CS 512 Software Systems (ESD)

Lab 02 - Frontend Development

Lab Objectives

In this lab we will have a brief overview of html, css and JavaScript. We would be having some hands-on the html tags, forms, list and beautifying these html pages with css. Finally we'll make those webpages alive using JavaScript.

At the end of this lab, the student will be able to carry out the following tasks:

1. Create their own webpage.
2. Beautify it with css (bootstrap)
3. Event handling using javascript
4. Form validation
5. Integrating it with intellij
6. Running a code module on apache tomcat

Lab Setup

The front-end won't be needing much of the lab setup we already have intellij installed on our systems, but it's up to you to try out any other editor for html and js.

Before coming to the lab, student is expected to have carried out the following setup activities

1. Download Apache Tomcat : Apache Tomcat (Core -> tar.gz)
2. git clone https://github.com/shubhamaggarwal890/esd-frontend.git
3. Download bootstrap examples : Bootstrap

Lab Activities

1. Cloning esd-frontend repository which contains some examples to html, css and javascript

```
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware$ mkdir Lab2-ESD
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware$ cd Lab2-ESD/
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware/Lab2-ESD$ git clone https://github.com/shubhamaggarwal890/esd-frontend
Cloning into 'esd-frontend'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 63 (delta 17), reused 59 (delta 16), pack-reused 0
Unpacking objects: 100% (63/63), 466.88 KiB | 646.00 KiB/s, done.
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware/Lab2-ESD$ cd esd-frontend/
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware/Lab2-ESD/esd-frontend$ ls
Basics-1  README.md  WithCss-2  WithJS-3
shubham@shubham:~/Documents/IIITB/2020-SystemSoftware/Lab2-ESD/esd-frontend$
```

