Lesson 01 Demo 02

Creating a Spring Boot Project in Eclipse

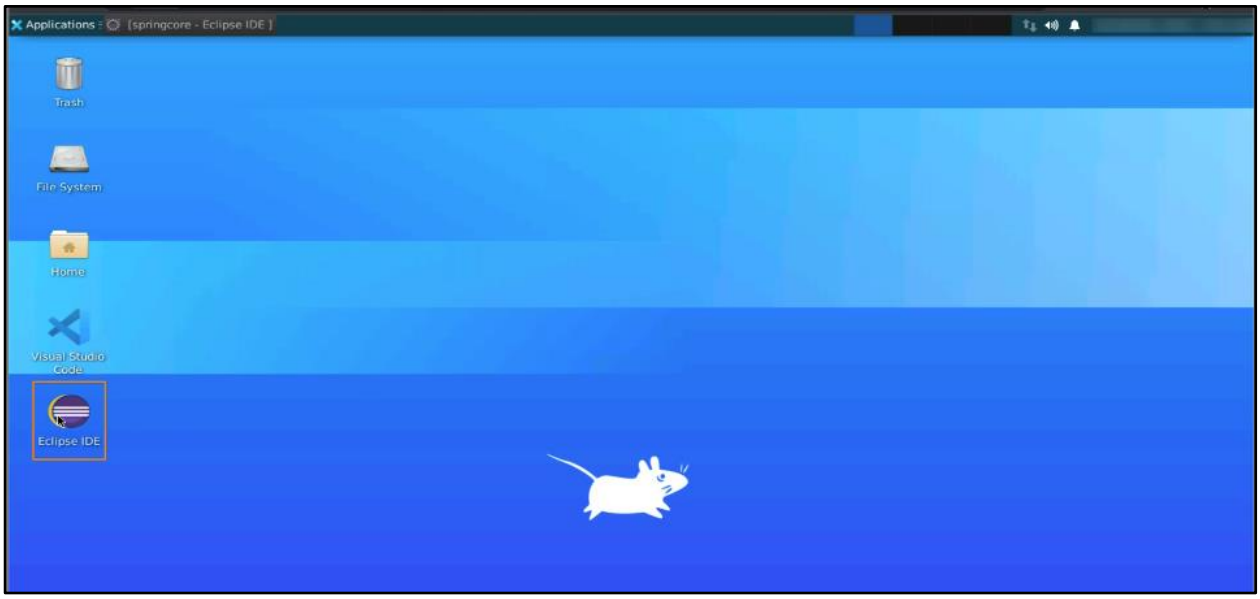
Objective: To create a Spring Boot project in Eclipse IDE **Tool required:** Eclipse IDE **Prerequisites:** None

Steps to be followed:

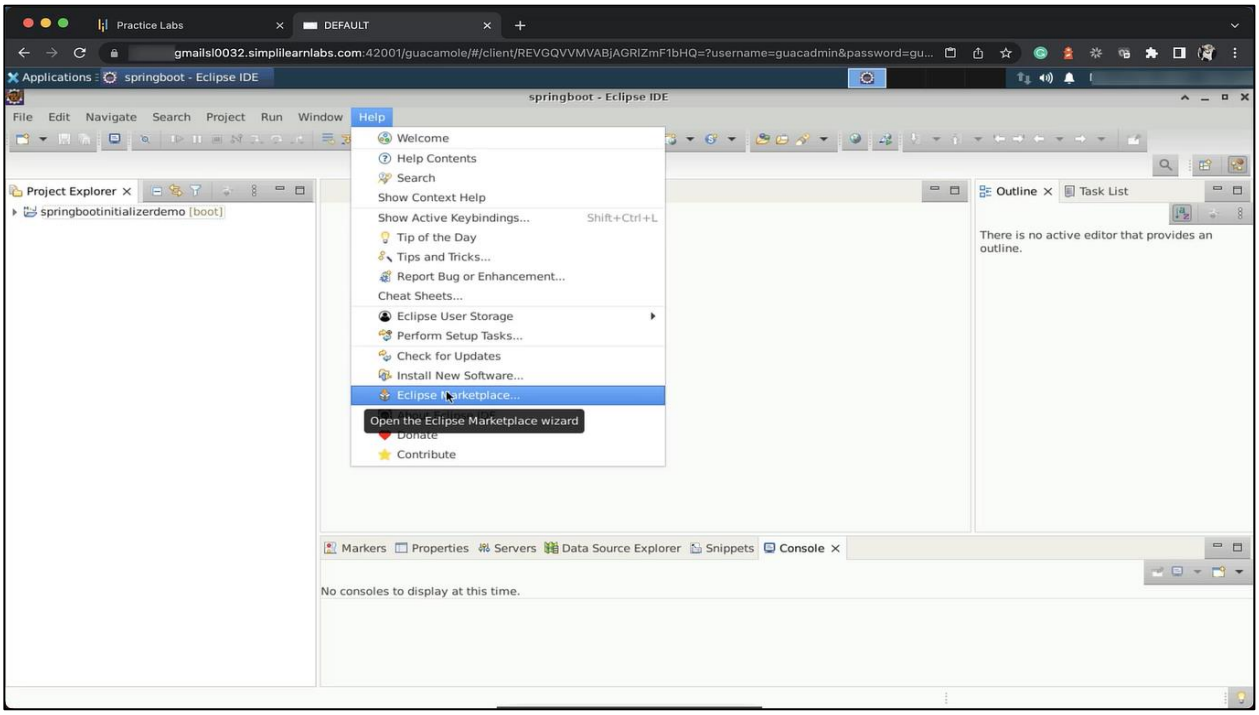
1. Installing the Spring Tool Suite plugin
2. Setting up a Spring Boot project in Eclipse IDE
3. Initializing the Spring Boot project

Step 1: Installing the Spring Tool Suite plugin

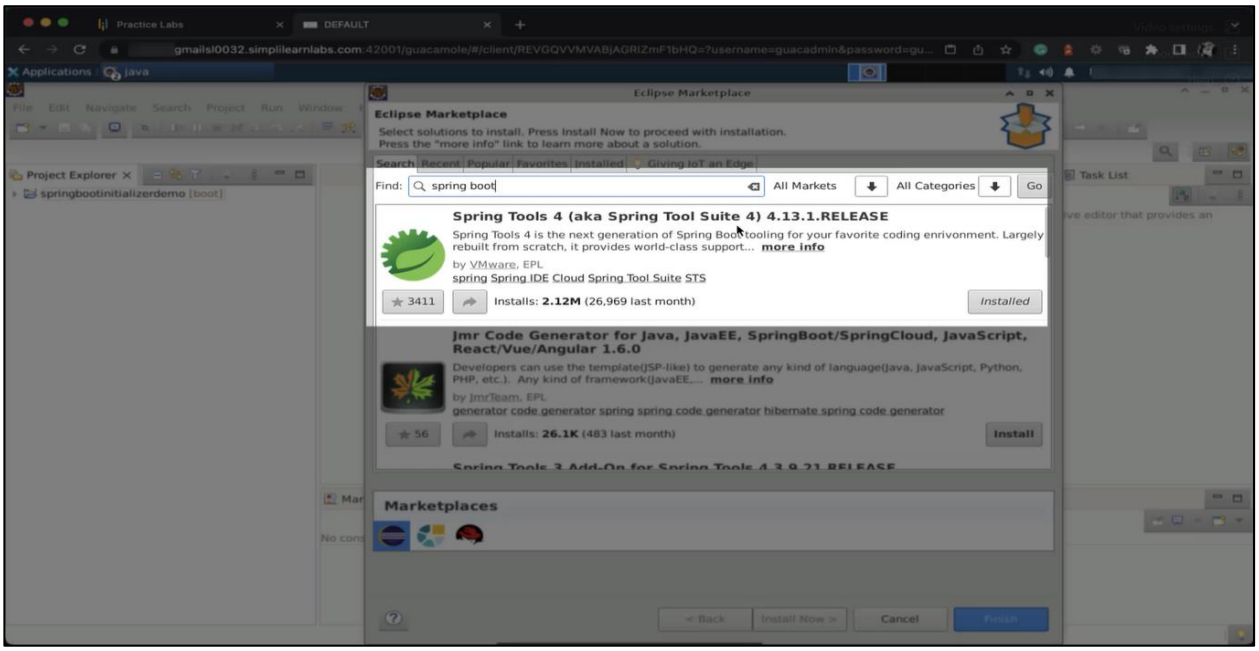
1.1 Open **Eclipse IDE**



1.2 Click on **Help** in the menu bar and select **Eclipse Marketplace**. We need to install a **Spring Tool Suite** plugin.



