

# **PROJECT PRESENTATION**



**SRI LAKSHMI TULASI II FUNCTIONAL TEST ENGINEER**

# TOOLS

**01**

ECLIPSE

**02**

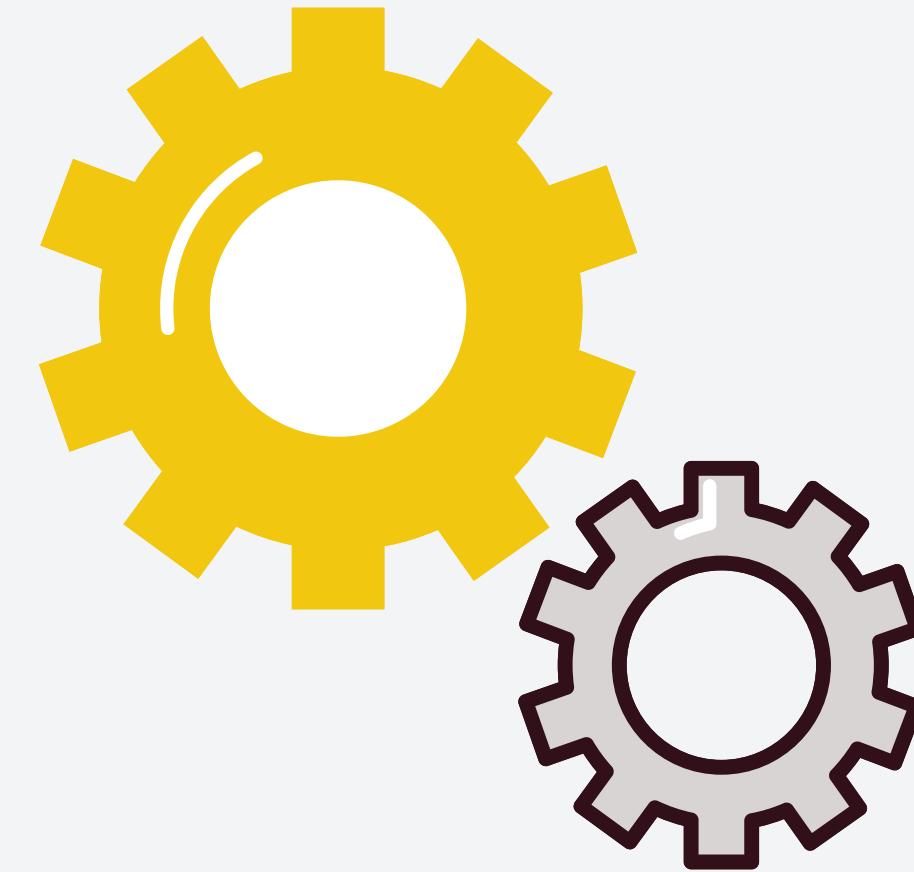
JENKINS

**03**

GITHUB

**04**

SELENIUM





spring  
Boot

# STEP1: Create a spring boot project>>maven>>java>>add dependencies

The screenshot shows the Spring Initializr website at <https://start.spring.io>. The page is titled "spring initializr". It has sections for "Project", "Language", "Dependencies", and "Spring Boot". Under "Project", "Maven" is selected. Under "Language", "Java" is selected. Under "Dependencies", "Spring Web" is selected. Under "Spring Boot", "3.1.1" is selected. At the bottom, there are "GENERATE" and "EXPLORE" buttons.

Meet the Spring team this August at SpringOne.

# spring initializr

Project

- Gradle - Groovy
- Gradle - Kotlin
- Maven

Language

- Java
- Kotlin
- Groovy

Dependencies

**ADD DEPENDENCIES... CTRL + B**

**Spring Web WEB**

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Boot

- 3.2.0 (SNAPSHOT)
- 3.1.2 (SNAPSHOT)
- 3.1.1
- 3.0.9 (SNAPSHOT)
- 3.0.8
- 2.7.14 (SNAPSHOT)
- 2.7.13

Project Metadata

STRATEGY N°1 STRATEGY N°2 STRATEGY N°3

GENERATE CTRL + ⌘ EXPLORE CTRL + SPACE SHARE...

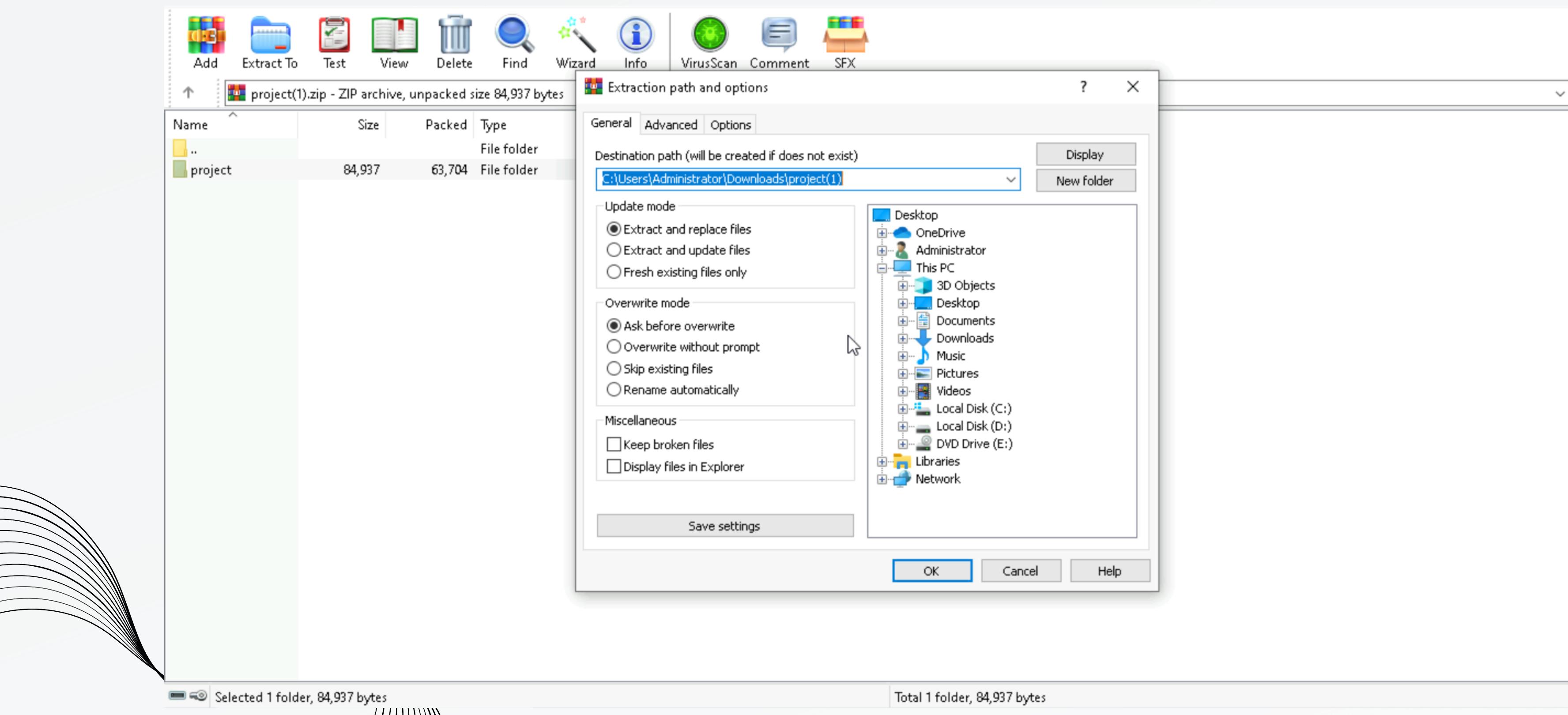
# STEP2: Group>>artifact>>description>>package names>>generate

The screenshot shows the configuration page for a new project on the Start Bootstrap website. The top navigation bar includes links for Home, About, Services, Portfolio, Contact, and Log In. A sidebar on the left features a navigation menu with Home, Services, Portfolio, and Contact, and a search bar. The main content area displays the following project metadata:

- Group: firstpro
- Artifact: project
- Name: project
- Description: spring boot project
- Package name: firstpro.project
- Packaging:  Jar  War
- Java:  20  17  11  8

At the bottom, there are three buttons: "GENERATE" (CTRL + ⌘), "EXPLORE" (CTRL + SPACE), and "SHARE...".