Figure 1: Cloning esd-frontend from github

2. Start intellij application and open our cloned project on intellij

Figure 2: Start intellij and choose for **Open or Import** option

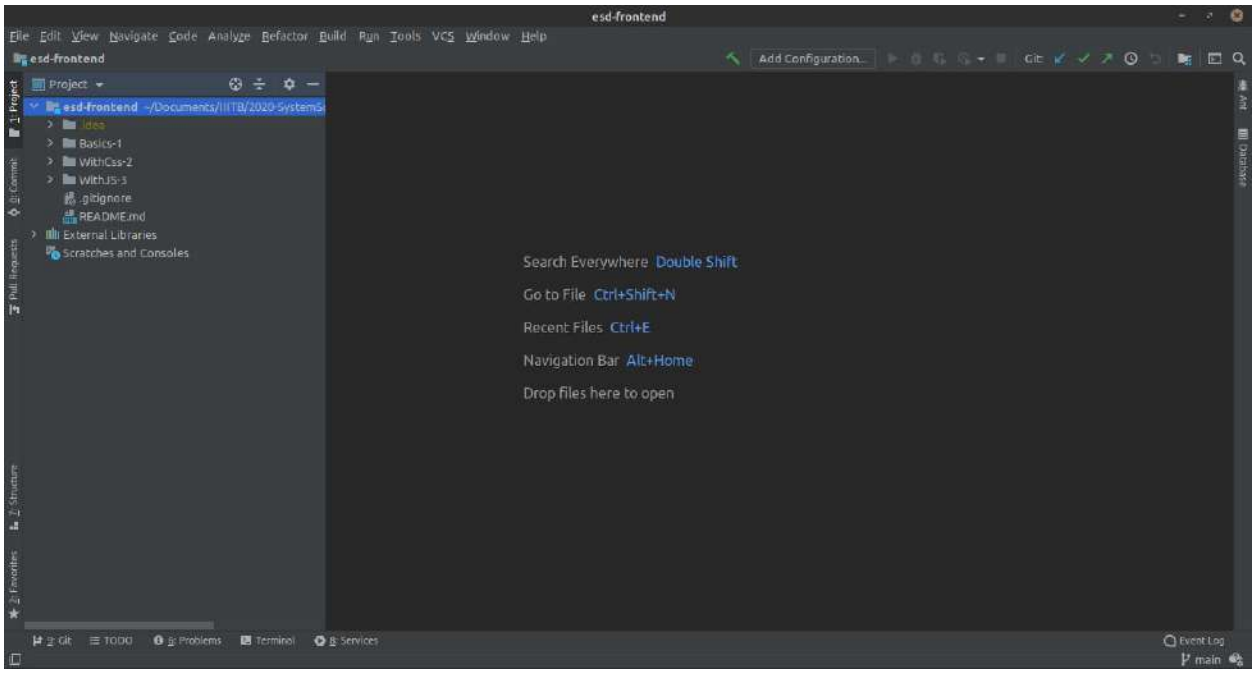

Figure 3: Select where the folder where you have downloaded the git repo and open it

3. Start with the Basics-1 module and open the html file and there on the top right corner select your browser on which you want to run on. Sometimes that little tab is not available, you can also right click and choose for **Open in Browser.**