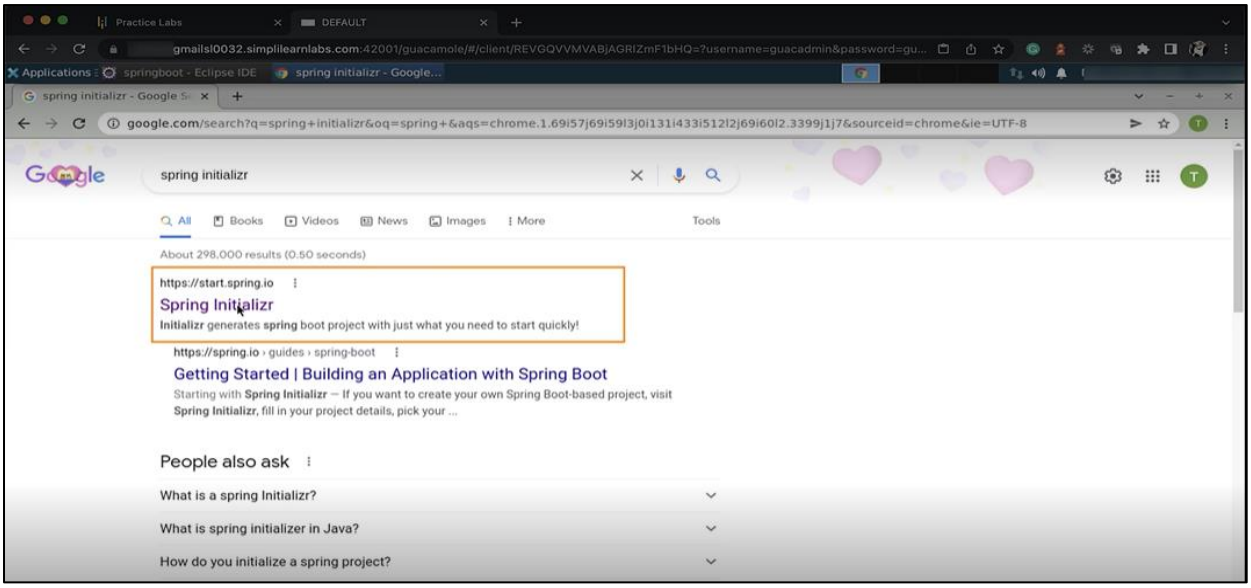
1.3 Search for **Spring Boot** in the search bar and install the **Spring Tools 4 (aka Spring Tool Suite 4)** plugin by accepting all the terms and conditions



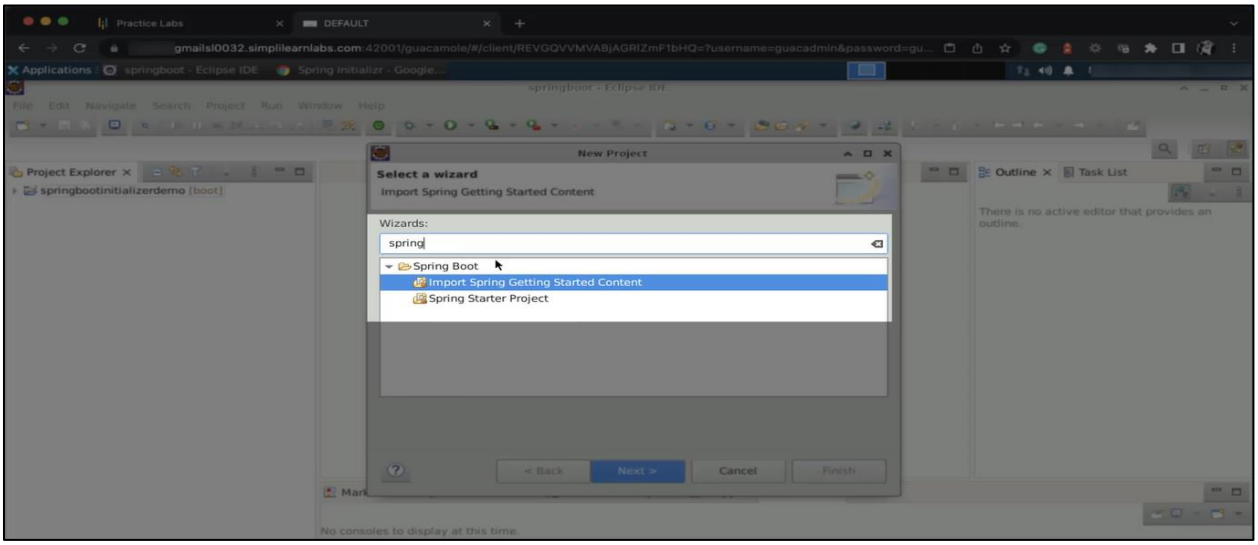
Note: The plugin will ask to restart Eclipse once it is installed in the Eclipse IDE.
Restart the IDE to open **Spring boot – Eclipse IDE**

Step 2: Setting up a Spring Boot project in Eclipse IDE

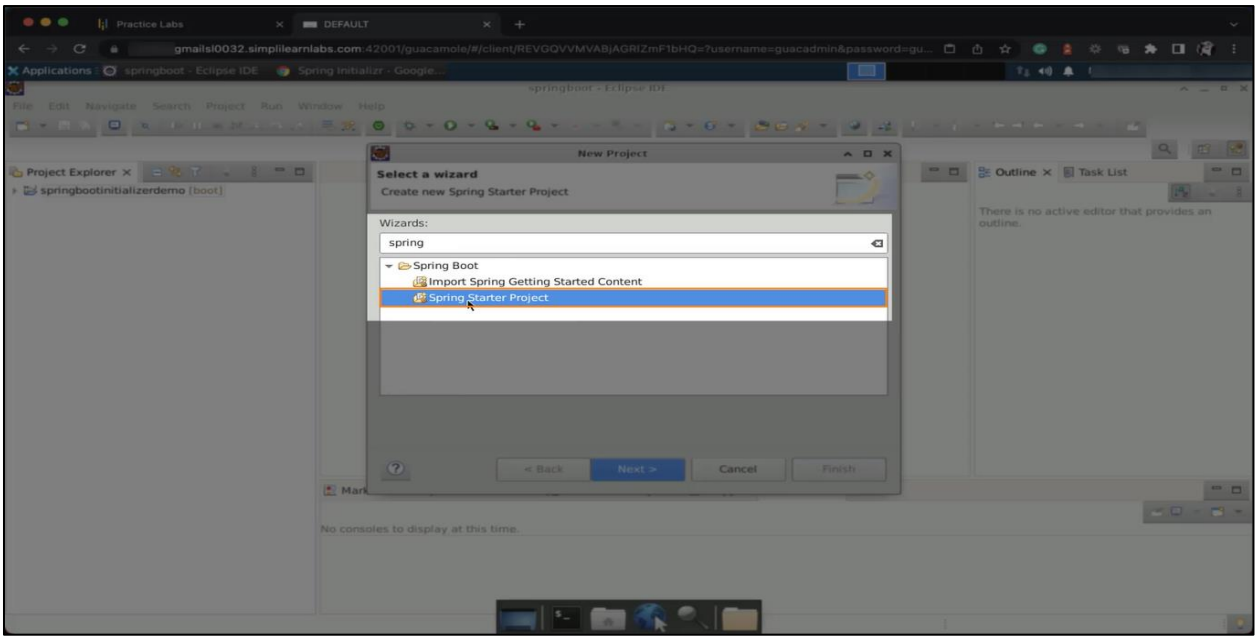
2.1 We can import the Spring Boot Project in two ways. First, search for the Spring Initializr in the browser and download the zip file. After that, import that project into the Eclipse IDE