# STEP3: Extract file to the resspective folder



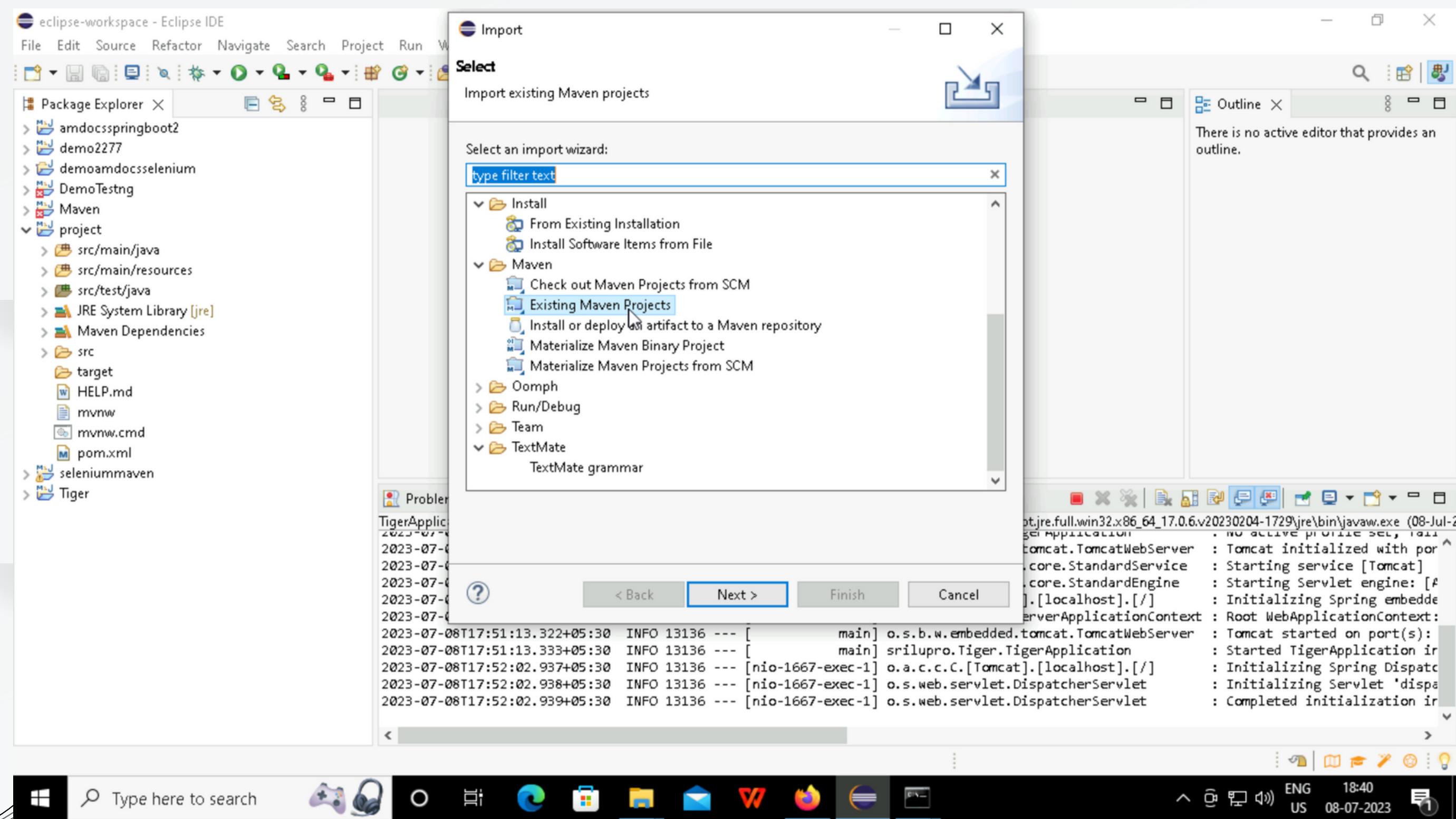


eclipseIDE

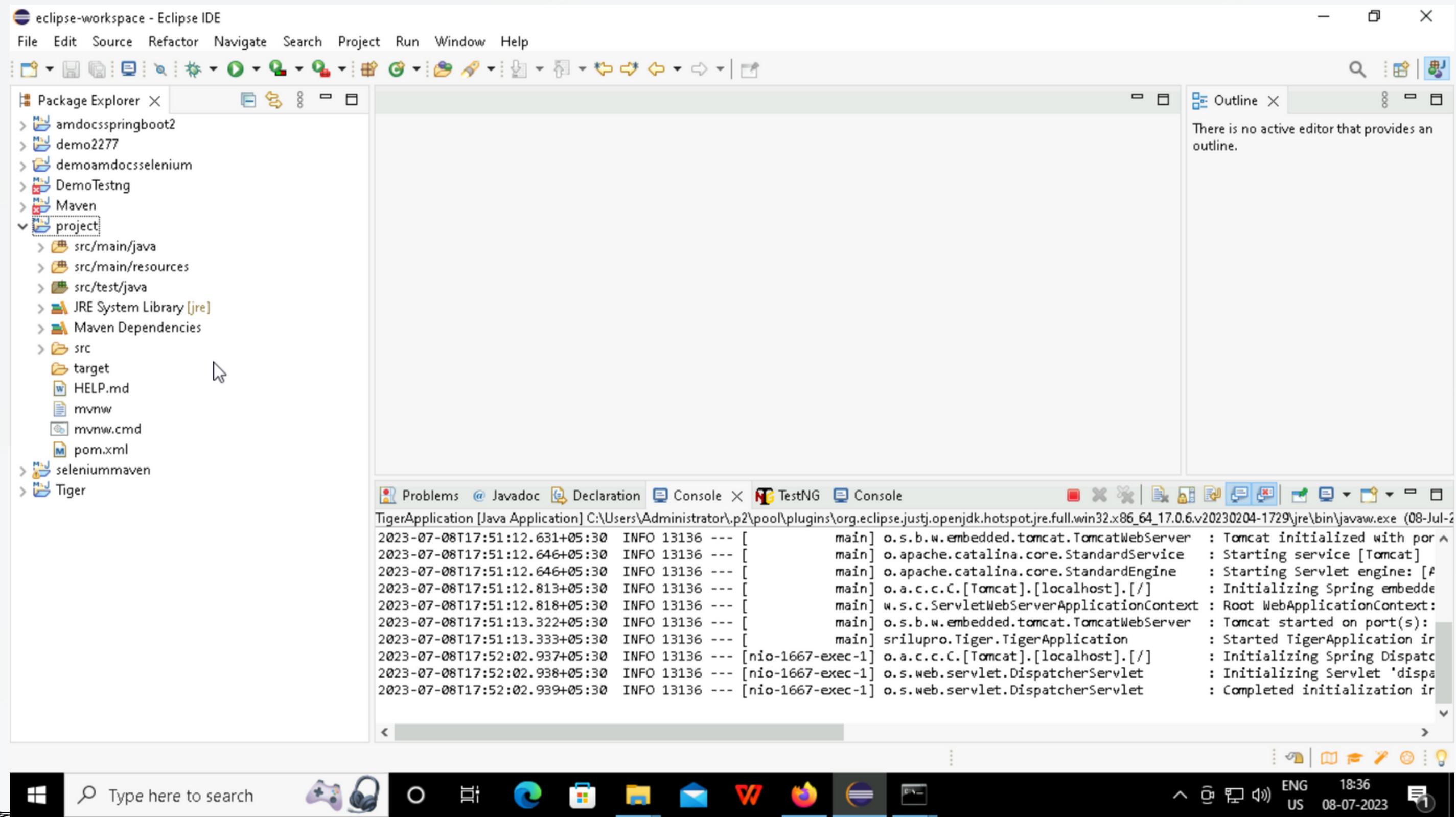
2022-09

Starting Eclipse IDE

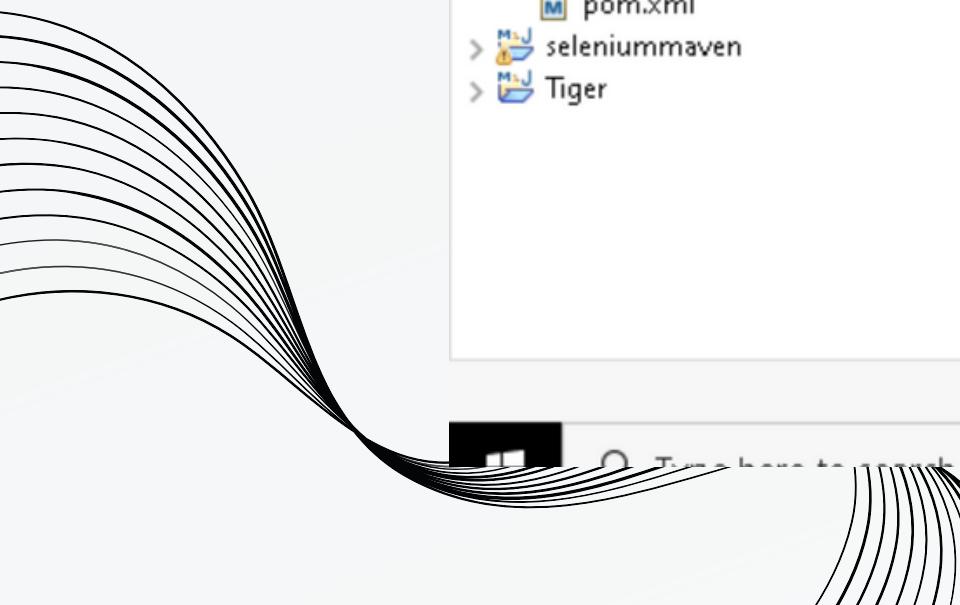
# STEP4: Open eclipse and import your folderb to the work space



# STEP5: Create a new class in src/main/java



# STEP 6: write your code and save it



A screenshot of an IDE interface showing a Java project structure and code editor.

**File Structure:**

- Project Explorer (left):
  - amdocspringboot2
  - demo2277
  - demoamdocselenium
  - DemoTestng
  - Maven
  - project
    - src/main/java
      - Firstpro.project
        - HandleCheckbox.java
        - ProjectApplication.java
        - seleniumproject1.java
        - Welcome.java
    - src/main/resources
    - src/test/java
    - JRE System Library [jre]
    - Maven Dependencies
    - Referenced Libraries
    - src
      - target
      - HELP.md
      - mvnw
      - mvnw.cmd
      - pom.xml
    - seleniummaven
    - Tiger

**Code Editor (center):**

```
1 package Firstpro.project;
2
3 import org.springframework.web.bind.annotation.GetMapping;
4
5 @RestController
6 public class Welcome {
7
8     @GetMapping("/get")
9     public String getData() {
10         return "welcome to my presentation";
11     }
12 }
13
14 }
```