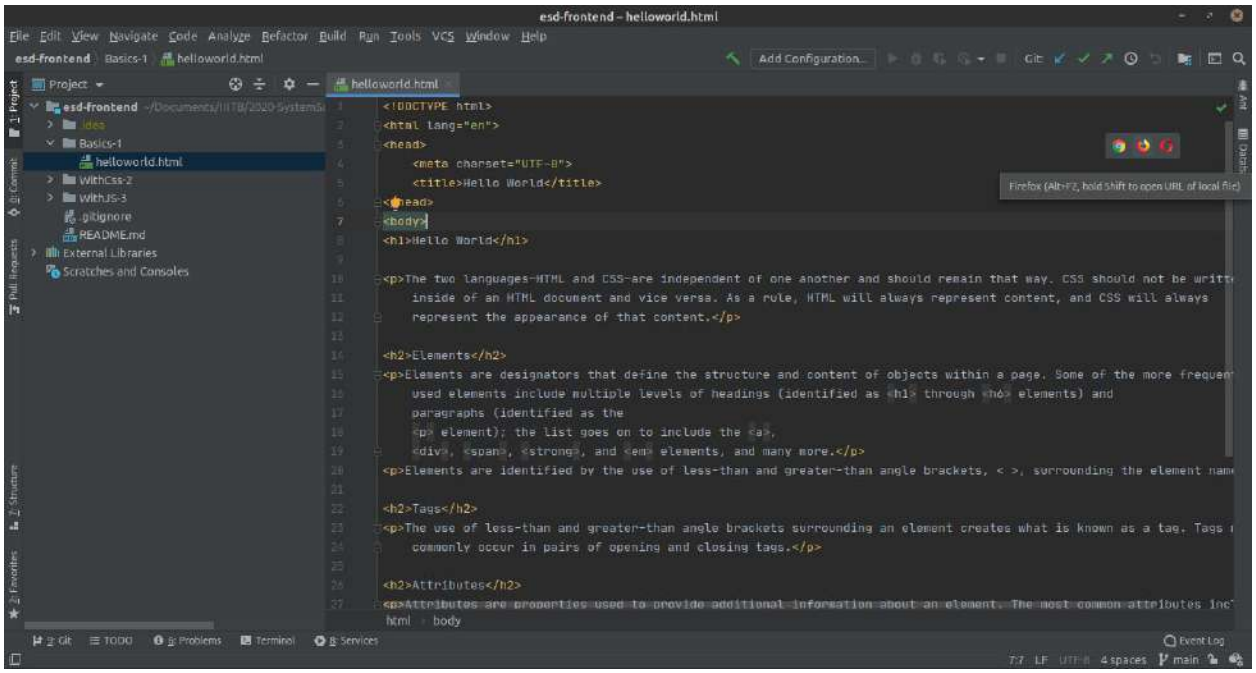

Figure 4: Editing screen for Basics-1 on intellij

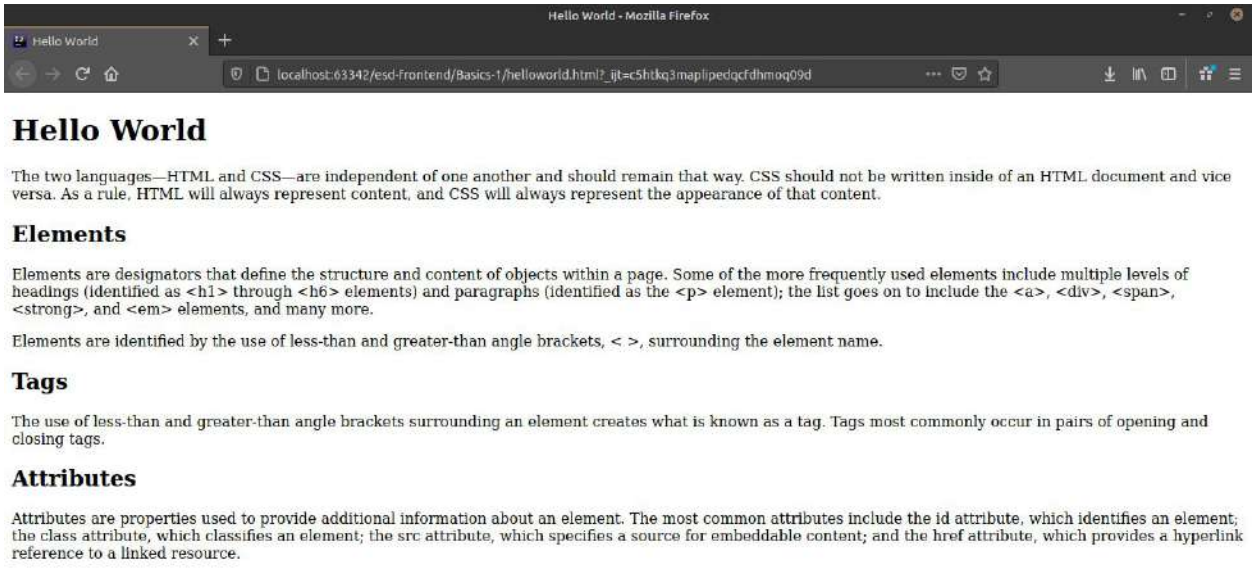

Figure 5: Intellij should create a local server and open the written html on your browser

4. Creating a new Java Enterprise project on intellij and configuring intellij to run on Tomcat server. Start a new Project, **File -> New -> Project.** Select Java Enterprise on the left drawer of intellij and check for Java. Press Next