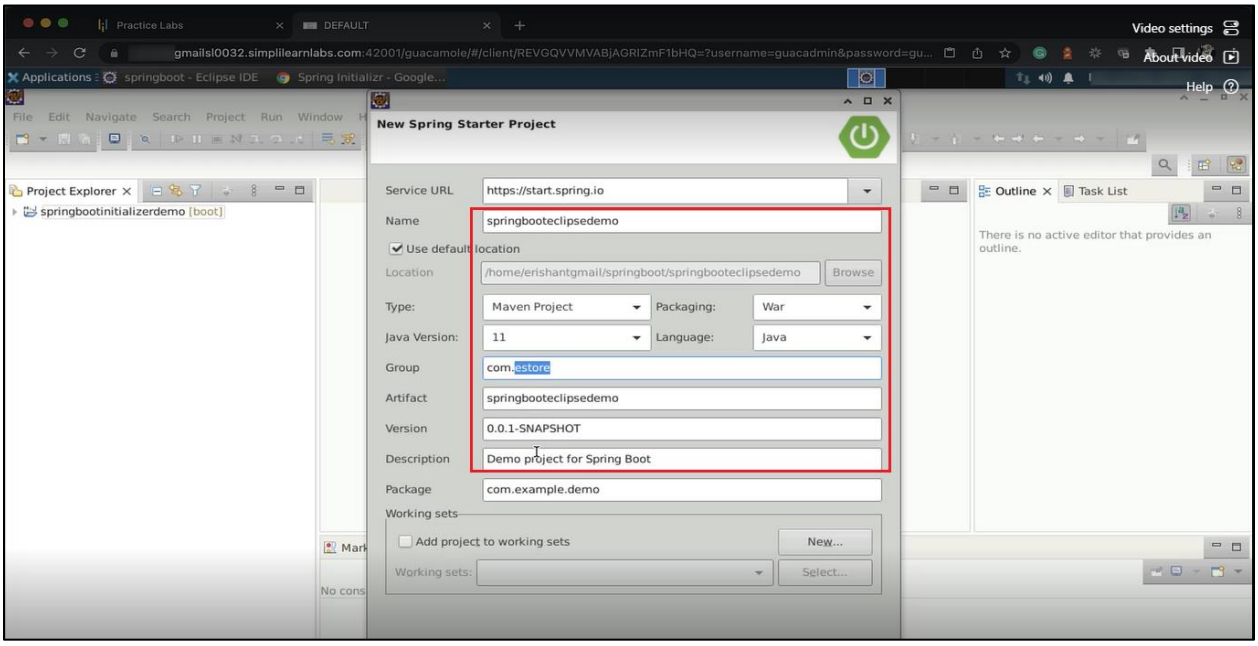
2.2 We can also import the Spring Boot project using the Eclipse IDE through the plugin that was installed in the above step. In the **Springboot - Eclipse IDE**, select **File > New > Project**, and search for **spring**



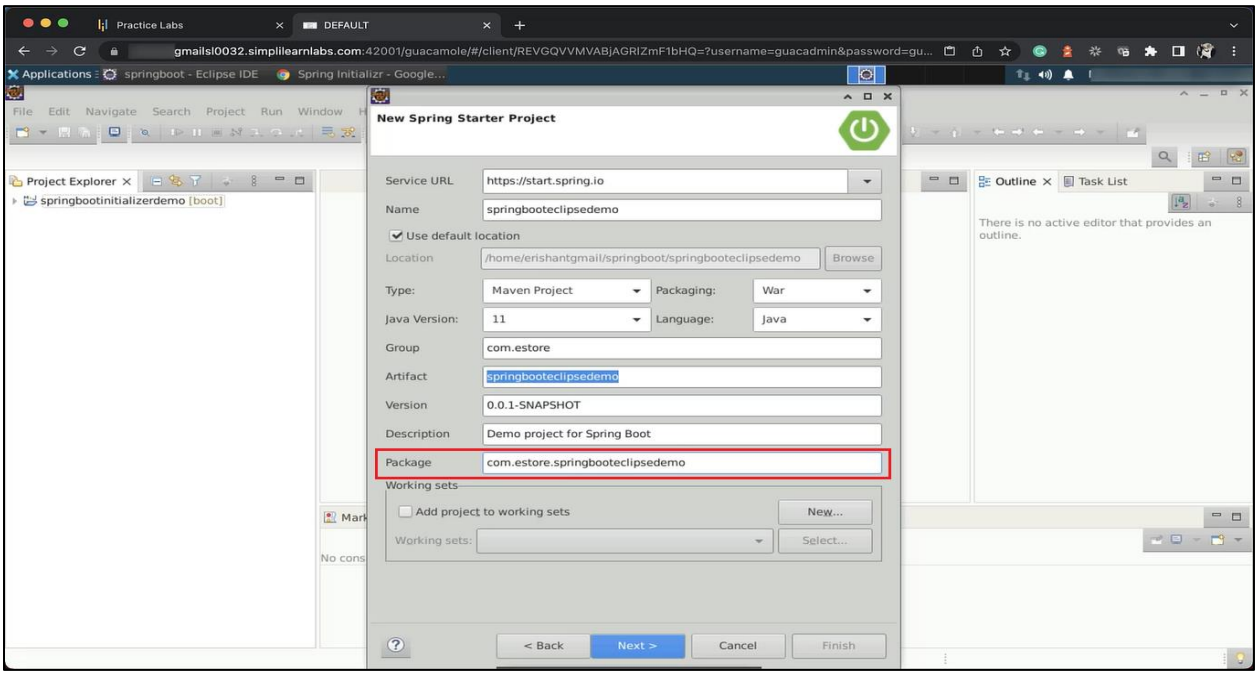
2.3 Select **Spring Starter Project** in the select wizard and then click on **Next**. This will create a new **Spring Starter Project** to specify all the required details regarding the project.