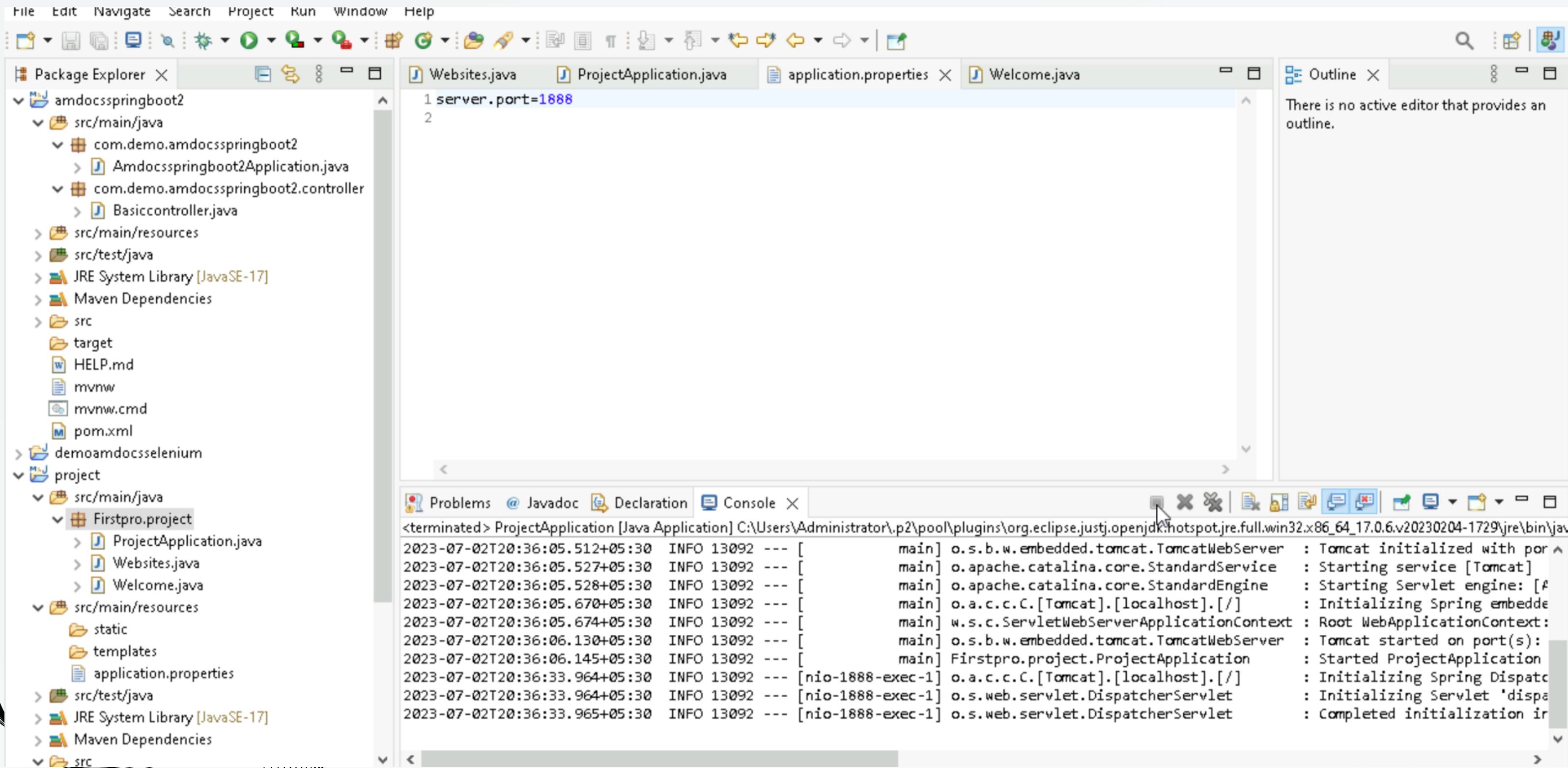
**Outline View (right):**

- Firstpro.project
  - Welcome
  - getData(): String

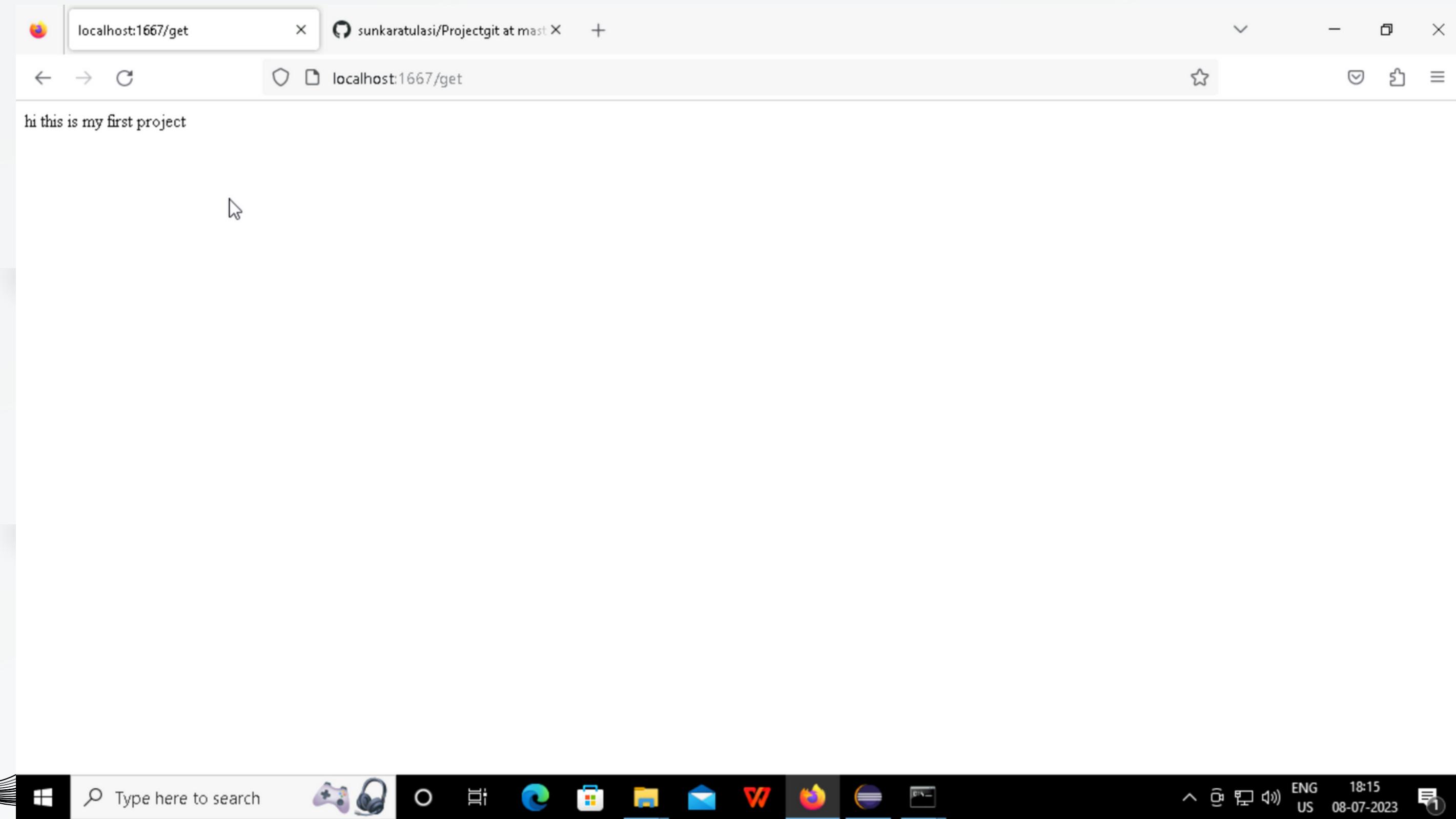
**Bottom Status Bar:**

- Writable
- Smart Insert
- 14 : 2 : 296
- ENG 17:21

# STEP7: Open application.properties in src/main/resources and give your port number



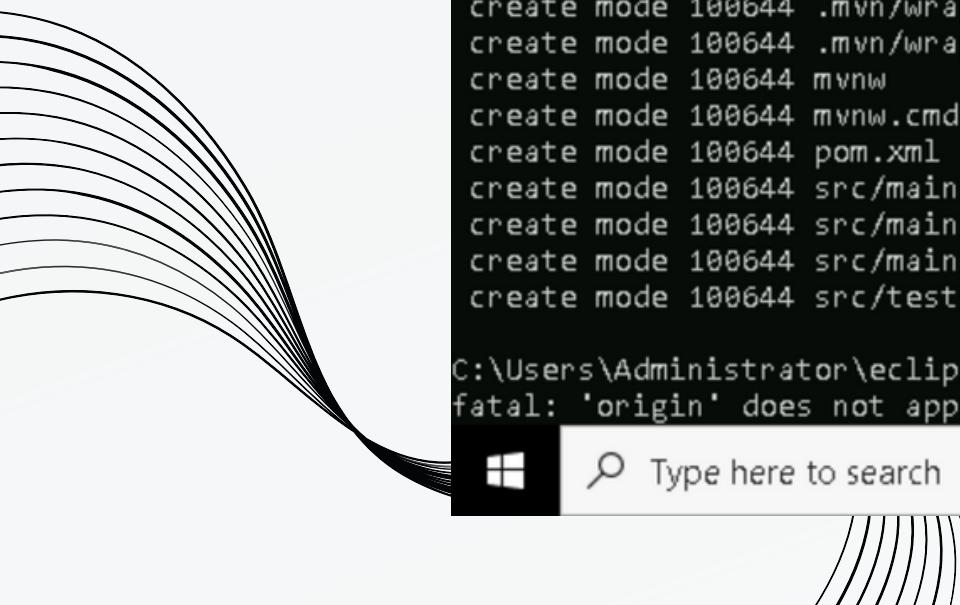
# STEP 8: Go to browser and enter your port number to see your output





**GitHub**

# STEP8: Open command prompt in your folder and push the files using git commands



```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\eclipse-workspace\Project>git init
Reinitialized existing Git repository in C:/Users/Administrator/eclipse-workspace/Project/.git/

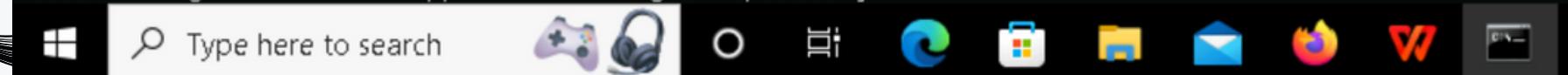
C:\Users\Administrator\eclipse-workspace\Project>git status
On branch master

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
  new file:  .gitignore
  new file:  .mvn/wrapper/maven-wrapper.jar
  new file:  .mvn/wrapper/maven-wrapper.properties
  new file:  mvnw
  new file:  mvnw.cmd
  new file:  pom.xml
  new file:  src/main/java/AQE/Project/ProjectApplication.java
  new file:  src/main/java/AQE/Project/controller/Basiccontroller.java
  new file:  src/main/resources/application.properties
  new file:  src/test/java/AQE/Project/ProjectApplicationTests.java

C:\Users\Administrator\eclipse-workspace\Project>git commit -m "first commit"
[master (root-commit) 1eaefc1] first commit
 10 files changed, 630 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 .mvn/wrapper/maven-wrapper.jar
 create mode 100644 .mvn/wrapper/maven-wrapper.properties
 create mode 100644 mvnw
 create mode 100644 mvnw.cmd
 create mode 100644 pom.xml
 create mode 100644 src/main/java/AQE/Project/ProjectApplication.java
 create mode 100644 src/main/java/AQE/Project/controller/Basiccontroller.java
 create mode 100644 src/main/resources/application.properties
 create mode 100644 src/test/java/AQE/Project/ProjectApplicationTests.java

C:\Users\Administrator\eclipse-workspace\Project>git push -u origin master
fatal: 'origin' does not appear to be a git repository
```



ENG 16:01  
US 08-07-2023

# STEP9: New>>repository>>public

The screenshot shows a browser window with three tabs: 'localhost:1667/get', 'New repository · GitHub', and 'Spring Initializr'. The main content is the 'Create a new repository' page on GitHub. The 'Repository name' field contains 'gitprojects', which is highlighted with a blue border and has a green checkmark below it indicating it's available. The 'Owner' dropdown is set to 'sunkaratulasi'. Below the form, there's a note about repository names being short and memorable, with a suggestion for 'congenial-octo-succotash'. The 'Description (optional)' field is empty. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. A note under 'Public' says 'Anyone on the internet can see this repository. You choose who can commit.' A note under 'Private' says 'You choose who can see and commit to this repository.' At the very bottom, there's a taskbar with icons for File Explorer, Mail, and other applications, along with system status indicators like battery level and network connection.

localhost:1667/get    New repository · GitHub    Spring Initializr

https://github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

Owner \*    Repository name \*

sunkaratulasi / gitprojects

gitprojects is available.

