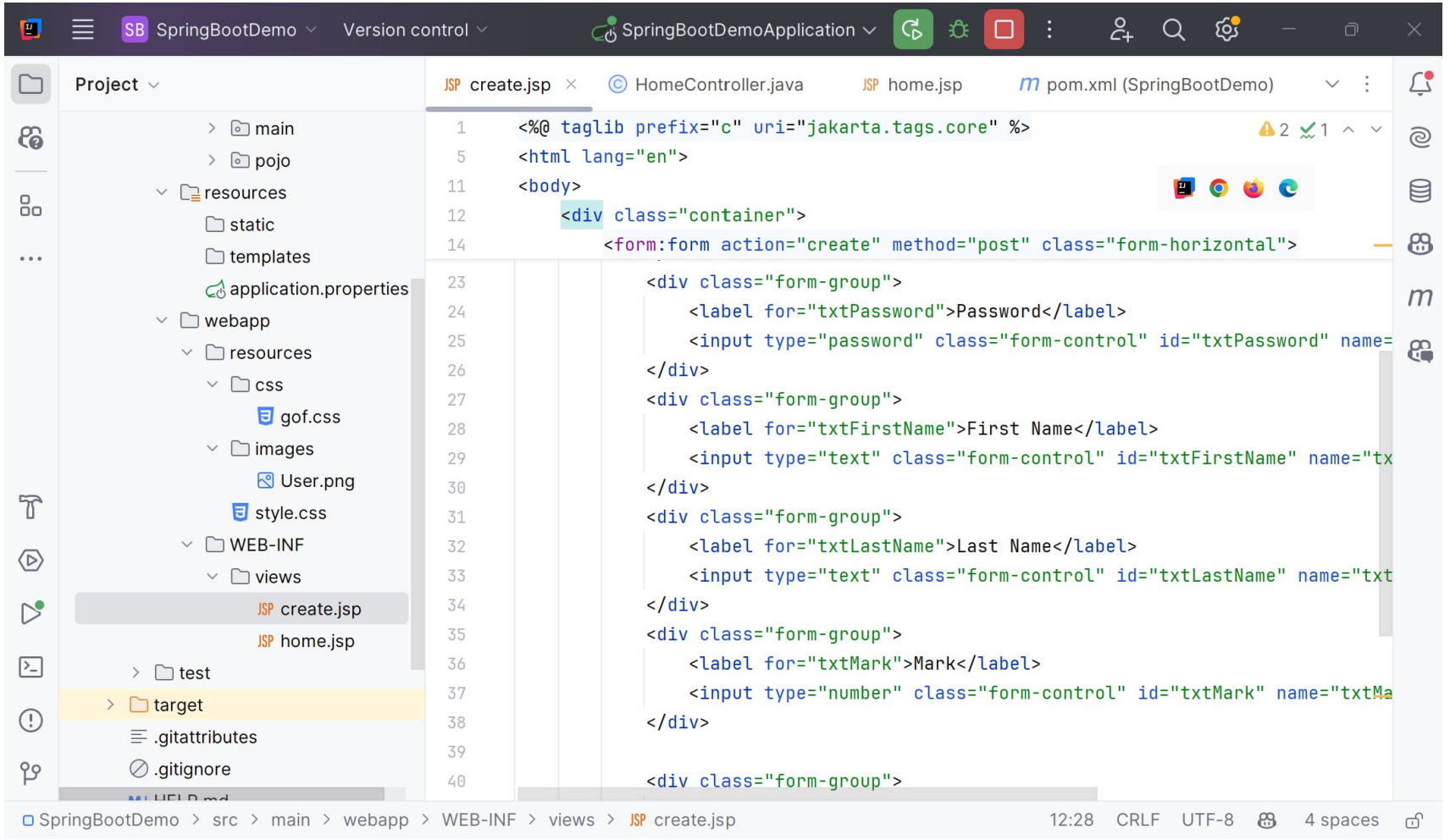
Create views : home.jsp



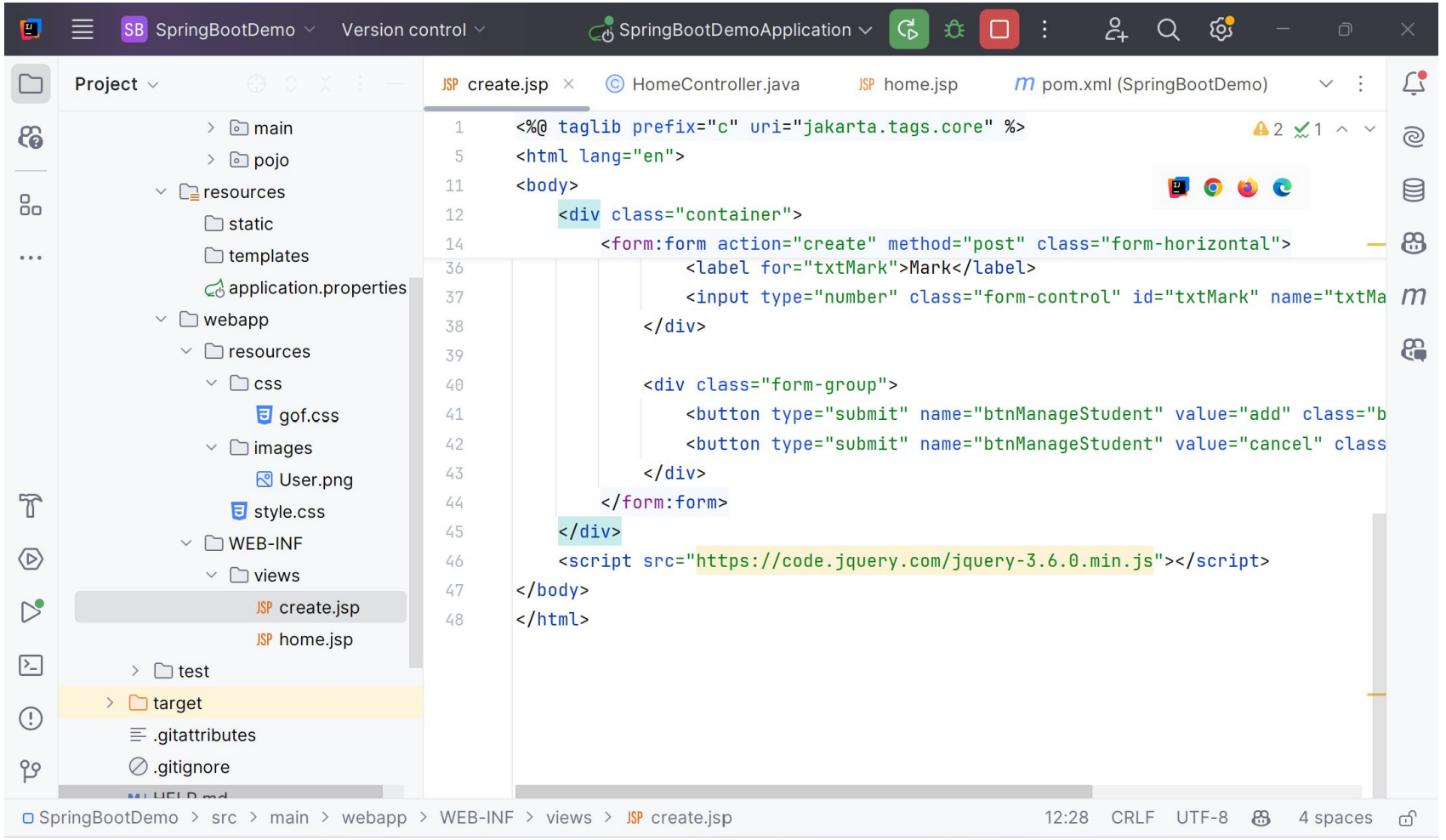
Create views : create.jsp