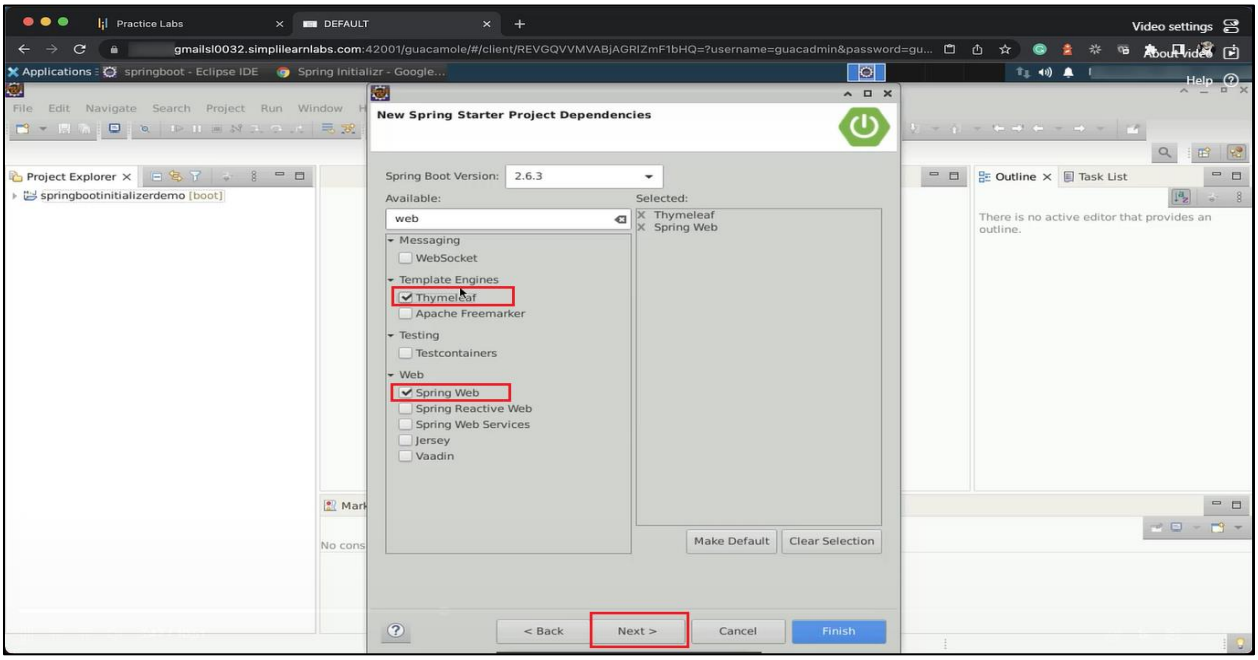
2.4 Now, provide a Spring Boot project name like **springbooteclipsedemo**. If it is a Maven project, the **Packaging** should be **War** type. Set the **Group** to **com.estore**. You'll see the **Artifact Id** for the Spring Boot project below.



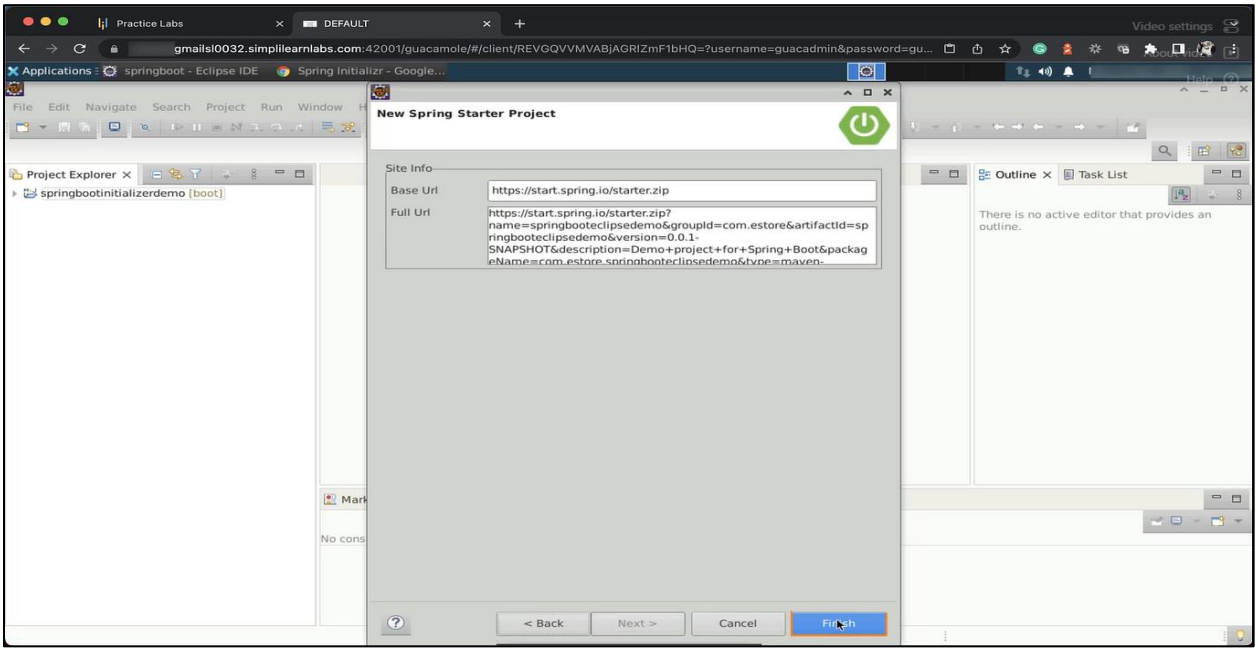
2.5 Now set the **Package Id** to **com.estore.springbooteclipsedemo** and click **Next**



2.6 Now add the dependencies by selecting the desired **Spring Boot Version**. Then search for the **Spring Web** dependency and select it. If web templates should be created, then select **Thymeleaf** and click **Next**



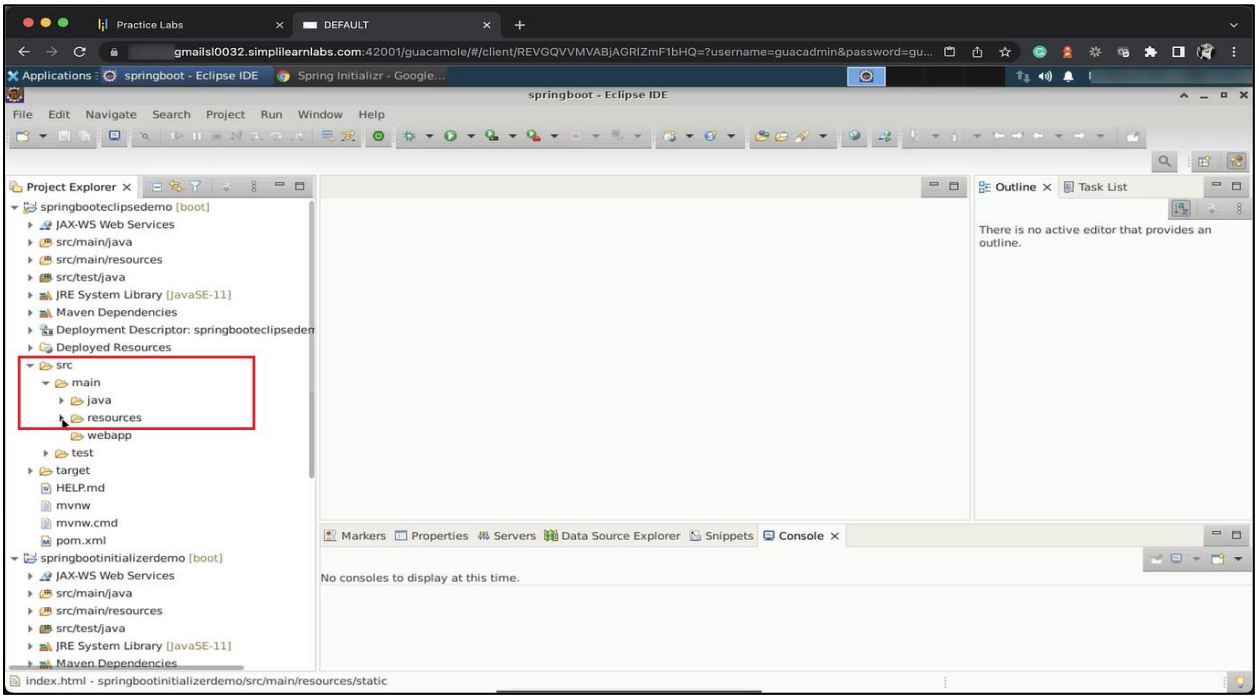
2.7 Next, you'll see important **Site info**, no changes are needed here. So, click **Finish**