Great repository names are short and memorable. Need inspiration? How about [congenial-octo-succotash](#)?

Description (optional)

Public    Anyone on the internet can see this repository. You choose who can commit.

Private    You choose who can see and commit to this repository.

Initialize this repository with:

Type here to search

ENG US 21:04 08-07-2023

# STEP10: Create repository

The screenshot shows a Firefox browser window with three tabs open:

- localhost:1667/get
- New repository · GitHub
- Spring Initializr

The main content area is titled "New repository". It includes the following fields and information:

- Privacy:** Private (radio button selected). Description: You choose who can see and commit to this repository.
- Initialize this repository with:**
  - Add a README file. Description: This is where you can write a long description for your project. [Learn more about READMEs](#).
- Add .gitignore**
  - .gitignore template: None (dropdown menu)
  - Description: Choose which files not to track from a list of templates. [Learn more about ignoring files](#).
- Choose a license**
  - License: None (dropdown menu)
  - Description: A license tells others what they can and can't do with your code. [Learn more about licenses](#).
- Information:** ⓘ You are creating a public repository in your personal account.
- Create repository** (Green button)

At the bottom of the browser window, the address bar shows <https://github.com/new>. The Windows taskbar at the bottom includes icons for File Explorer, Task View, Edge, File Manager, Mail, and a search bar.

# STEP11: Check whether the files are pushed or not>>git

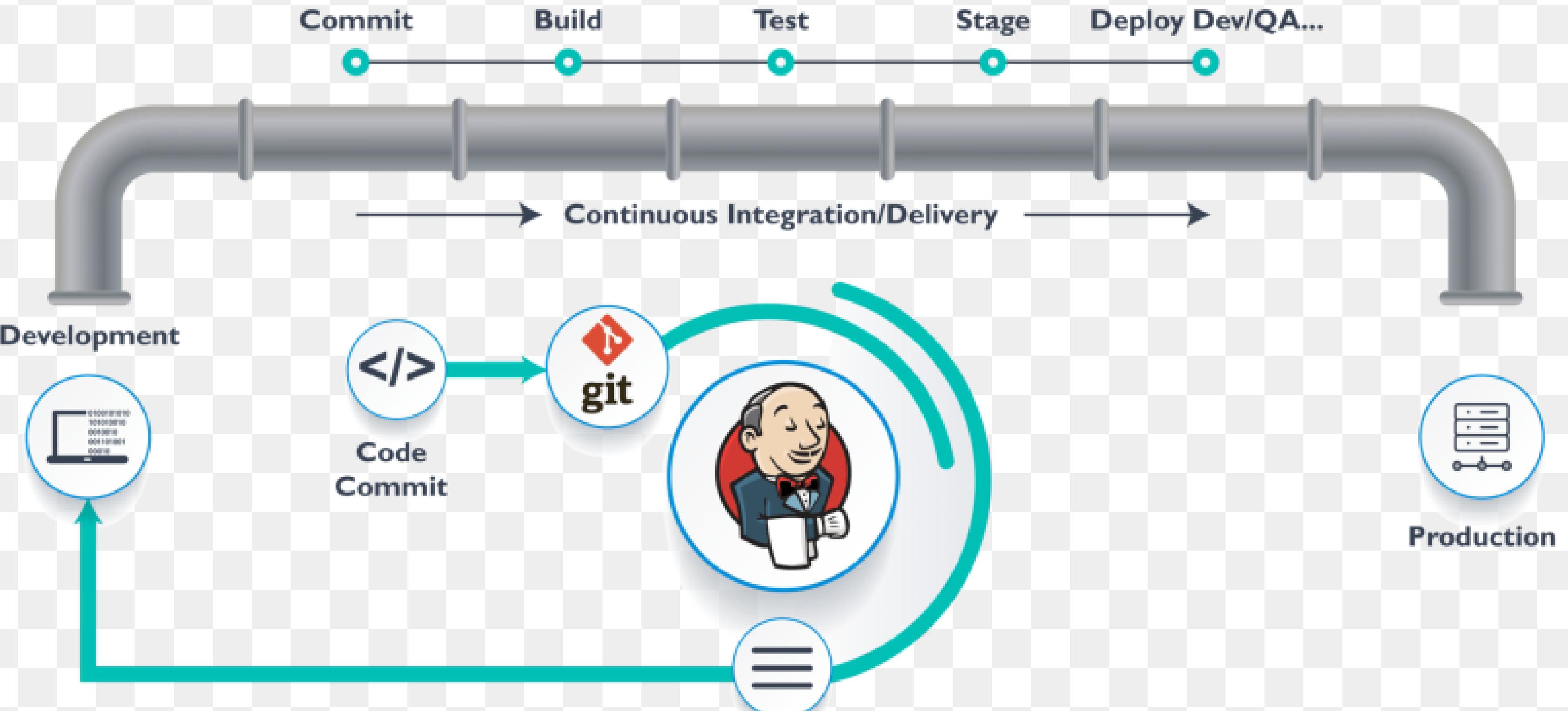
The screenshot shows a web browser window with three tabs open:

- localhost:1667/get
- sunkaratulasi/Projectgit at master
- Spring Initializr

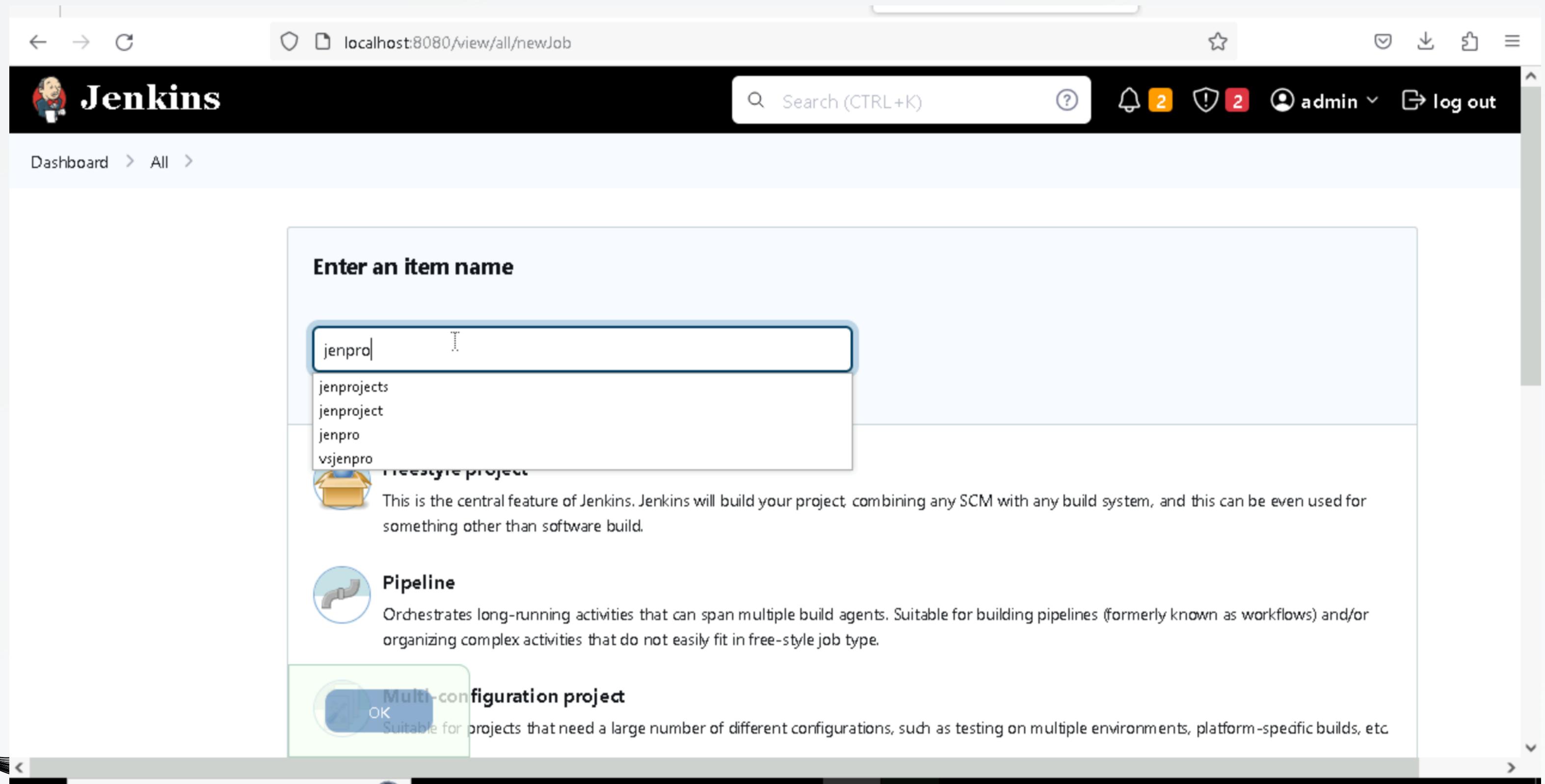
The main content area displays the GitHub repository page for 'Projectgit'. The repository is public and has 2 branches (master and main). The 'Code' tab is selected. A message indicates that the 'master' branch is 1 commit ahead and 1 commit behind the 'main' branch. The commit history shows the following commits:

Author	Commit Message	Date	Commits
sunkaratulasi	first commit	8dcf6c4 last week	1 commit
	.mvnw/wrapper	first commit	last week
	src	first commit	last week
	.gitignore	first commit	last week
	mvnw	first commit	last week
	mvnw.cmd	first commit	last week
	pom.xml	first commit	last week

The right sidebar contains sections for 'About', 'Activity', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Activity' section shows 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates 'No releases published' and 'Create a new release'. The 'Packages' section is currently empty.



