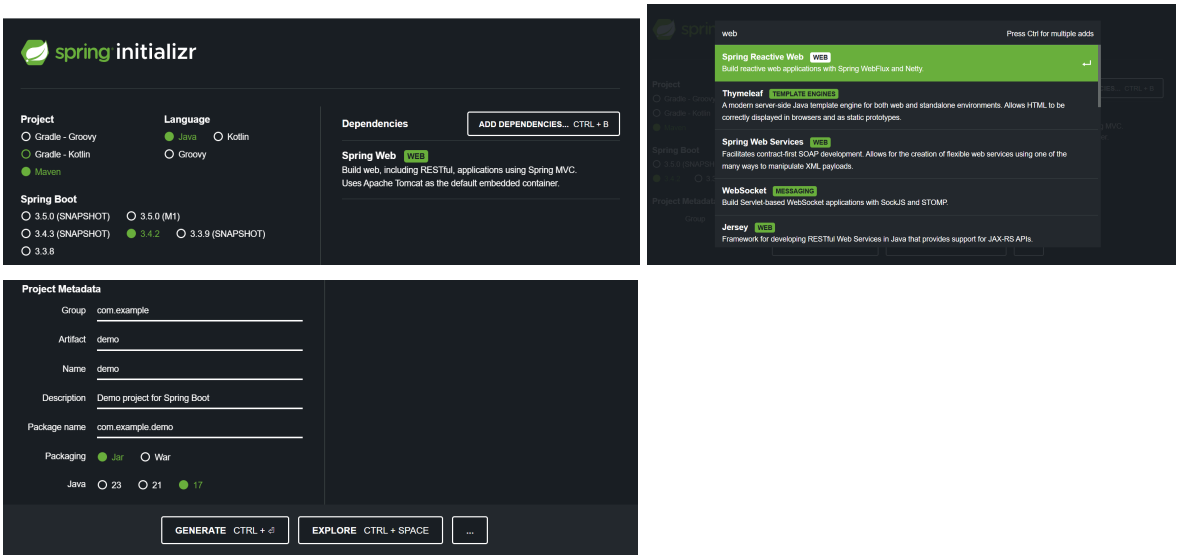
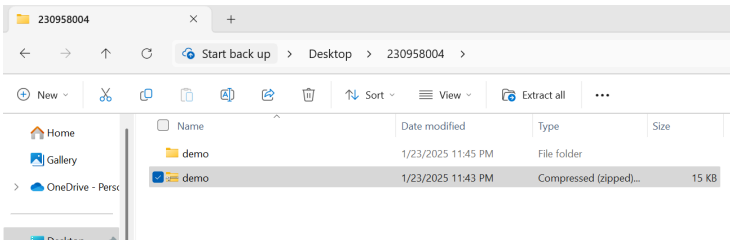
IFT LAB REPORT
LAB 01-23/01/2025

1. Setup the spring boot application and print the hello world message

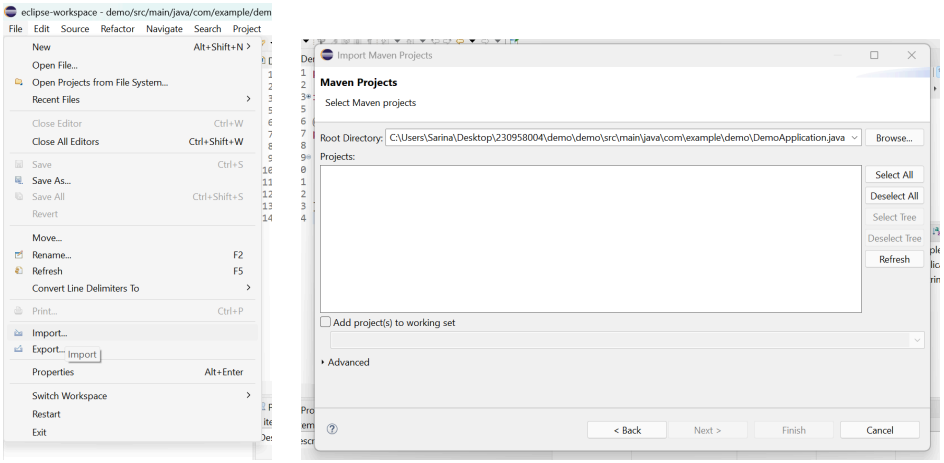
Step 1: Go to https://start.spring.io/
Step 2: Select (i)Project-'Maven'
(ii)Language-'Java'
(iii)Spring Boot-'3.4.2'
(iv)Dependencies-'Spring Web [WEB]'.
(v)Project metadata- name your file. Here, I used 'demo'.
(vi)Packaging-'Jar'
(vii)Java-'17'
Step 3: Click on Generate(downloads file in zip file format).