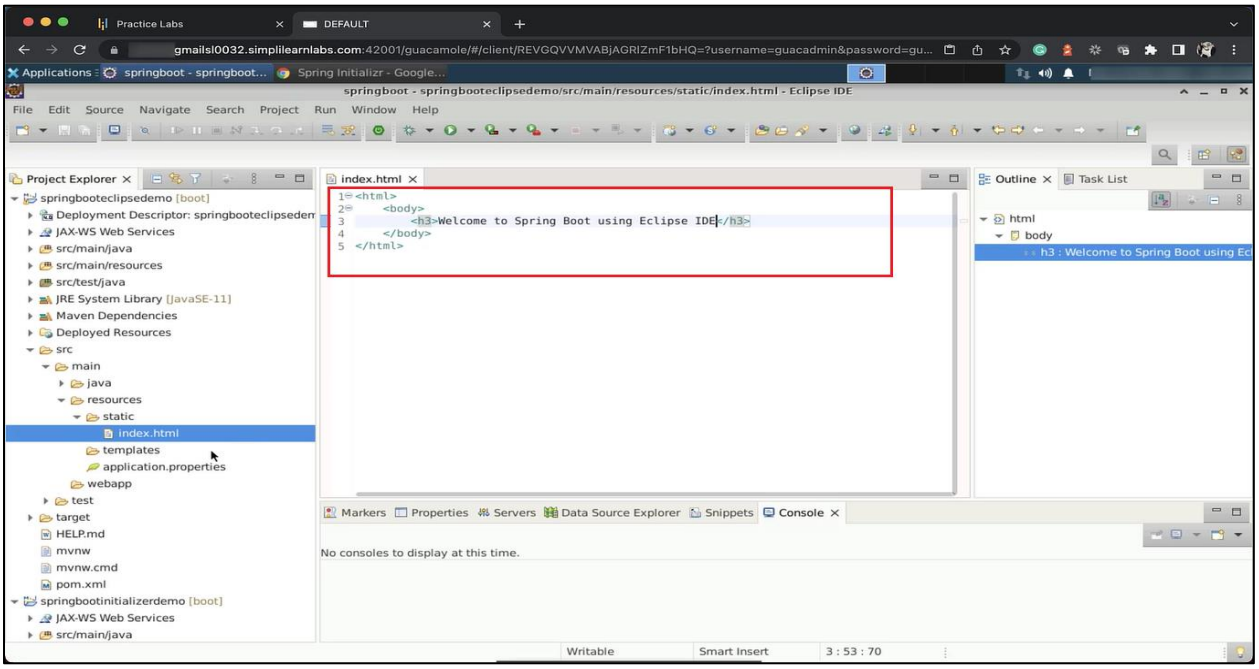
Now the Spring Boot project has been set up inside the Eclipse IDE without using the **Spring Initializr**.

Step 3: Initializing the Spring Boot Project

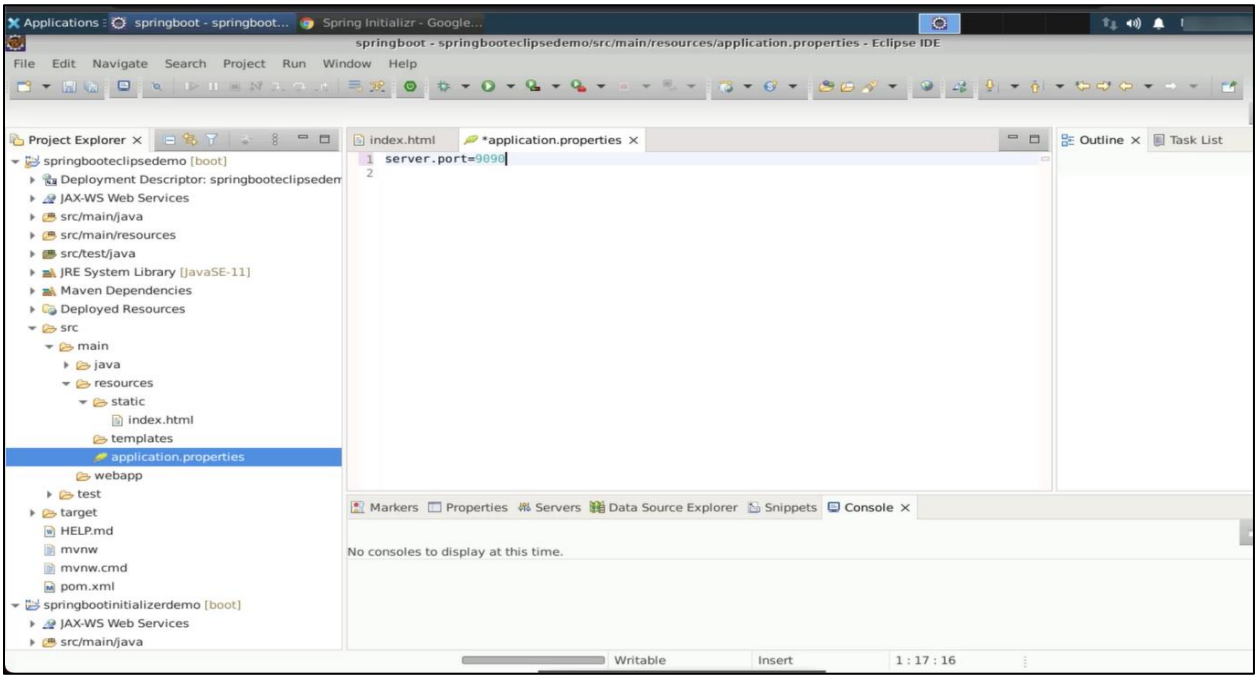
3.1 Now the complete Spring Boot project structure can be seen in the Project Explorer of the Eclipse IDE. Open the **resource** folder under **src > main** to edit the web interface of the Spring Boot project



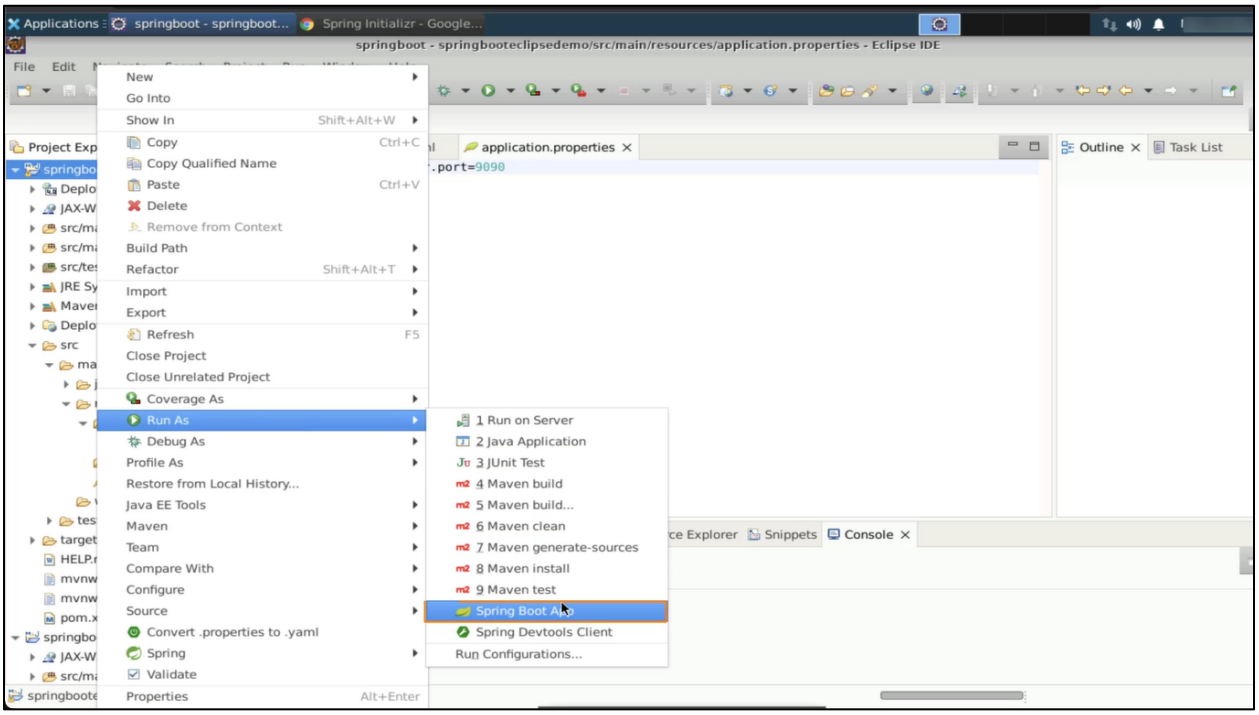
3.2 Now open the **index.html** file under the static folder and write the highlighted HTML code in the **index.html** file