# STEP12: Jenkin>>New item>>name>>ok



The screenshot shows a web browser window for Jenkins at the URL `localhost:8080/view/all/newJob`. The Jenkins logo is in the top left, and the top navigation bar includes a search bar, notifications (2), and a user account for 'admin'. Below the header, the breadcrumb navigation shows 'Dashboard > All >'. The main content area has a title 'Enter an item name' and a text input field containing 'jenpro'. A dropdown menu below the input field lists several items: 'jenprojects', 'jenproject', 'jenpro', and 'vsjenpro'. To the right of the input field, there is a brief description of a 'Free-style project': 'This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.' Below this, there is a section titled 'Pipeline' with a description: 'Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.' At the bottom, there is a 'Multi-configuration project' section with a large green 'OK' button and a note: 'Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.'

# STEP13: Git>>url>>credentials

The screenshot shows a browser window with three tabs: 'localhost:1667/get', 'sunkaratulasi/Projectgit at mast X', and 'presentationjenkin Config [Jenk X]'. The active tab is 'presentationjenkin Config [Jenk]'. The URL in the address bar is 'localhost:8080/job/presentationjenkin/configure'. The page title is 'Source Code Management' under 'Dashboard > presentationjenkin > Configuration'.

The main content is titled 'Configure' and shows the 'Source Code Management' section selected. Under 'Source Code Management', the 'Git' option is chosen. The 'Repositories' section contains a single repository configuration:

- Repository URL:** https://github.com/sunkaratulasi/Projectgit.git
- Credentials:** sunkaratulasi/\*\*\*\*\*\*\*\*\*
- Buttons:** Add, Advanced..., Add Repository, Save, Apply

The bottom of the screen shows the Windows taskbar with various icons and the system tray indicating the date and time as 08-07-2023 at 17:46.

# STEP14: Select poll SCM

The screenshot shows a browser window with three tabs: 'localhost:1667/get', 'sunkaratulasi/Projectgit at mast X', and 'presentationjenkin Config [Jenk X]'. The active tab is 'presentationjenkin Config [Jenk X]'.

The URL in the address bar is 'localhost:8080/job/presentationjenkin/configure'.

The page title is 'Dashboard > presentationjenkin > Configuration'.

The main content area is titled 'Build Triggers'.

The 'Configure' section contains several trigger options:

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

The 'Poll SCM' option is selected, indicated by a blue checkmark icon.

The 'Schedule' field contains the cron expression: `* * * * *`.

Below the schedule, there is an unchecked checkbox:  Ignore post-commit hooks ?

At the bottom of the configuration form are 'Save' and 'Apply' buttons.

The Windows taskbar at the bottom of the screen shows various pinned icons and the system tray with the date and time (08-07-2023, 17:46).

# STEP15: Goals>>Install

The screenshot shows a browser window with the URL `localhost:8080/job/presentationjenkin/configure`. The page is titled "Configure" for the job "presentationjenkin". On the left, there is a sidebar with links: General, Source Code Management, Build Triggers, Build Environment, **Build Steps** (which is currently selected), and Post-build Actions. The main content area is titled "Build Steps" and contains a section for "Invoke top-level Maven targets". Under "Goals", the value "install" is entered. There is also an "Advanced..." button and a "Save" button at the bottom.

localhost:1667/get    sunkaratulasi/Projectgit at mast    presentationjenkin Config [Jenk X]

localhost:8080/job/presentationjenkin/configure

Dashboard > presentationjenkin > Configuration

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant ?

## Configure

General

Source Code Management

Build Triggers

Build Environment

**Build Steps**

Post-build Actions

Goals

install

Advanced...

Add build step ▾

Save    Apply

Type here to search    Windows icon    Search icon    Game controller icon    Headphones icon    Task View icon    Start button    Edge icon    File Explorer icon    Mail icon    Word icon    Firefox icon    Jenkins icon    Taskbar icons    ENG US    17:46 08-07-2023

# STEP16: Paths will be build on left

localhost:1667/get    sunkaratulasi/Projectgit at mast    jenpro [Jenkins]

localhost:8080/job/jenpro/

Dashboard > jenpro > Workspace

Add description    Disable Project

Build Now    Configure    Delete Project    Rename

### Permalinks

- Last build (#5), 18 sec ago
- Last stable build (#5), 18 sec ago
- Last successful build (#5), 18 sec ago
- Last completed build (#5), 18 sec ago

### Build History

trend

Filter builds... /

#	Date
#5	8 Jul 2023, 17:57
#4	8 Jul 2023, 17:57
#3	8 Jul 2023, 17:57
#2	8 Jul 2023, 17:57
#1	4 Jul 2023, 13:00

Atom feed for all    Atom feed for failures

REST API Jenkins 2.3752

Type here to search    Windows Start button    Taskbar icons: File Explorer, Mail, Edge, etc.    Status bar: ENG US 17:58 08-07-2023

# STEP17: Success

The screenshot shows a web browser window with four tabs open:

- localhost:1667/get
- New repository · GitHub
- Spring Initializr
- Dashboard [Jenkins]

The Jenkins dashboard is the active tab, displaying the following content:

**Dashboard >**

**Actions:**

- + New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views

**Build Status Legend:**

- S (Green checkmark)
- W (Yellow sun)
- L (Blue gear)

**Icon legend:**

- Atom feed for all
- Atom feed for failures
- Atom feed for just latest builds

**Build Queue:**

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	jenpro	5.3 sec #7	N/A	0.15 sec
...	☀️	projectjenkin	N/A	N/A	N/A

**Build Executor Status:**

1 Idle

**Taskbar:**

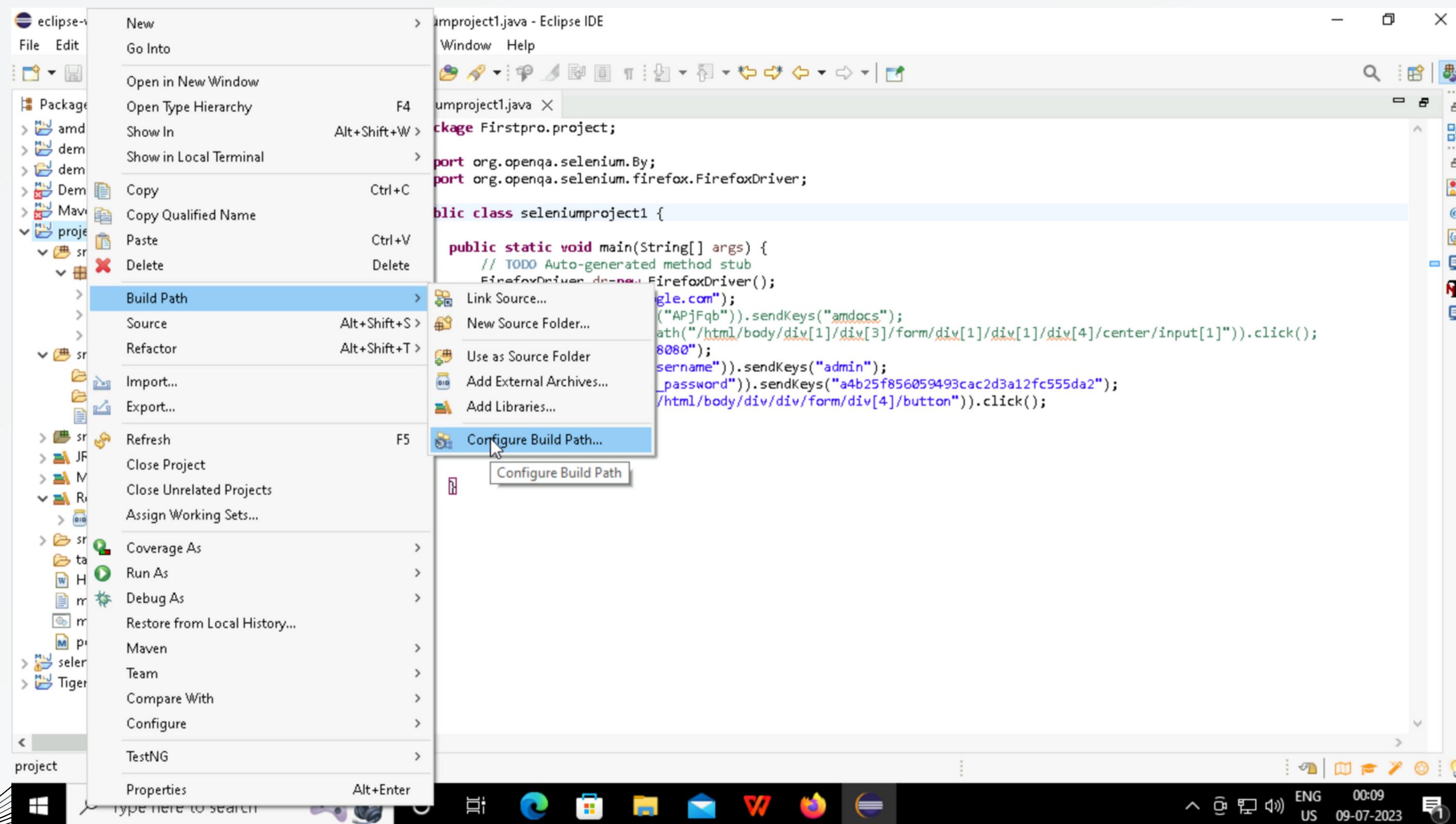
- Type here to search
- File Explorer
- File
- Run
- Task View
- Start
- Microsoft Edge
- File Explorer
- Mail
- Word
- Firefox
- PowerShell
- File