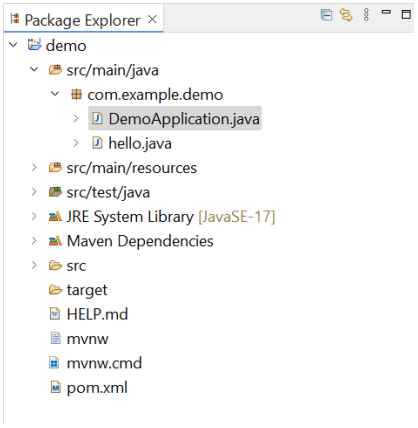
Step 4: Go to 'libraries' -> Go to 'downloads' -> Move the downloaded zip folder to a new file (we named the file our registration number) on the desktop -> In the new location of zip file, extract all the files from the zip file.



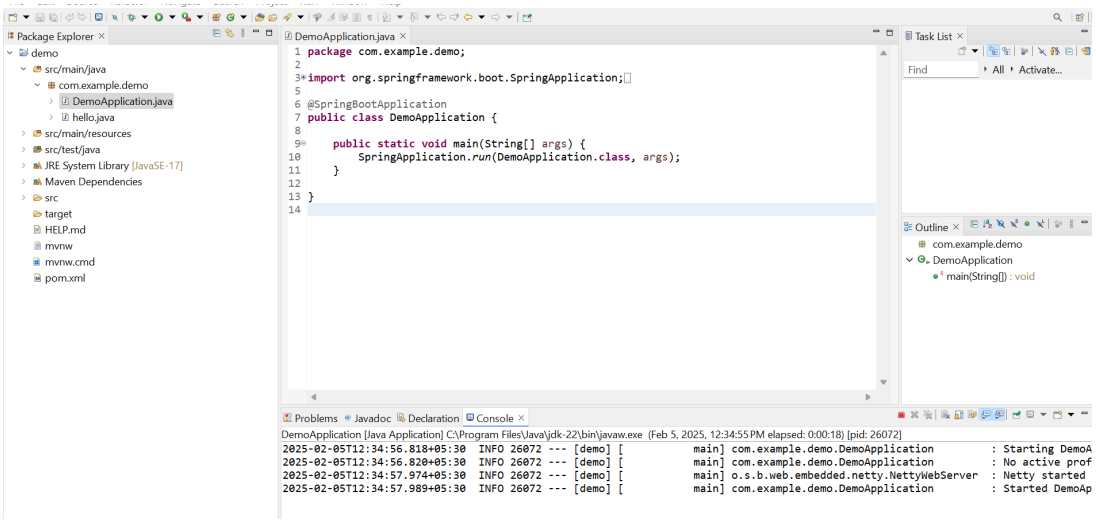
Step 5: Open Eclipse -> Go to File -> Go to Import.. -> Select existing maven projects -> Browse the file you extracted and select -> Click Finish to import the file.



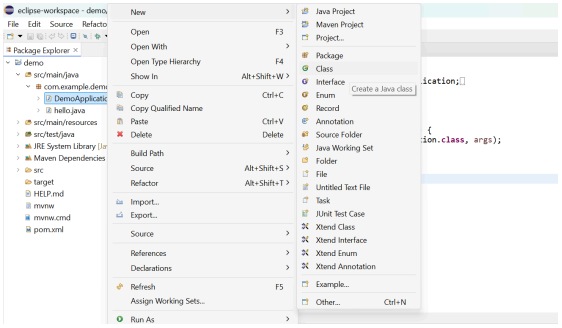
Step 6: Click ‘demo’ -> Select ‘scr/main/java’ -> Select ‘com.example.demo’ -> Select ‘DemoApplication.java’



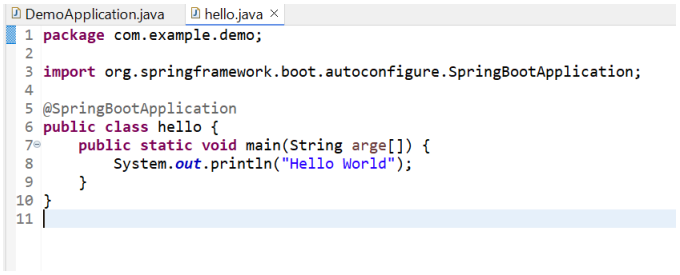
Step 7: Run ‘DemoApplication.java’