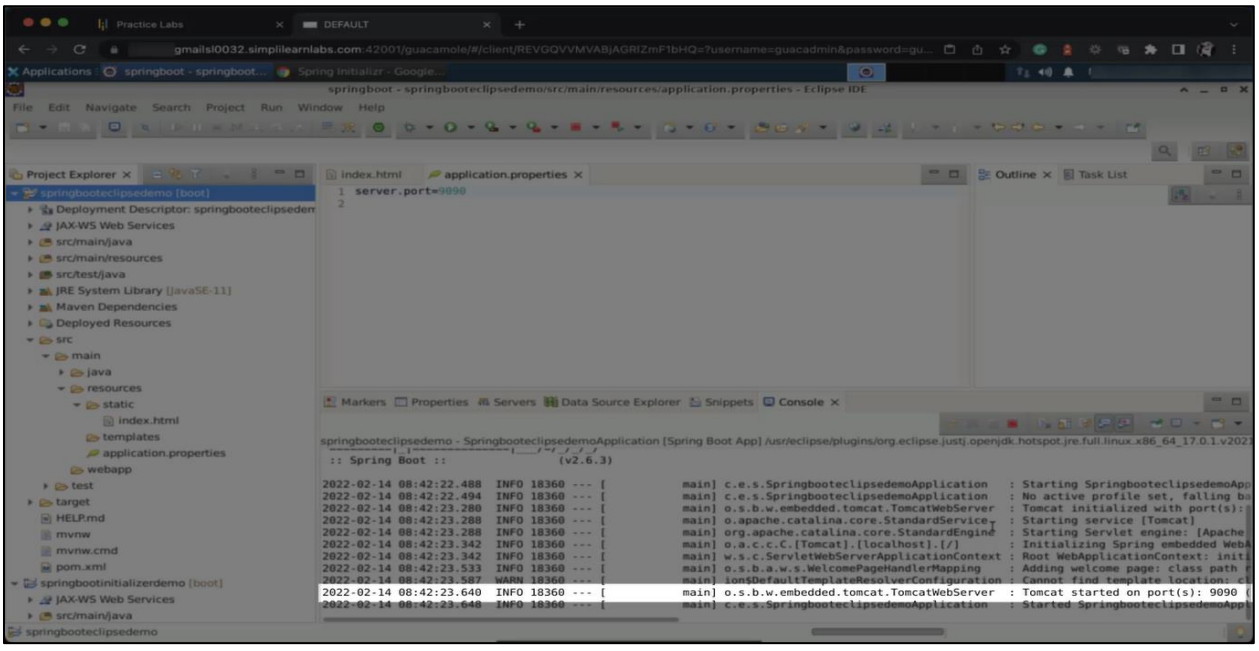
3.3 In the **application.properties,** specify the server port number as **9090**, then save this configuration. This port number will be used to host the Spring Boot project.



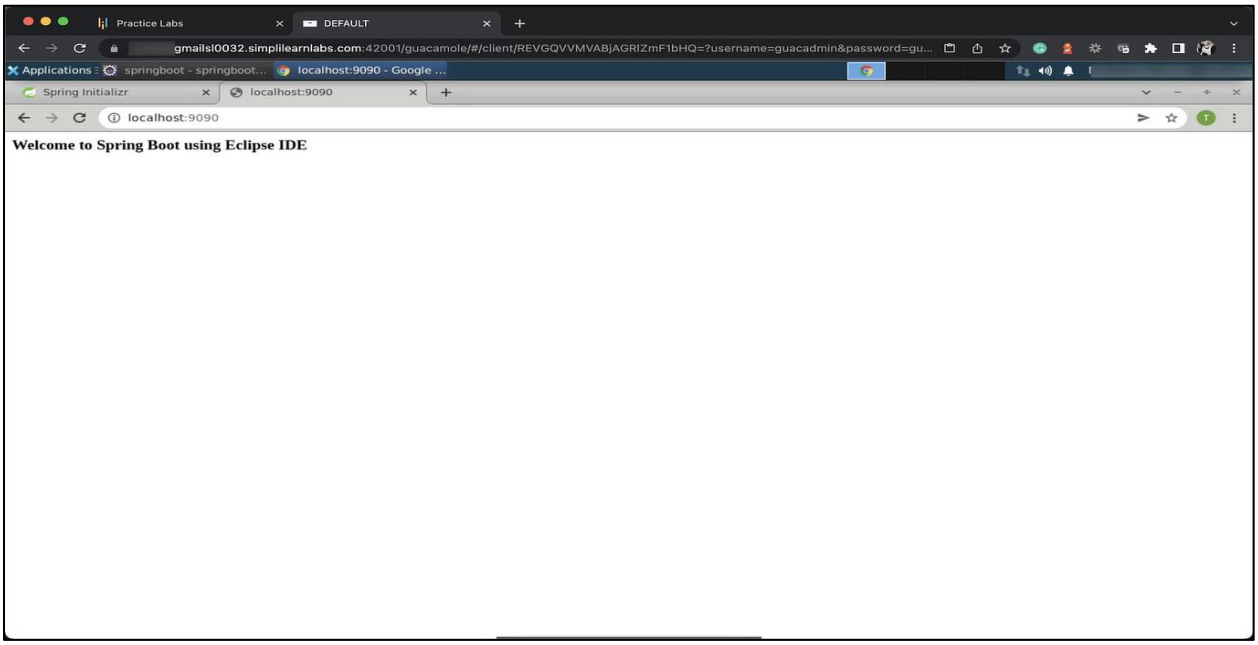
3.4 Now **right-click** on the project and select **Run as > Spring Boot Apps**



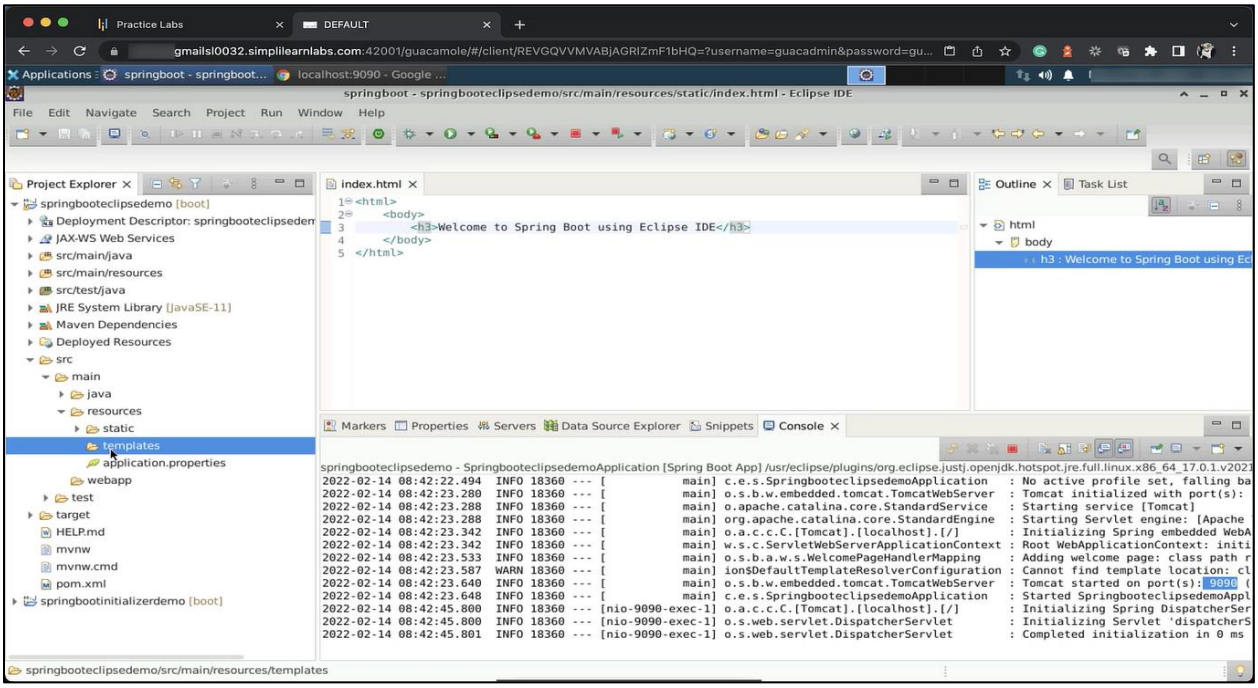
3.5 The project will begin to run as a **Spring Boot application**. Wait for the spring to finish. After that, you'll see the embedded Tomcat Server has been initialized and is running on the specified server, **9090**.