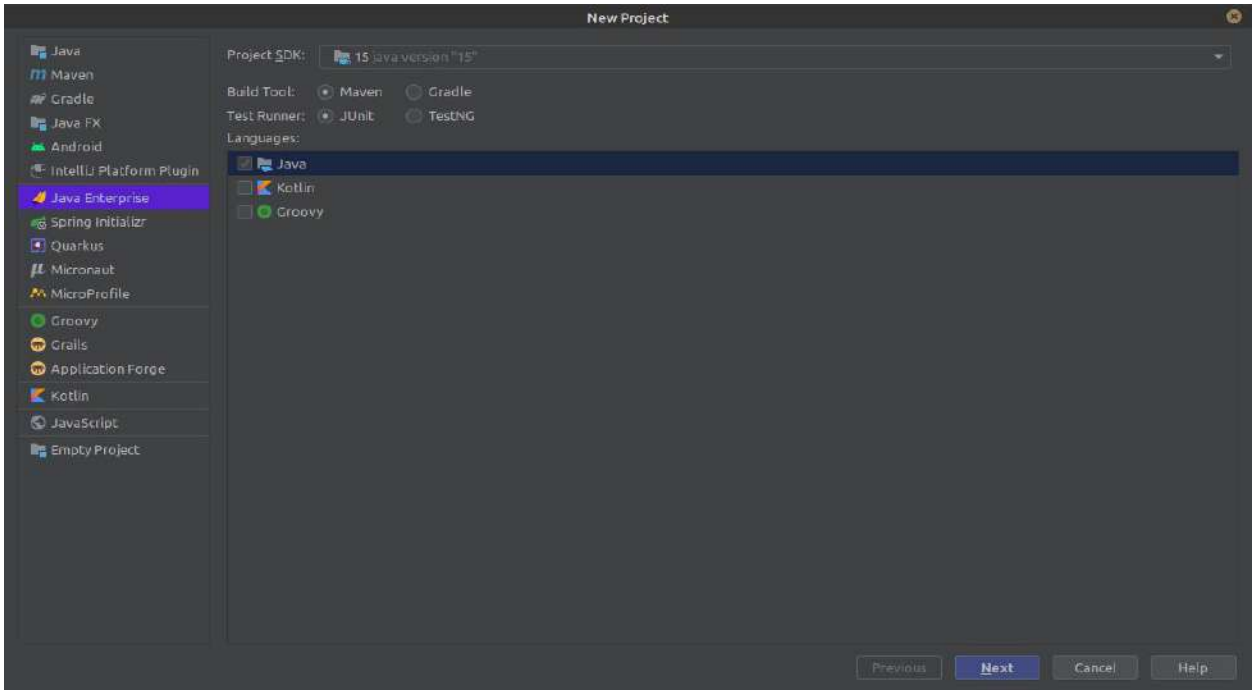

Figure 6: New project on intellij, choose Java Enterprise on the left drawer

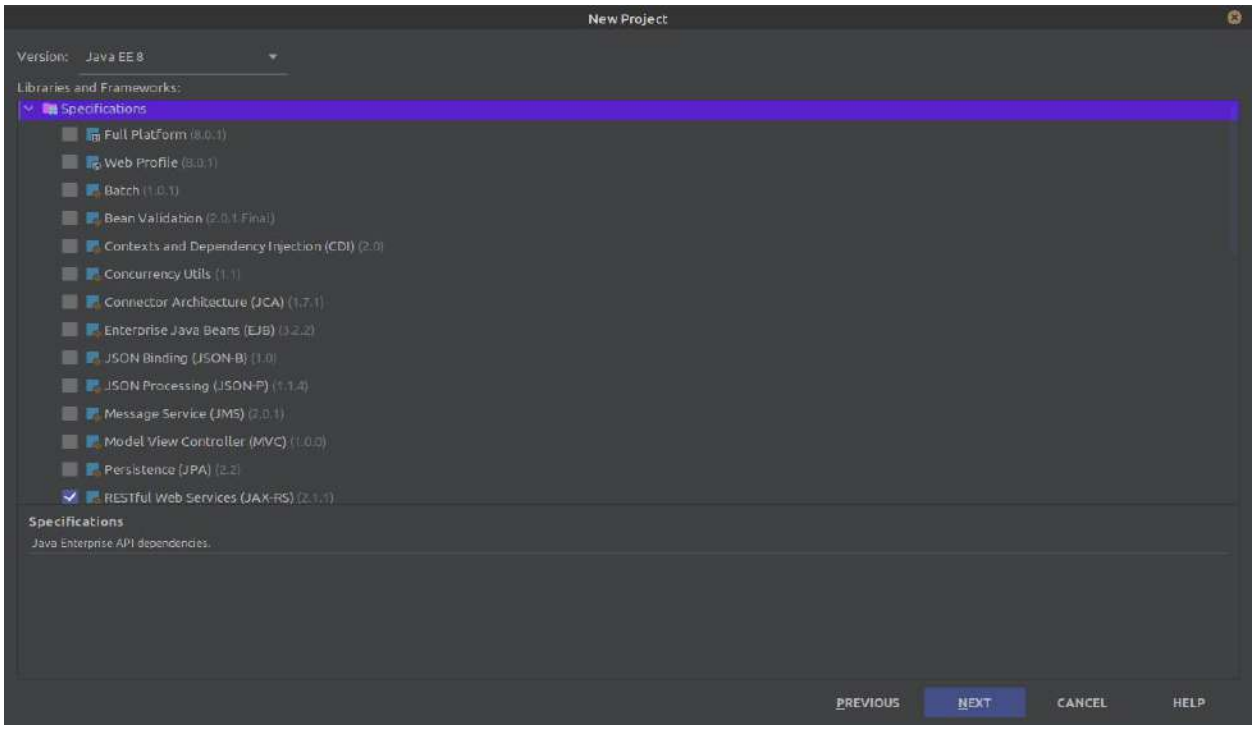