Step 8: Create a new class: Right-click on ‘com.example.demo’ -> Select ‘New’ -> Select ‘Class’ -> Name the new class ‘Hello’

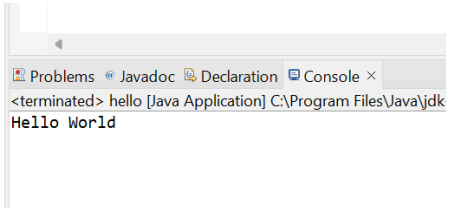


Step 9: Run the following code



```
DemoApplication.java  hello.java ×
1  package com.example.demo;
2
3  import org.springframework.boot.autoconfigure.SpringBootApplication;
4
5  @SpringBootApplication
6  public class hello {
7      public static void main(String args[]) {
8          System.out.println("Hello World");
9      }
10 }
11
```

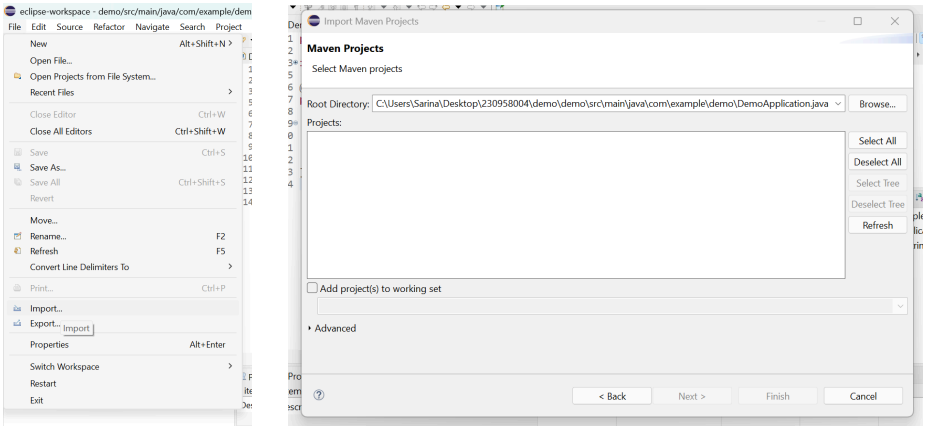
You will get the following result in the console.



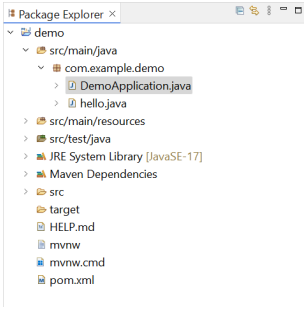
```
Problems  Javadoc  Declaration  Console ×
<terminated> hello [Java Application] C:\Program Files\Java\jdk
Hello World
```

2. Setup the spring boot application with hello world message in separate package named controller

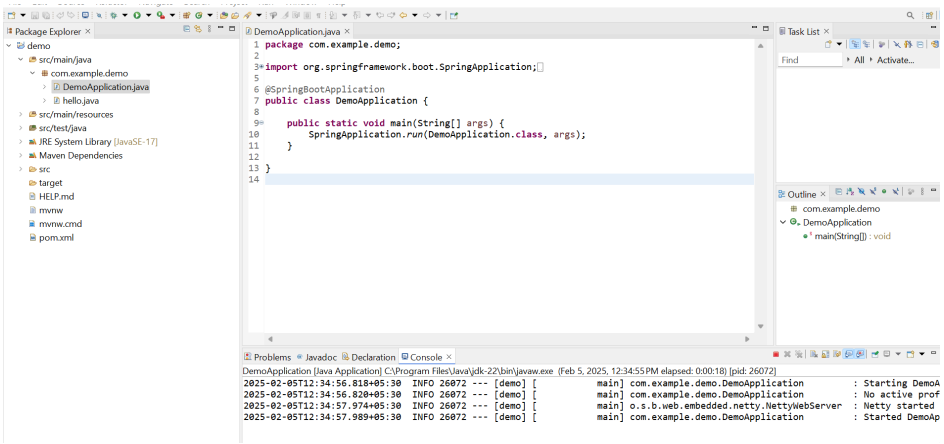
Step 1: Go to https://start.spring.io/
Step 2: Select (i)Project-'Maven'
(ii)Language-'Java'
(iii)Spring Boot-'3.4.2'
(iv)Dependencies-'Spring Web [WEB]'.
(v)Project metadata- name your file. Here, I used 'demo'.
(vi)Packaging-'Jar'
(vii)Java-'17'
Step 3: Click on Generate(downloads file in zip file format).
Step 4: Go to libraries -> Go to downloads -> Move the downloaded zip folder to a new file (we named the file our registration number) on the desktop -> In the new location of zip file, extract all the files from the zip file.
Step 5: Open Eclipse -> Go to File -> Go to Import.. -> Select existing maven projects -> Browse the file you extracted and select -> Click Finish to import the file.
Step 6: Click 'demo' -> Select 'scr/main/java' -> Select 'com.example.demo' -> Select 'DemoApplication.java'