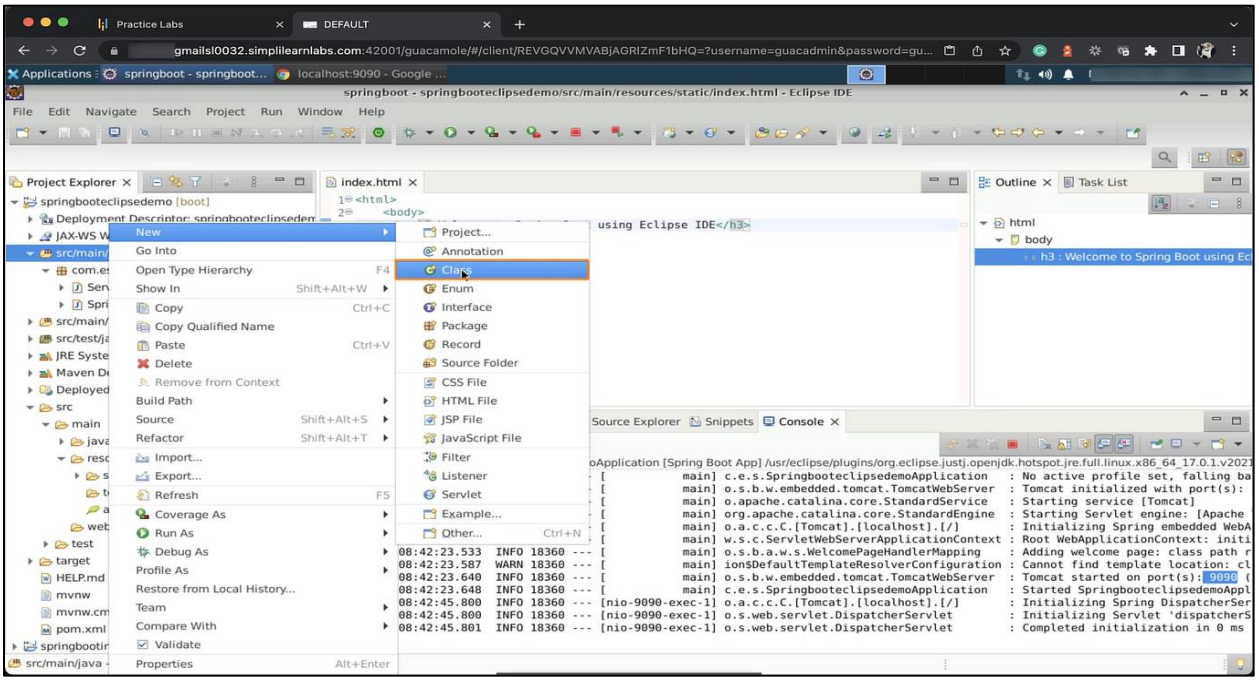
3.6 In the browser, open the **localhost:9090** to run the Spring Boot project



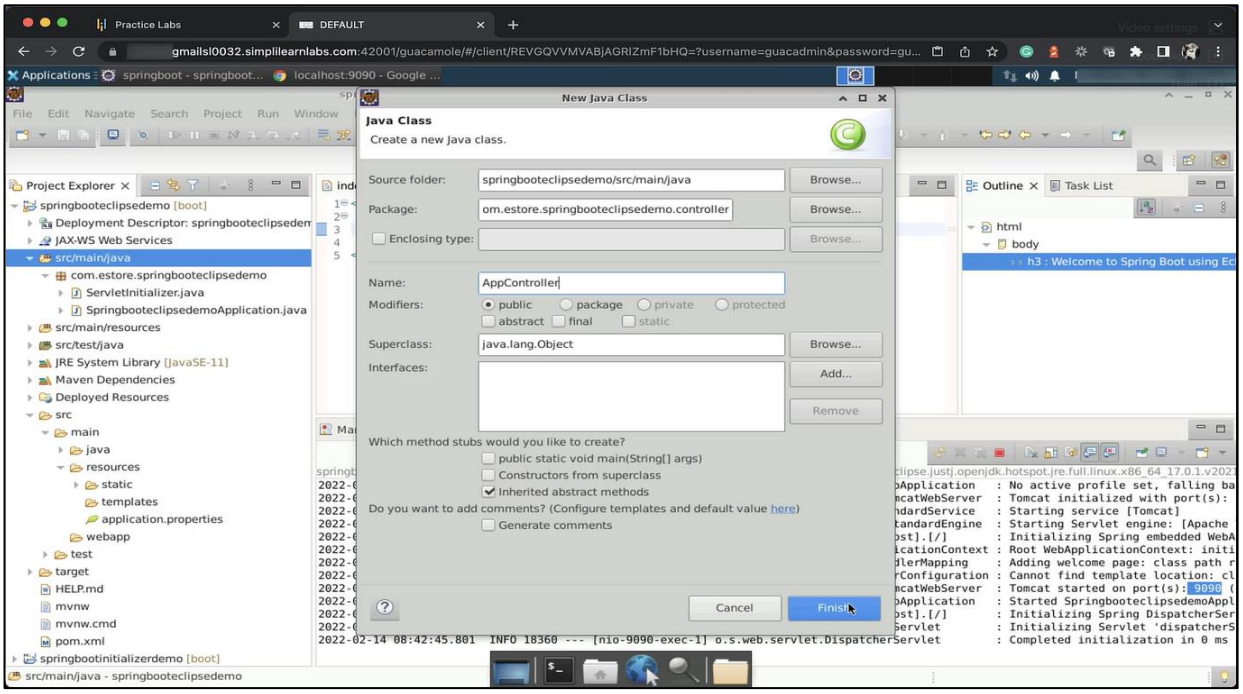
3.7 You can also create the required templates in the **template directory**. These templates can be rendered by the controllers in the web MVC structure.