**System Tray:**

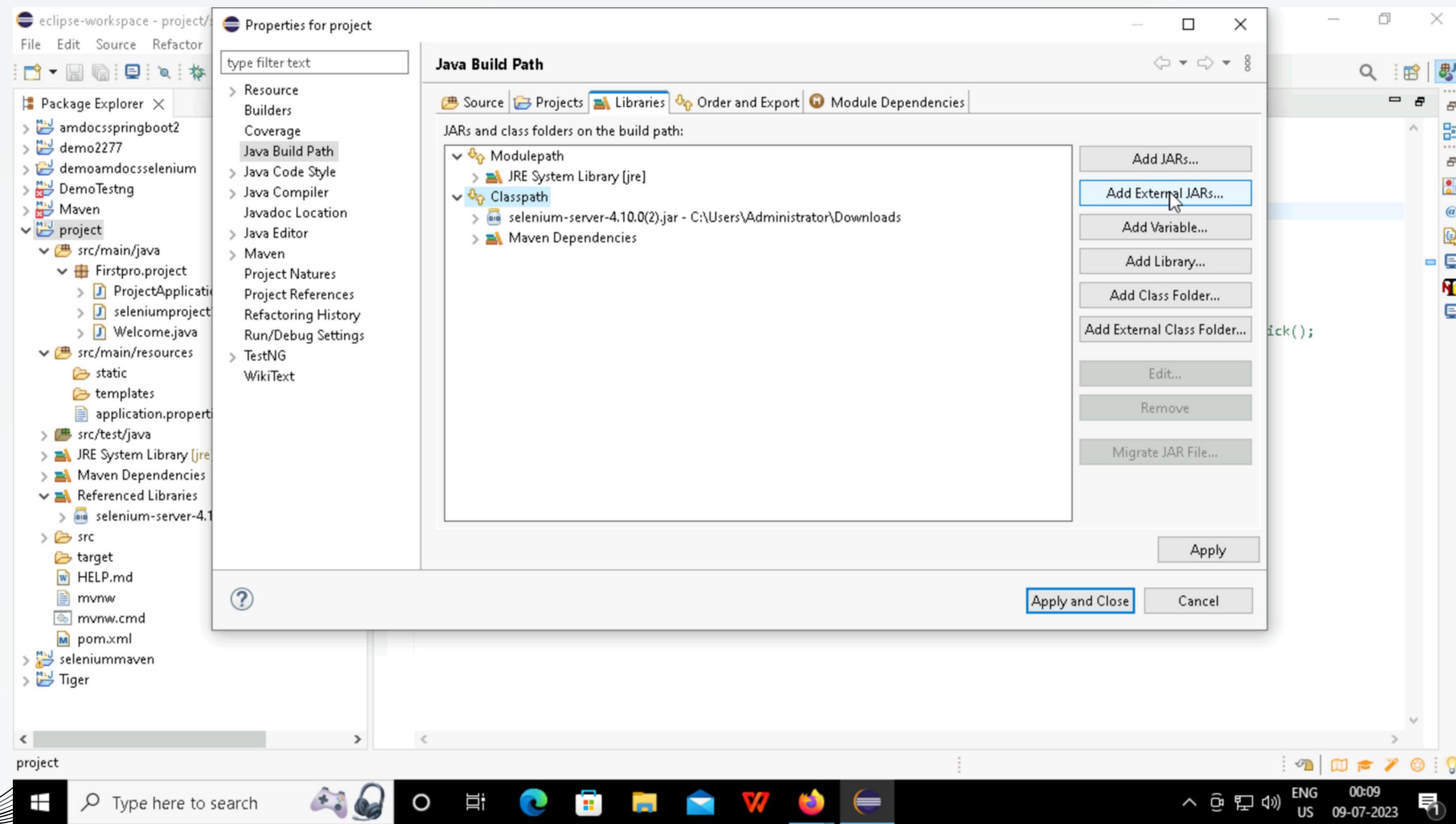
- ENG US
- 21:47
- 08-07-2023
- 1



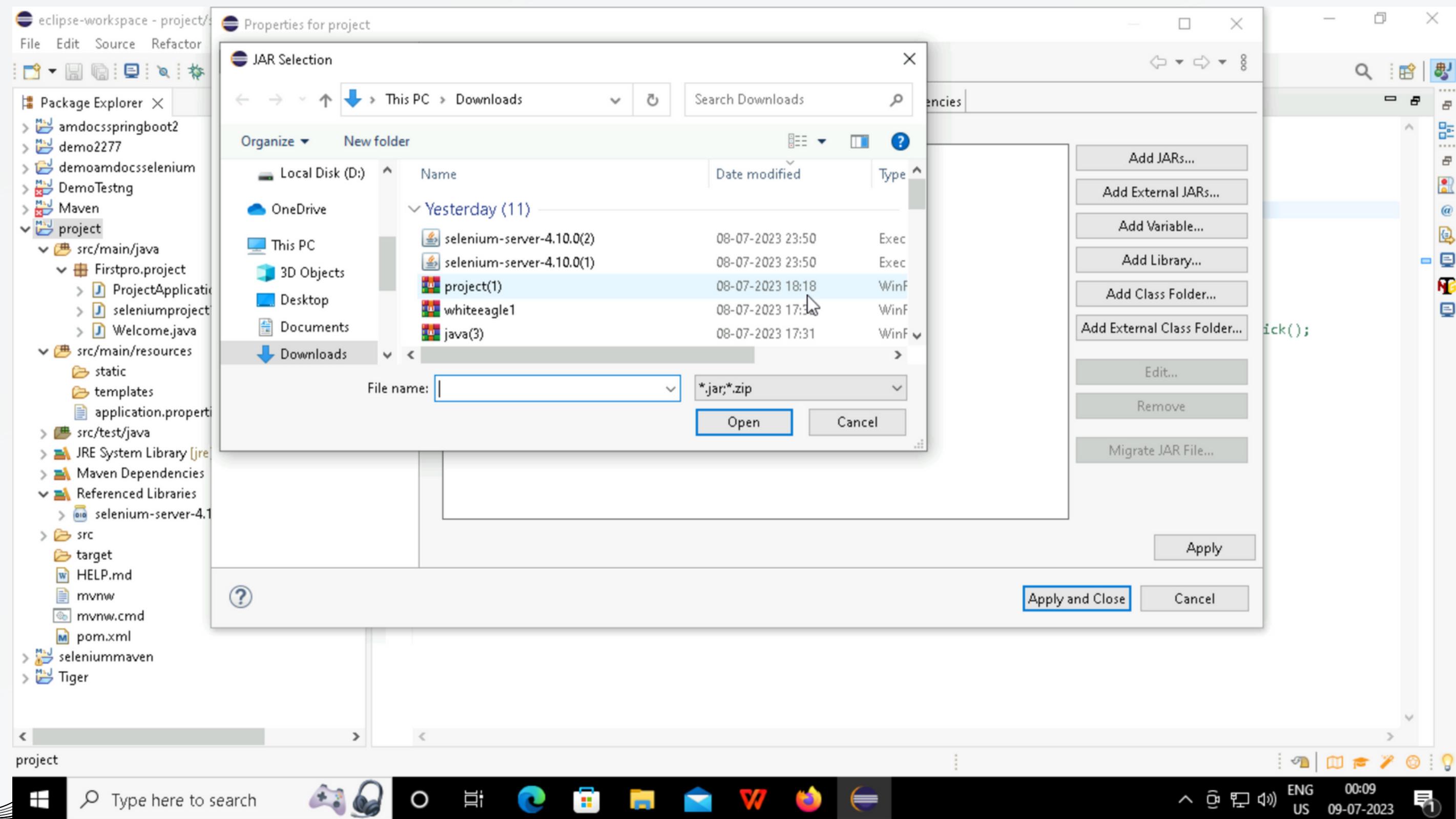
# STEP18: Download>>project>>Buildpath>>configure



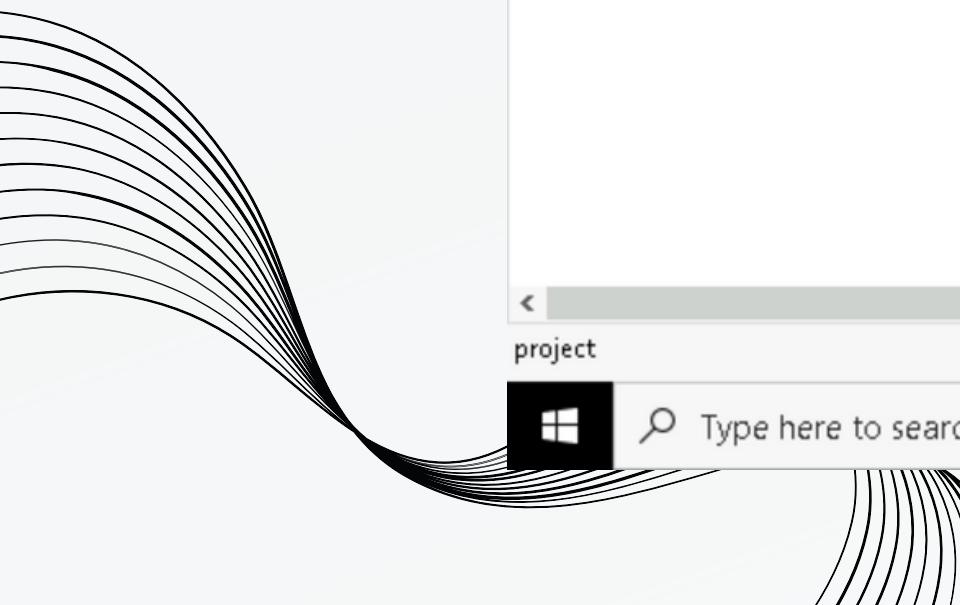
# STEP19: Selenium>>Add External jars>>Apply and close