Figure 7: Choose **RESTful Web Services (JAX-RS) under Specifications**

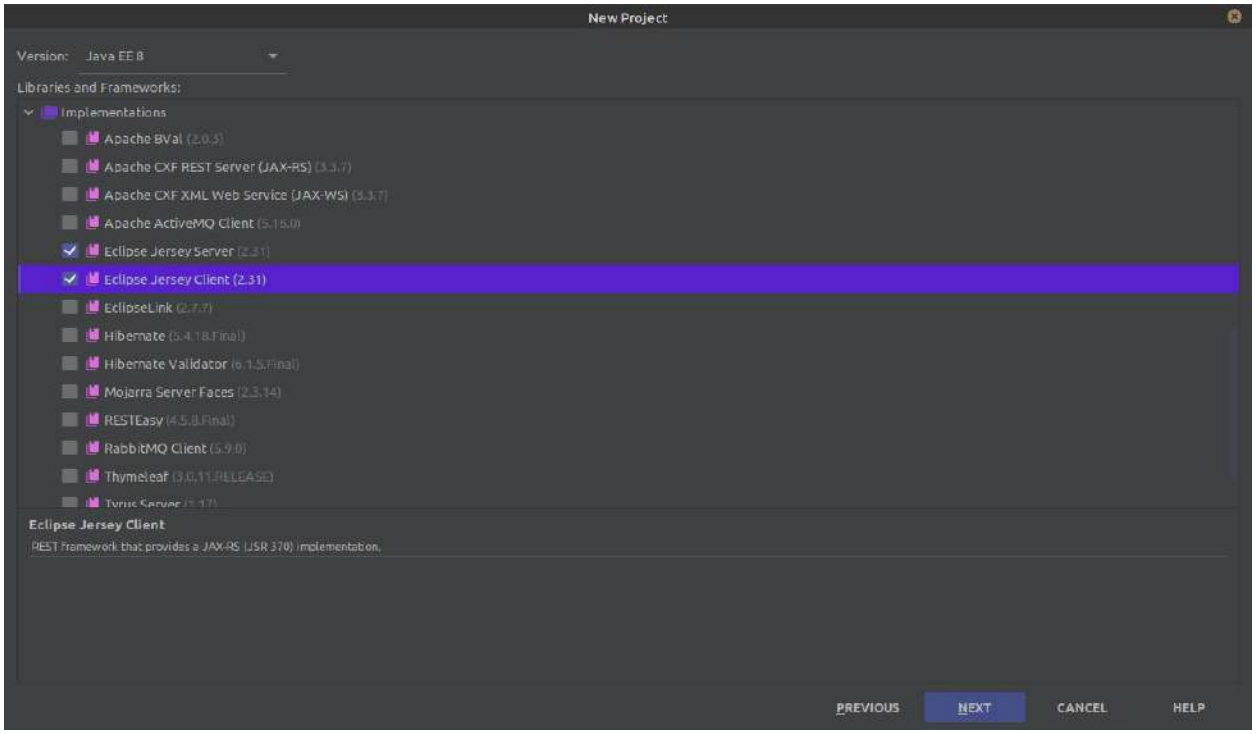

Figure 7.1: Choose **Eclipse Jersey Server and Eclipse Jersey Client** under Implementations