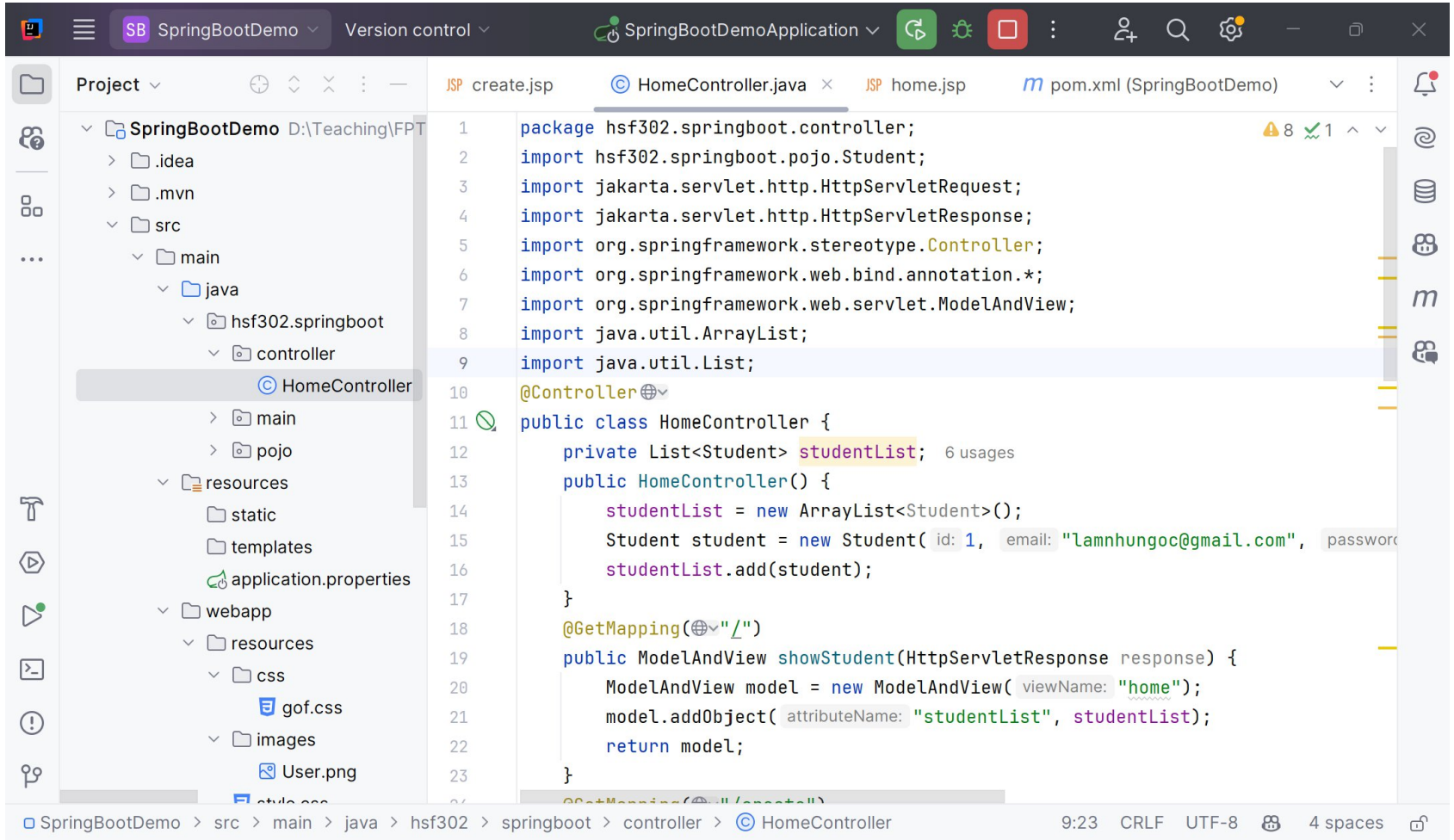
Create views : create.jsp



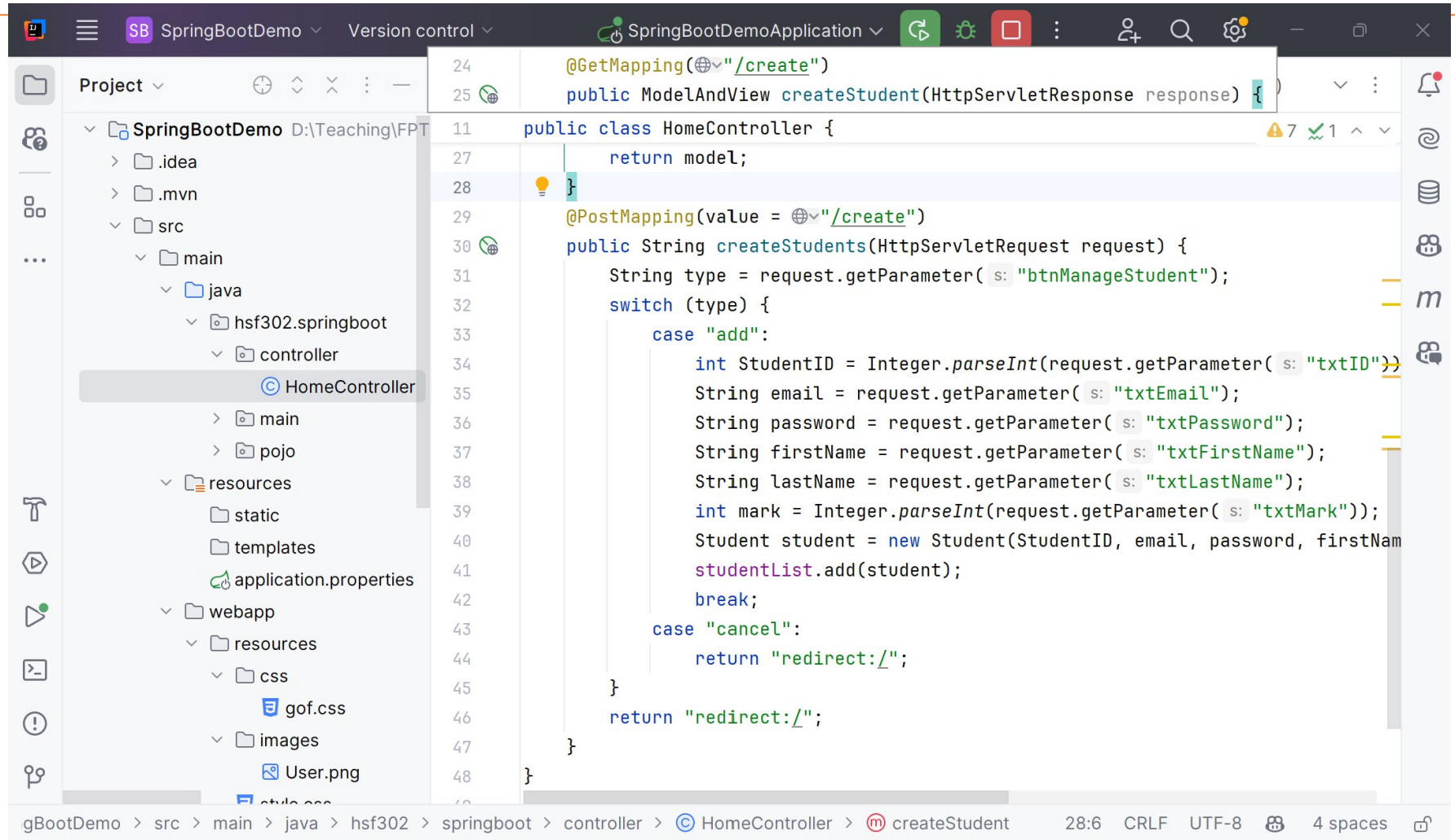
Create views : create.jsp