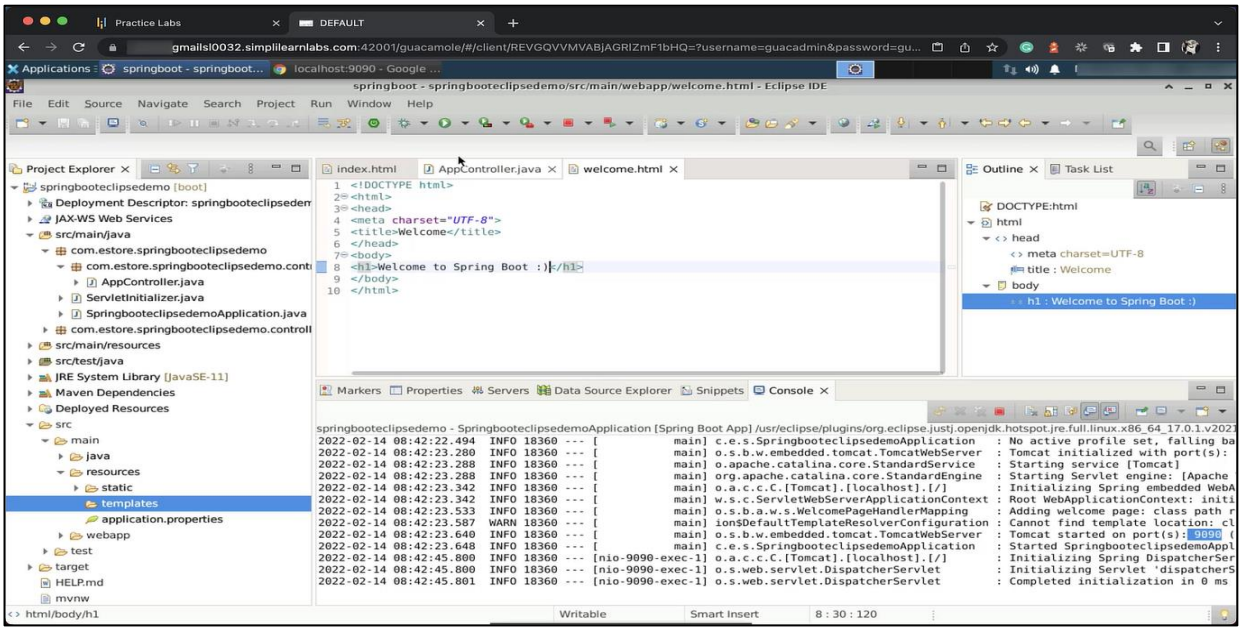
3.8 You can create controllers for your project, which will be created as a new class under the **src/main/java** directory



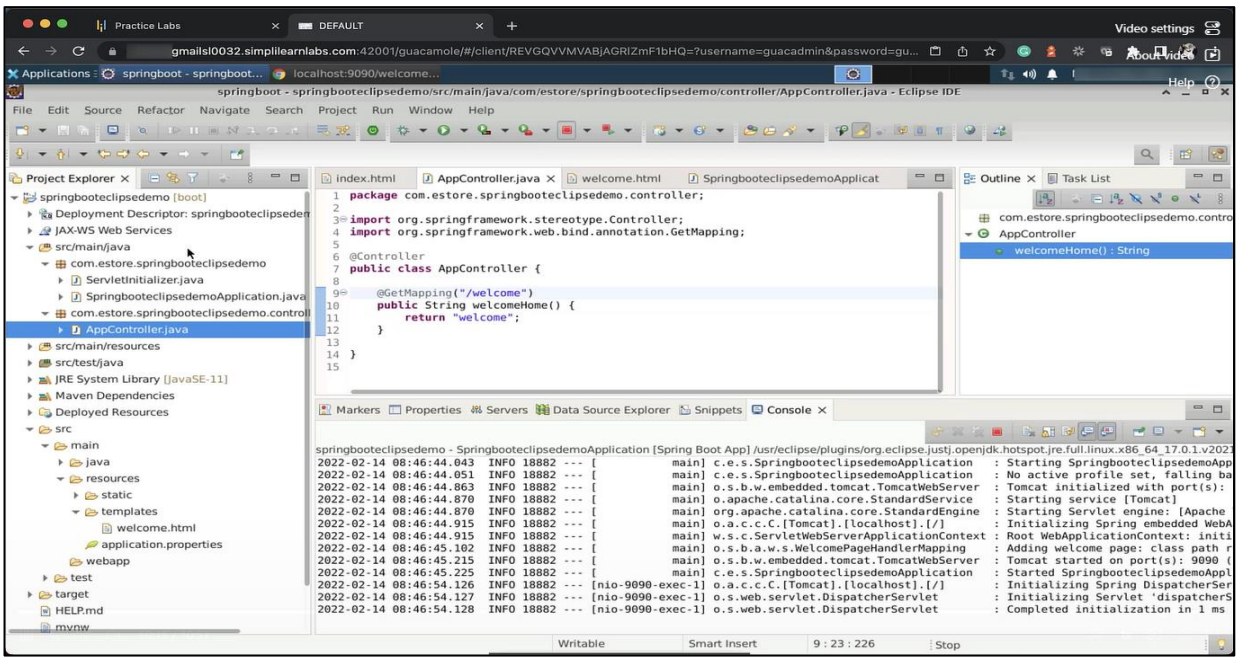
3.9 Set the appropriate **package** and **name for the controller** class. In this class, you can use the respective spring annotations.



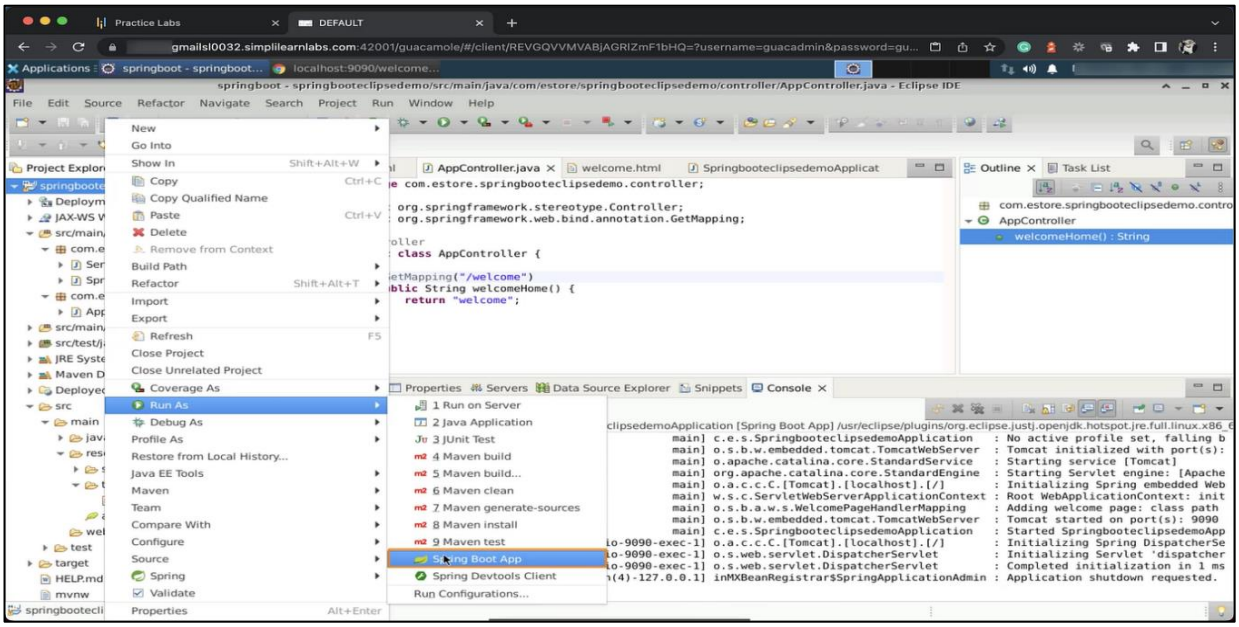
3.10 Create an HTML file by right-clicking the templates directory and selecting **New > HTML File**. Name the file **welcome.html**, then click on **Next** and **Finish**



3.11 In the **AppController.java** file, you can create a public method with a return type as **string WelcomeHome()** and this web method can return the template created in the step above



3.12 Rerun the project after the changes are made. For that, refresh **src/main/java**, **right-click** over the project, and select **Run As > Spring Boot App**



3.13 In the web browser, type **localhost:9090/welcome** and hit **Enter**. This will show the output of the Spring Boot project **springbooteclipsedemo**