# STEP 20: Open selenium



# STEP21: New class>>code to open jenkins



eclipse-workspace - project/src/main/java/Firstpro/project/seleniumproject1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X seleniumproject1.java X

```
1 package Firstpro.project;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.firefox.FirefoxDriver;
5
6 public class seleniumproject1 {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        FirefoxDriver dr=new FirefoxDriver();
11        dr.get("http://www.google.com");
12        //dr.findElement(By.id("APjFqb")).sendKeys("amdocs");
13        //dr.findElement(By.xpath("//html/body/div[1]/div[3]/form/div[1]/div[4]/center/input[1]")).click();
14        dr.get("http://localhost:8080");
15        dr.findElement(By.id("j_username")).sendKeys("admin");
16        dr.findElement(By.name("j_password")).sendKeys("a4b25f856059493cac2d3a12fc555da2");
17        dr.findElement(By.xpath("//html/body/div/div/form/div[4]/button")).click();
18    }
19
20
21
22
23
24
25 }
```

Project Explorer View

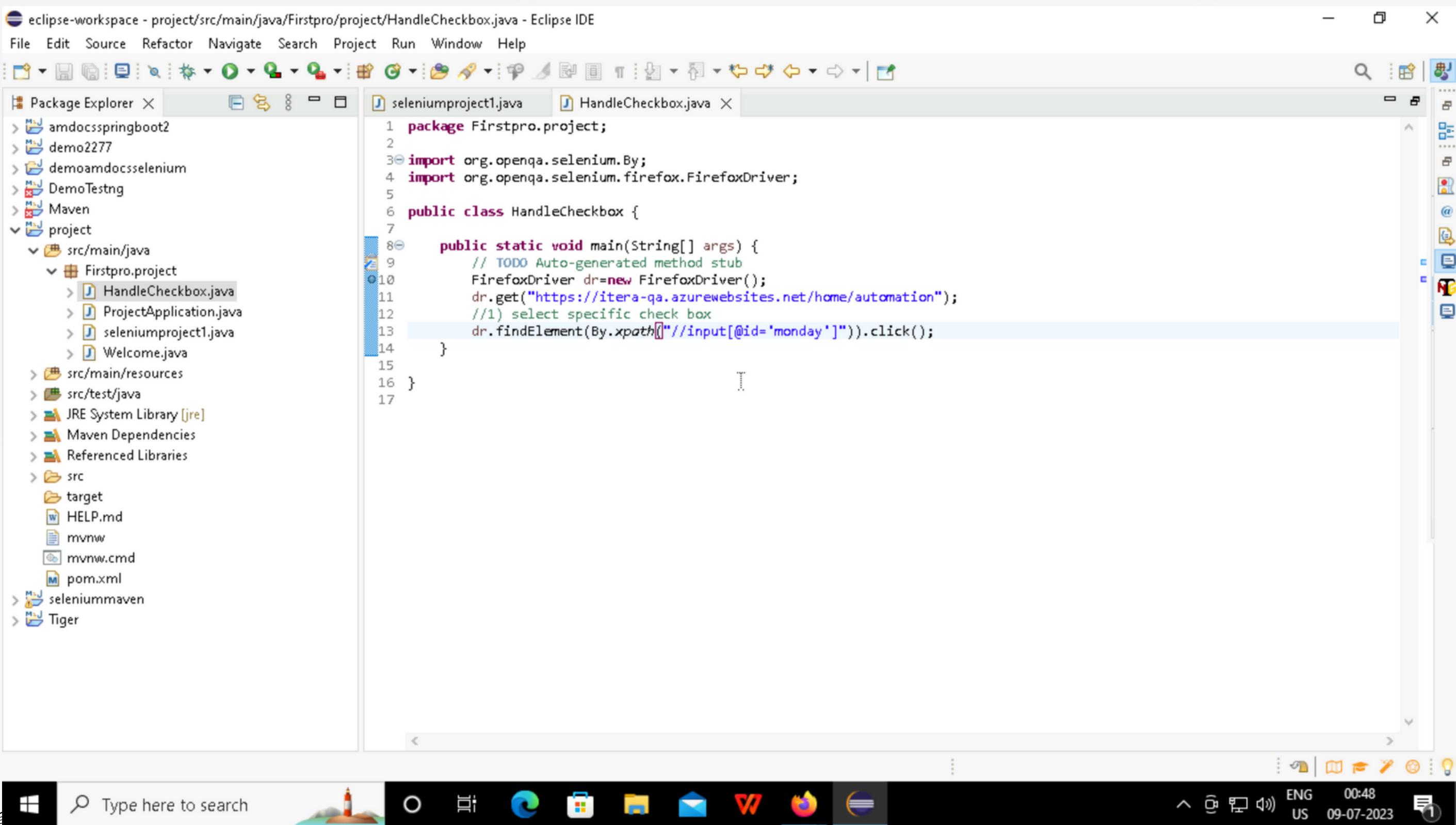
- amdocsspringboot2
- demo2277
- demoamdocsseleium
- DemoTesting
- Maven
- project
  - src/main/java
  - src/main/resources
  - src/test/java
  - JRE System Library [jre]
  - Maven Dependencies
  - Referenced Libraries
    - selenium-server-4.10.0(2).jar - C:\Users\Admini
  - src
  - target
  - HELP.md
  - mvnw
  - mvnw.cmd
  - pom.xml
- seleniummaven
- Tiger

Search Bar: Type here to search

Taskbar Icons: File Explorer, Task View, Start, Edge, File Manager, Mail, Word, Powerpoint, Firefox, File Explorer

System Tray: ENG US 00:10 09-07-2023

# STEP22: Code select specific check box

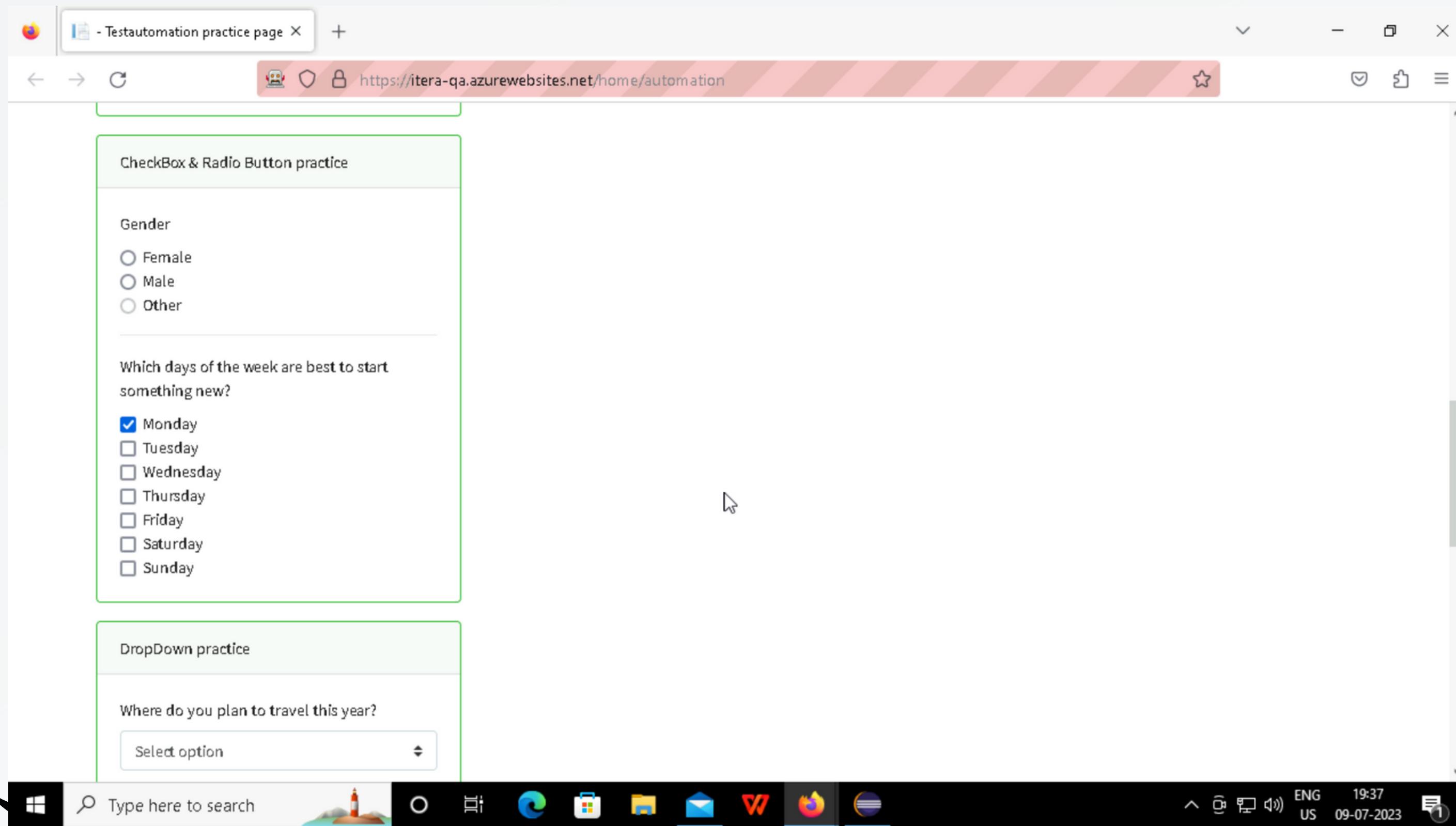


The screenshot shows the Eclipse IDE interface with the title bar "eclipse-workspace - project/src/main/java/Firstpro/project/HandleCheckbox.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Find, and Run. The left sidebar is the Package Explorer showing projects like "amdocsspringboot2", "demo2277", "demoamdocsseleium", "DemoTesting", "Maven", and "project". The "src/main/java" folder under "project" contains packages "Firstpro.project" with files "HandleCheckbox.java", "ProjectApplication.java", "seleniumproject1.java", and "Welcome.java", and "src/main/resources", "src/test/java", "JRE System Library [jre]", "Maven Dependencies", "Referenced Libraries", "src" with "target", "HELP.md", "mvnw", "mvnw.cmd", "pom.xml", "seleniummaven", and "Tiger". The right panel displays the code editor for "HandleCheckbox.java". The code is as follows:

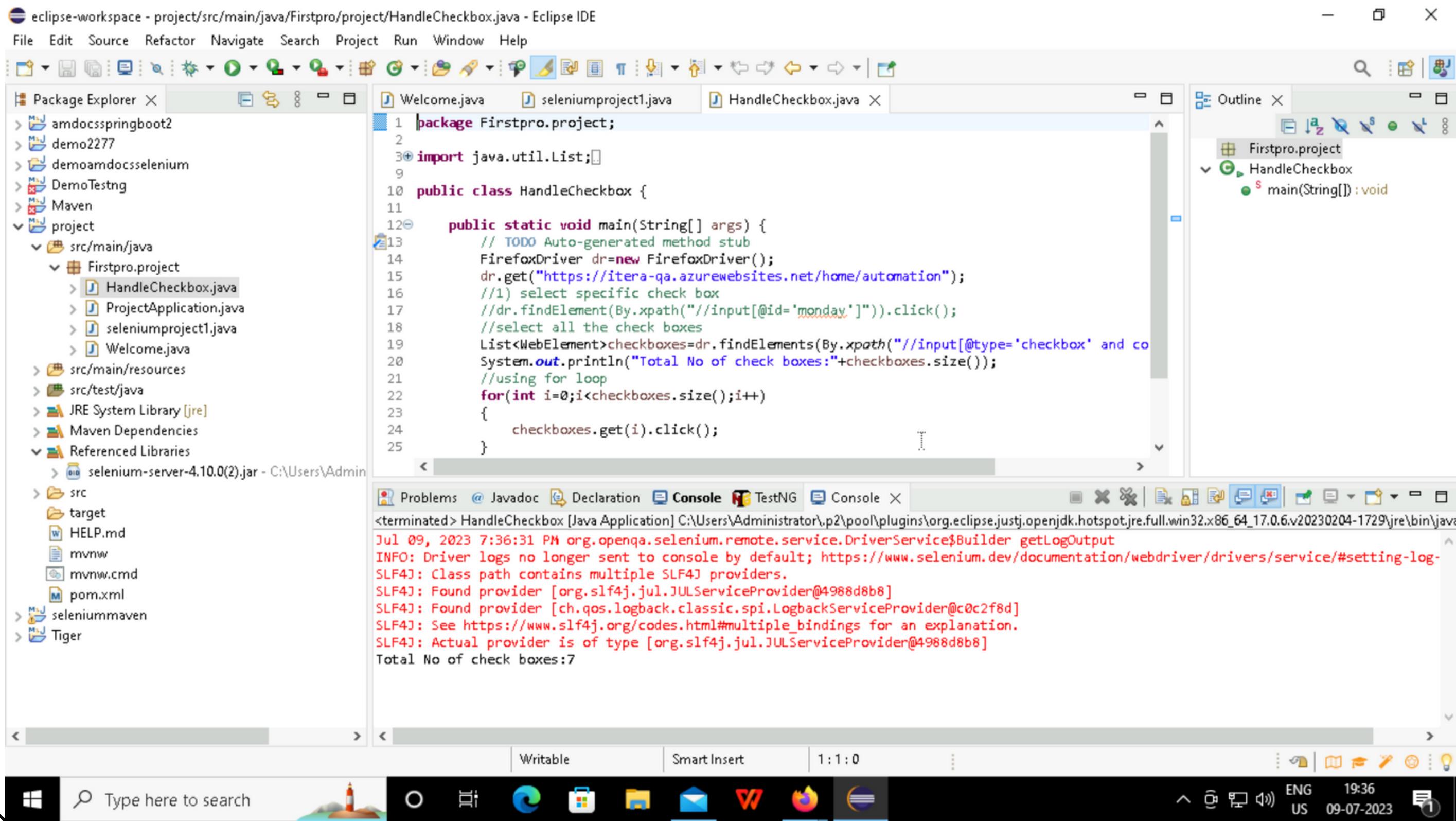
```
1 package Firstpro.project;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.firefox.FirefoxDriver;
5
6 public class HandleCheckbox {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        FirefoxDriver dr=new FirefoxDriver();
11        dr.get("https://itera-qa.azurewebsites.net/home/automation");
12        //1) select specific check box
13        dr.findElement(By.xpath("//input[@id='monday']")).click();
14    }
15
16 }
17
```

The code imports the necessary Selenium packages and defines a class "HandleCheckbox". The "main" method initializes a Firefox driver, navigates to a specific URL, and then performs an action on an element located by its XPath.

# STEP23: Output of one checkbox



# STEP24: Code to select all checkboxes



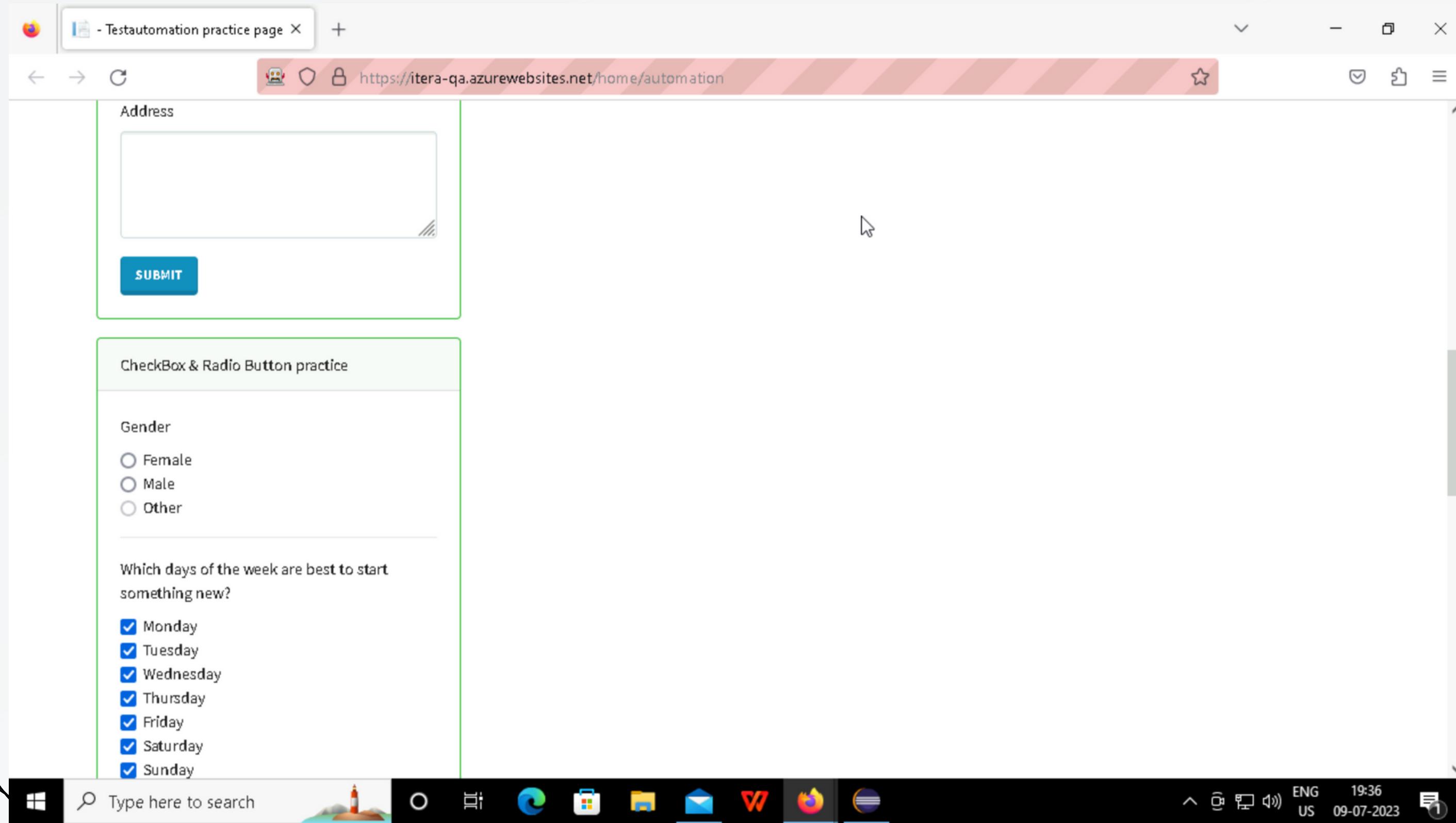
The screenshot shows the Eclipse IDE interface with the following components:

- Package Explorer:** Shows the project structure with a package named "Firstpro.project" containing files like HandleCheckbox.java, ProjectApplication.java, seleniumproject1.java, and Welcome.java.
- Editor:** Displays the Java code for "HandleCheckbox.java". The code uses Selenium WebDriver to navigate to a website, find checkboxes, and click them one by one using a for loop.
- Outline:** Shows the class structure with a main method and a constructor.
- Console:** Displays the terminal output of the Java application, showing the log messages from Selenium and the result of the script execution.

```
1 package Firstpro.project;
2
3 import java.util.List;
4
5 public class HandleCheckbox {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         FirefoxDriver dr=new FirefoxDriver();
10        dr.get("https://itera-qa.azurewebsites.net/home/automation");
11        //1) select specific check box
12        //dr.findElement(By.xpath("//input[@id='monday']")).click();
13        //select all the check boxes
14        List<WebElement>checkboxes=dr.findElements(By.xpath("//input[@type='checkbox' and co
15        System.out.println("Total No of check boxes:"+checkboxes.size());
16        //using for loop
17        for(int i=0;i<checkboxes.size();i++)
18        {
19            checkboxes.get(i).click();
20        }
21    }
22}
```

Jul 09, 2023 7:36:31 PM org.openqa.selenium.remote.service.DriverService\$Builder getLogOutput  
INFO: Driver logs no longer sent to console by default; <https://www.selenium.dev/documentation/webdriver/drivers/service/#setting-log>  
SLF4J: Class path contains multiple SLF4J providers.  
SLF4J: Found provider [org.slf4j.jul.JULServiceProvider@4988d8b8]  
SLF4J: Found provider [ch.qos.logback.classic.spi.LogbackServiceProvider@c0c2f8d]  
SLF4J: See [https://www.slf4j.org/codes.html#multiple\\_bindings](https://www.slf4j.org/codes.html#multiple_bindings) for an explanation.  
SLF4J: Actual provider is of type [org.slf4j.jul.JULServiceProvider@4988d8b8]  
Total No of check boxes:7

# STEP 24: Output of selecting all the checkboxes



*Thank  
You*

Ceo Of Ingoude  
Company

Ceo Of Ingoude  
Company

Ceo Of Ingoude  
Company