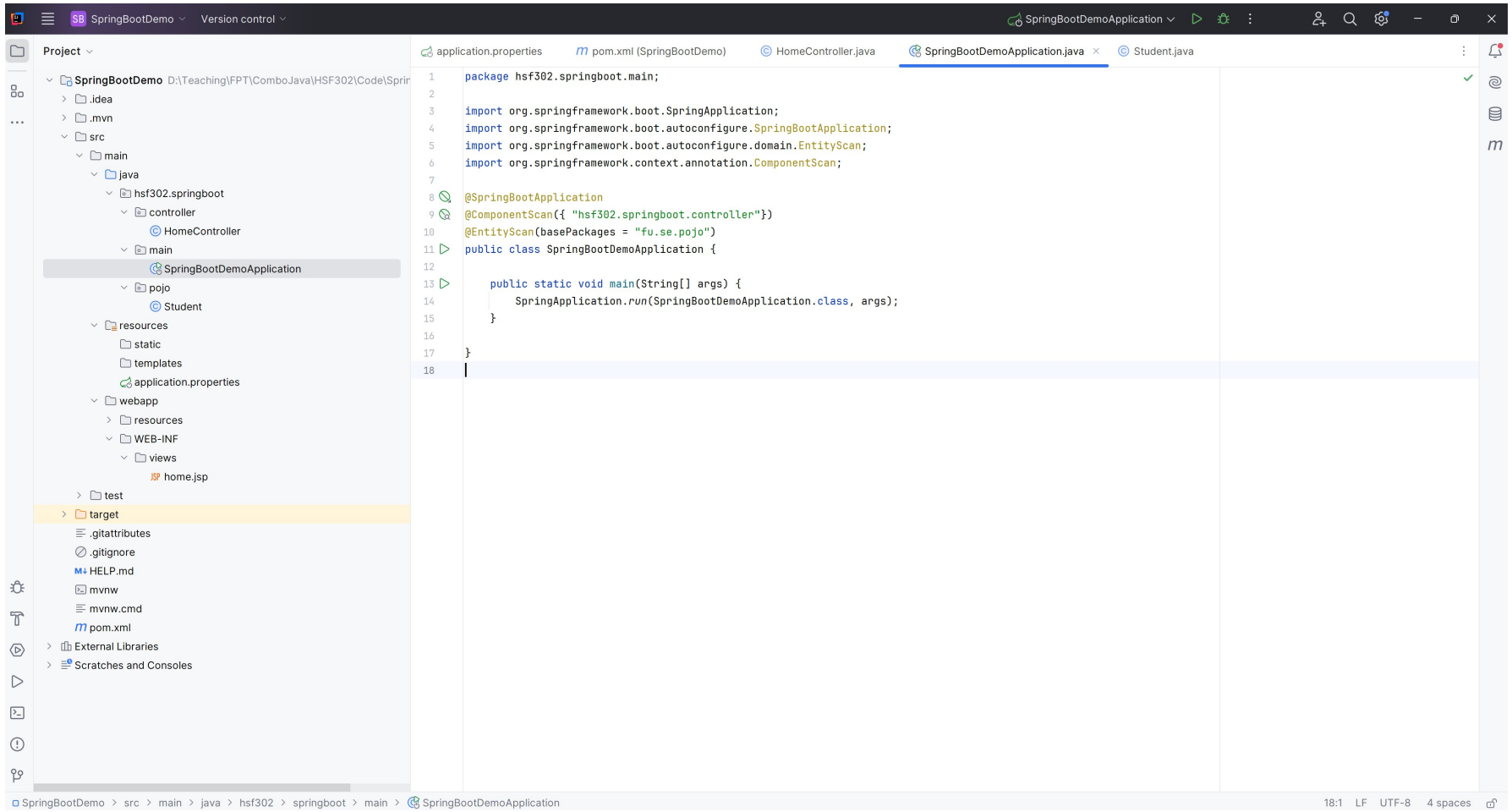
Create the HomeController.java



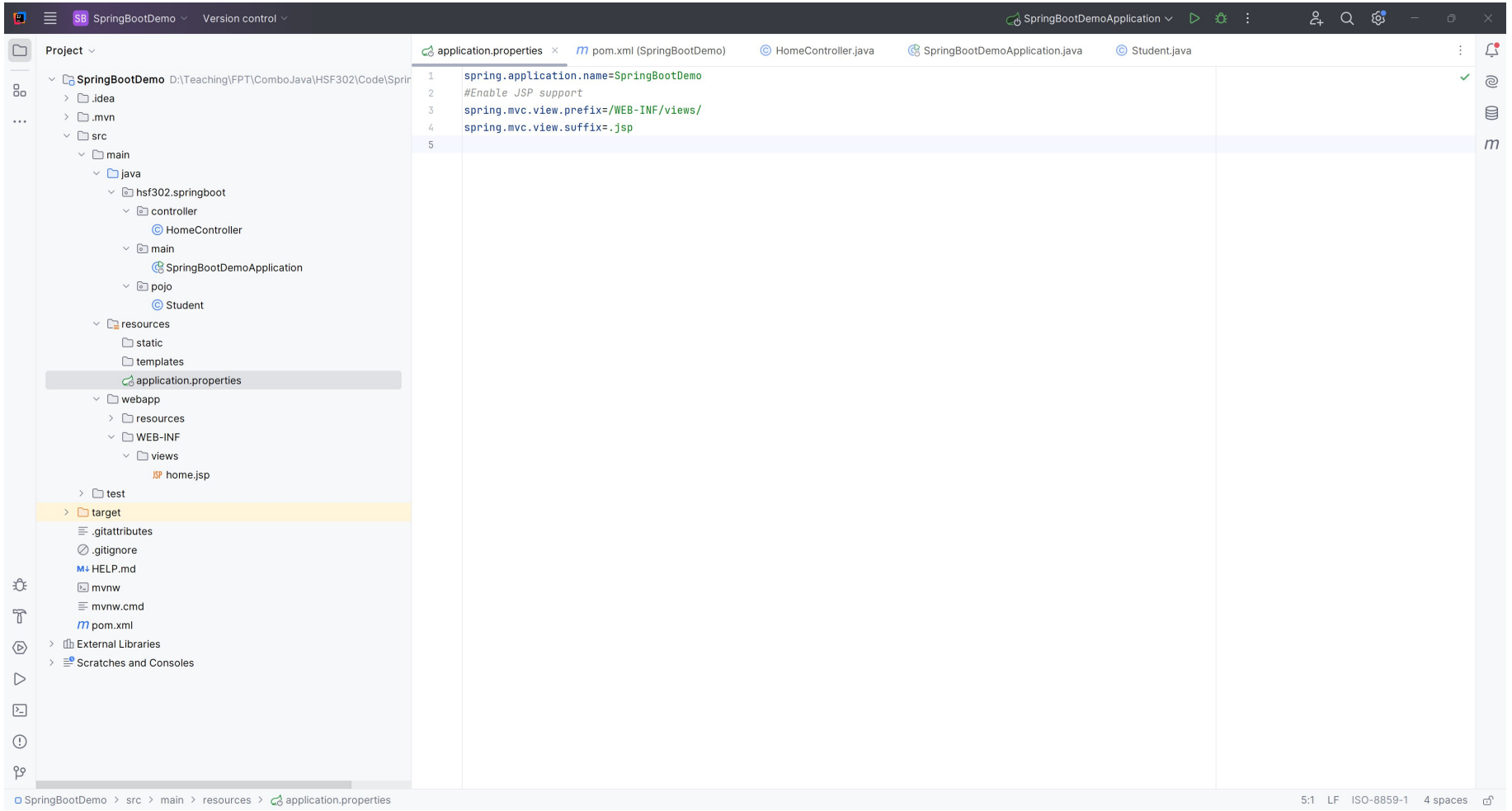
Create the HomeController.java