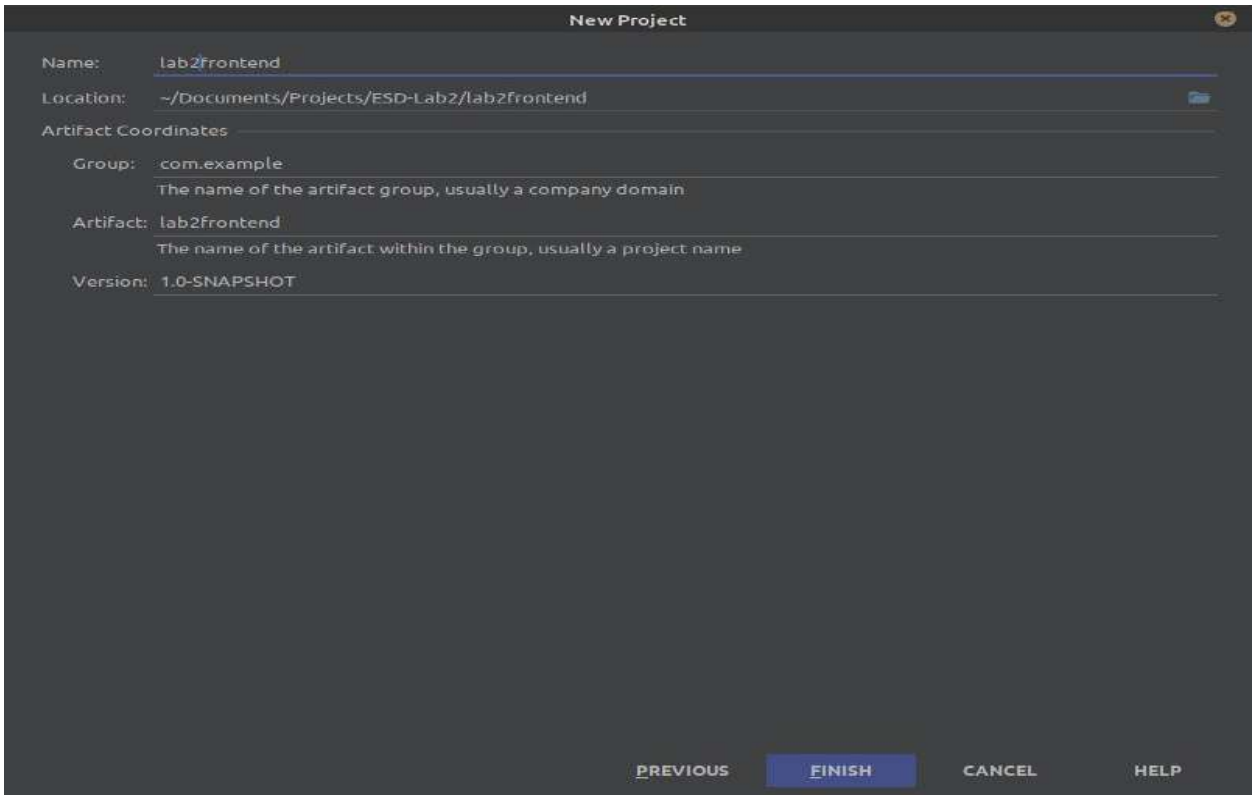

Figure 8: Finish new project while entering appropriate name, **one suggestion please set the name in lowercase and don't add any special character**

5. Here you'll see something like this as the finish screen, here under webapp, create a new folder called assets and under assets create three sub folders called css, fonts, js.
6. Under webapp create a new html file named **index.html** and write some html code in it.