Step 6: Click ‘demo’ -> Select ‘scr/main/java’ -> Select ‘com.example.demo’ -> Select ‘DemoApplication.java’

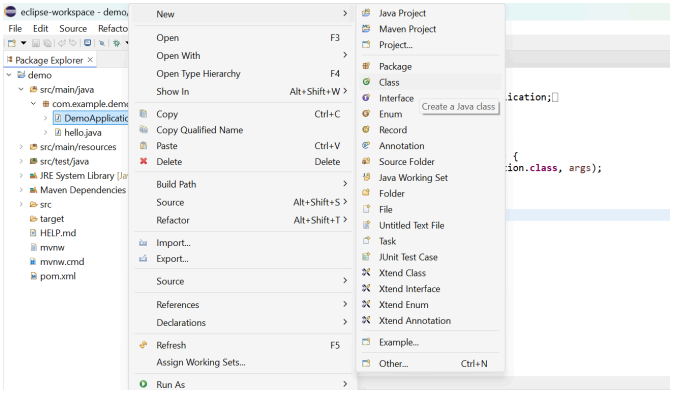


Step 7: Run ‘DemoApplication.java’

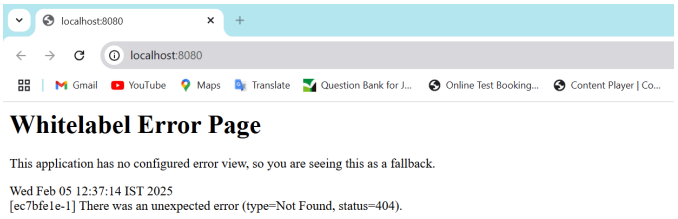


Step 8: Create a new class: Right-click on ‘com.example.demo’ -> Select ‘New’ -> Select ‘Package’ -> Name the new package ‘com.example.demo.controller’

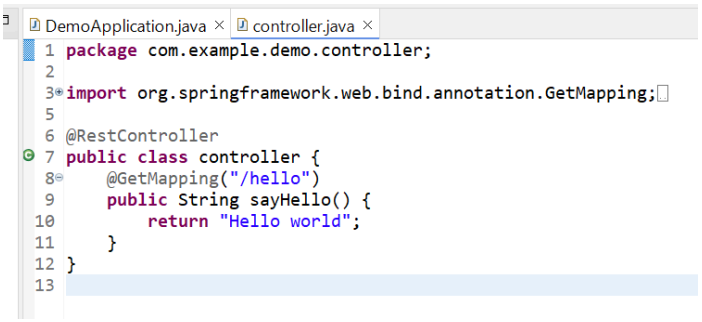
Step 9: Create a new class: Right-click on ‘com.example.demo.controller’ -> Select ‘New’ -> Select ‘Class’ -> Name the new class ‘Controller’



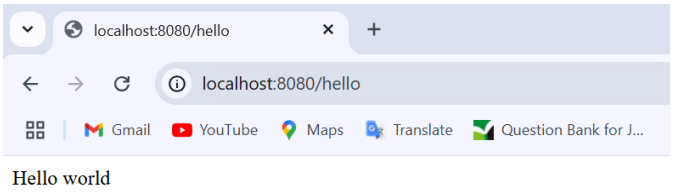
Step 10: Open Google Chrome -> Go to http://localhost:8080/ -> It will display 'White Lable Error'



Step 11: write the following code in controller class



Step 12: Run controller class.
Step 13: Go to http://localhost:8080/hello in Google Chrome.The page should display the message 'Hello World'



3. Test the above application in POSTMAN

Step 1: Open Eclipse -> run your controller and application class.
Step 2: Open the Postman app.
Step 3: Select GET -> enter http://localhost:8080/hello -> Click 'Send'

If your code runs correctly then you should get the response as ‘Hello world’(same as in the webpage)