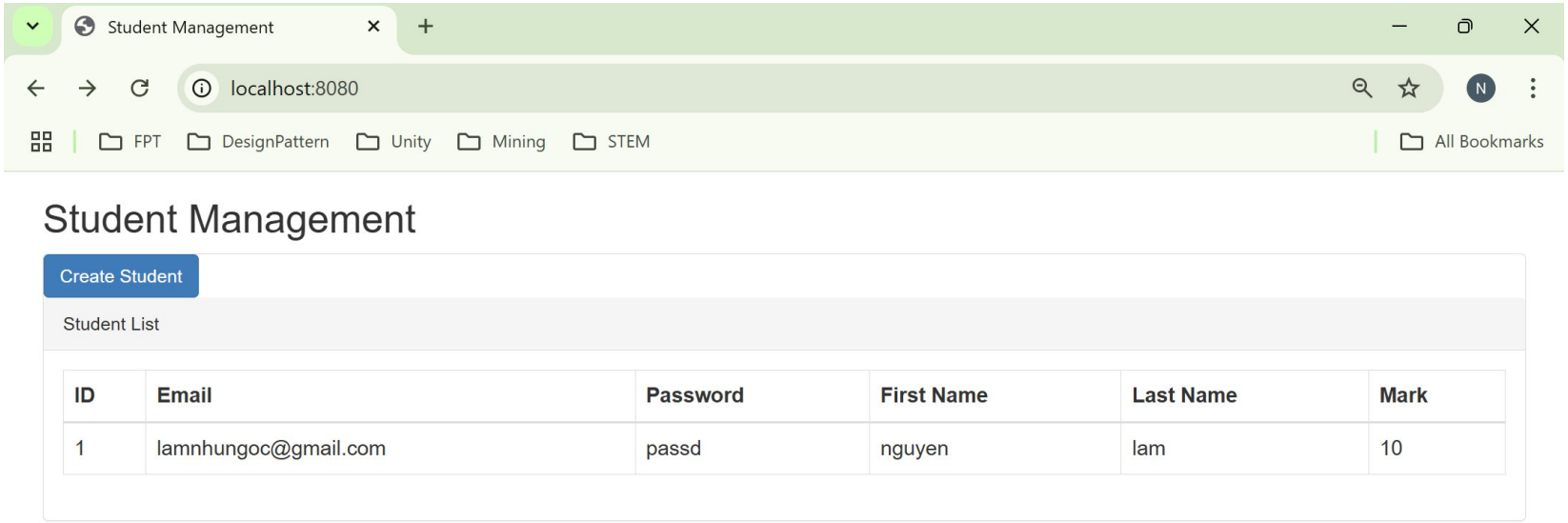
Edit SpringBoot Main



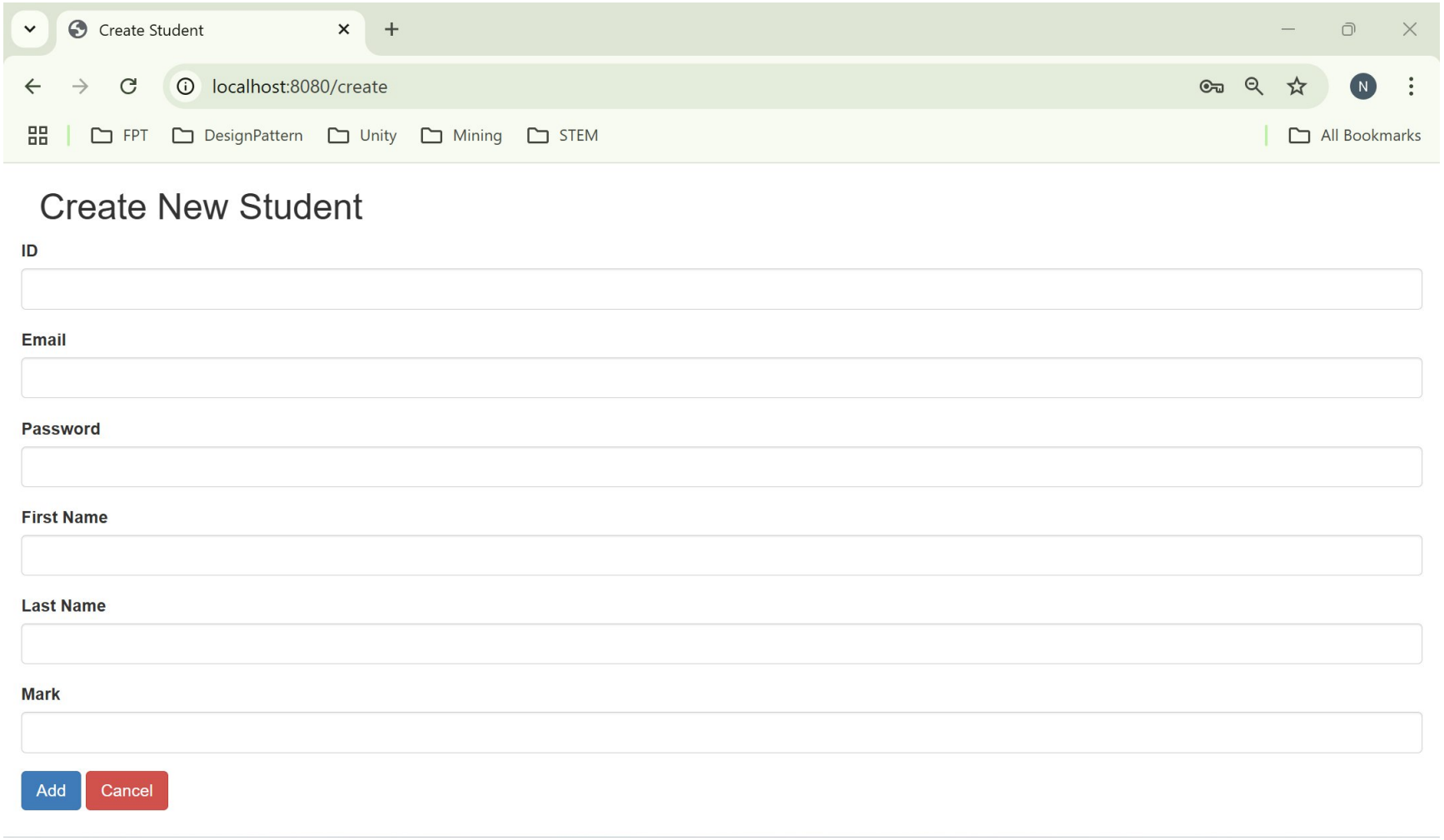
Edit the application.properties



Run Program


Student Management
localhost:8080
FPT
DesignPattern
Unity
Mining
STEM
All Bookmarks
Student Management
Create Student
Student List
ID
Email
Password
First Name
Last Name
Mark
1
lamnhungoc@gmail.com
passd
nguyen
lam
10

Result


Create Student
localhost:8080/create
FPT
DesignPattern
Unity
Mining
STEM
All Bookmarks
Create New Student
ID
Email
Password
First Name
Last Name
Mark
Add
Cancel

Summary

Concepts were introduced:

- Spring Boot
- Advantages of using Spring Boot
- Key features of Spring Boot
  - Demo Spring Boot with Collections