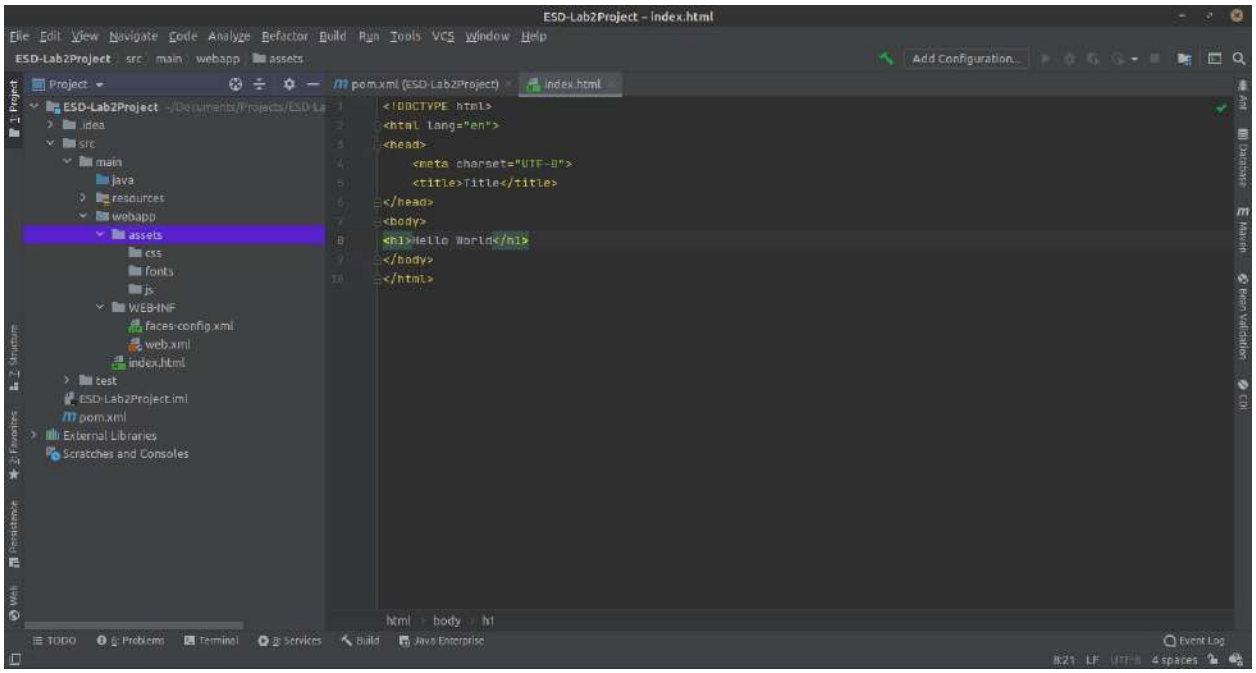

Figure 9: HTML file with hello world html code

7. We have our code ready, the maven project for front-end is set, now we have to build it and deploy it on Apache Tomcat server.
8. Under **File -> Settings -> Build, Execution, Deployment,** select **Application Servers.**

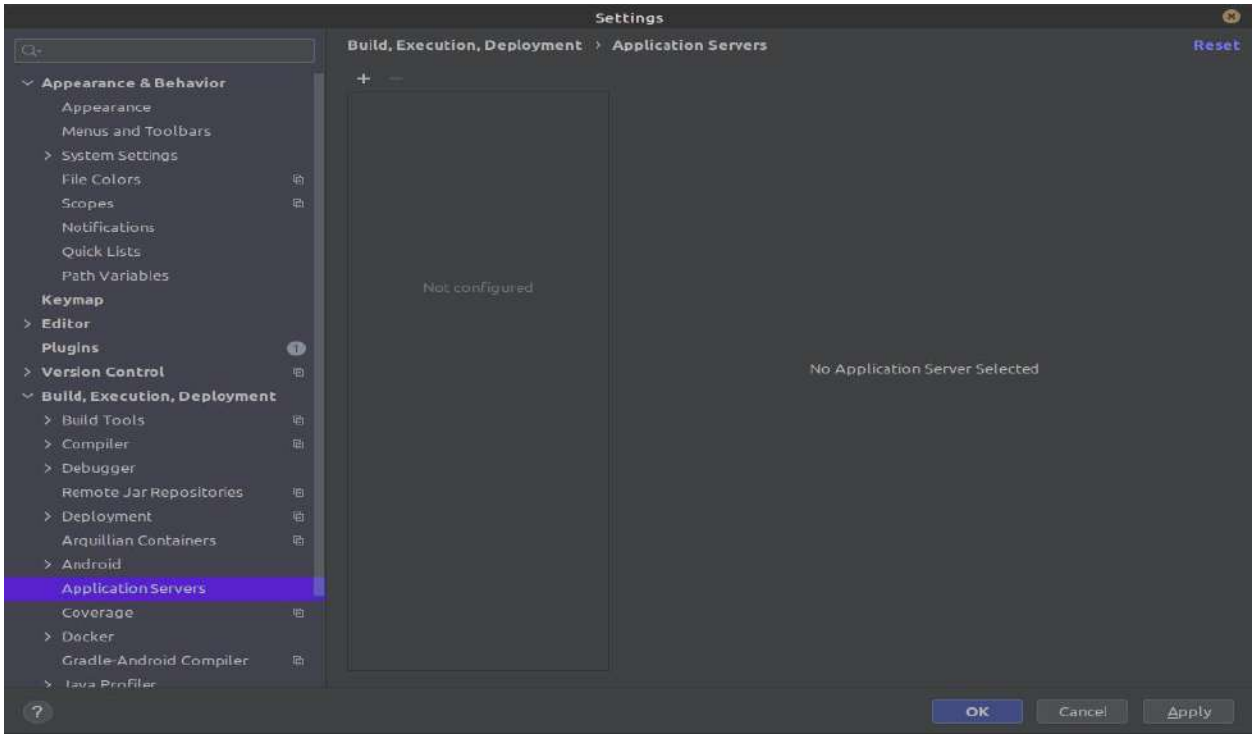

Figure 10: **File -> Settings -> Build, Execution, Deployment -> Application Servers,** click over that **+** symbol

9. Select **Tomcat Server,** and set Tomcat Home as the path where you have downloaded the Tomcat server. Only the home path is fine, no subdirectories path needed.

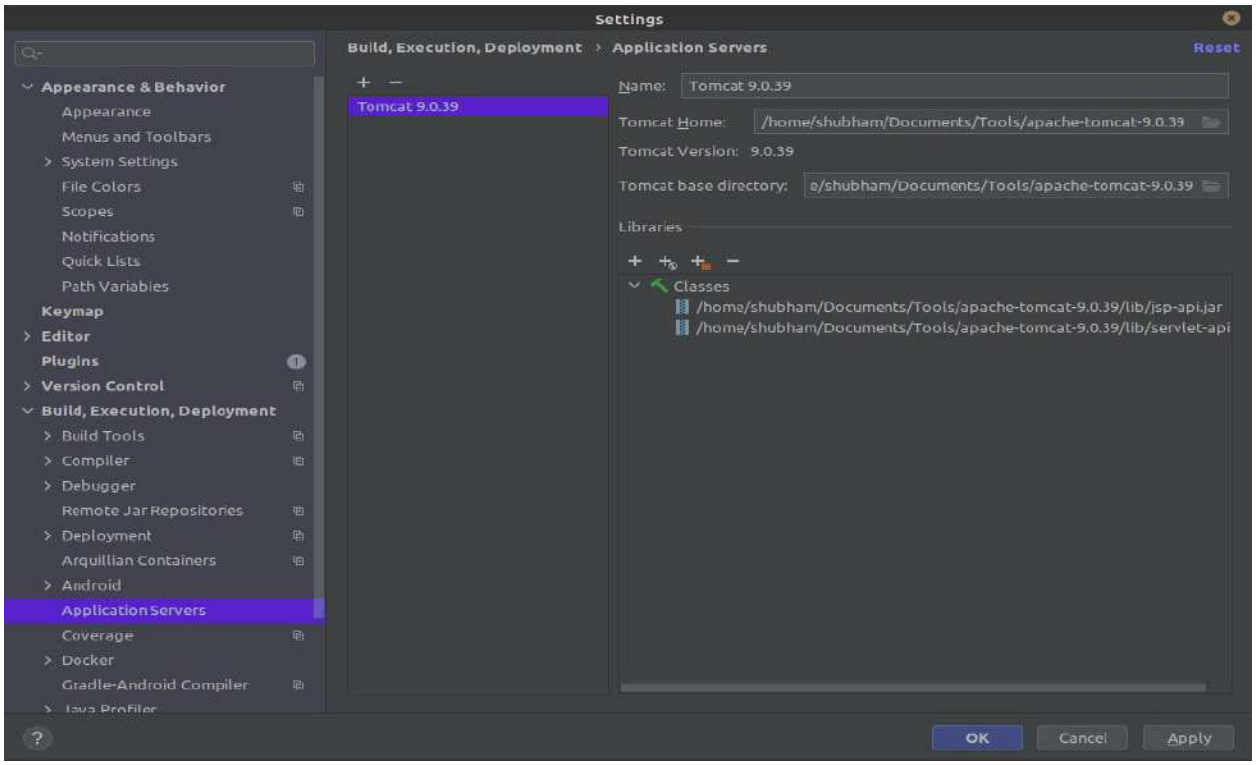

Figure 11: Tomcat server configured with Intellij

10. Now lets run our html code on tomcat server after we build it first in war file. Select **Run -> Edit Configurations.**

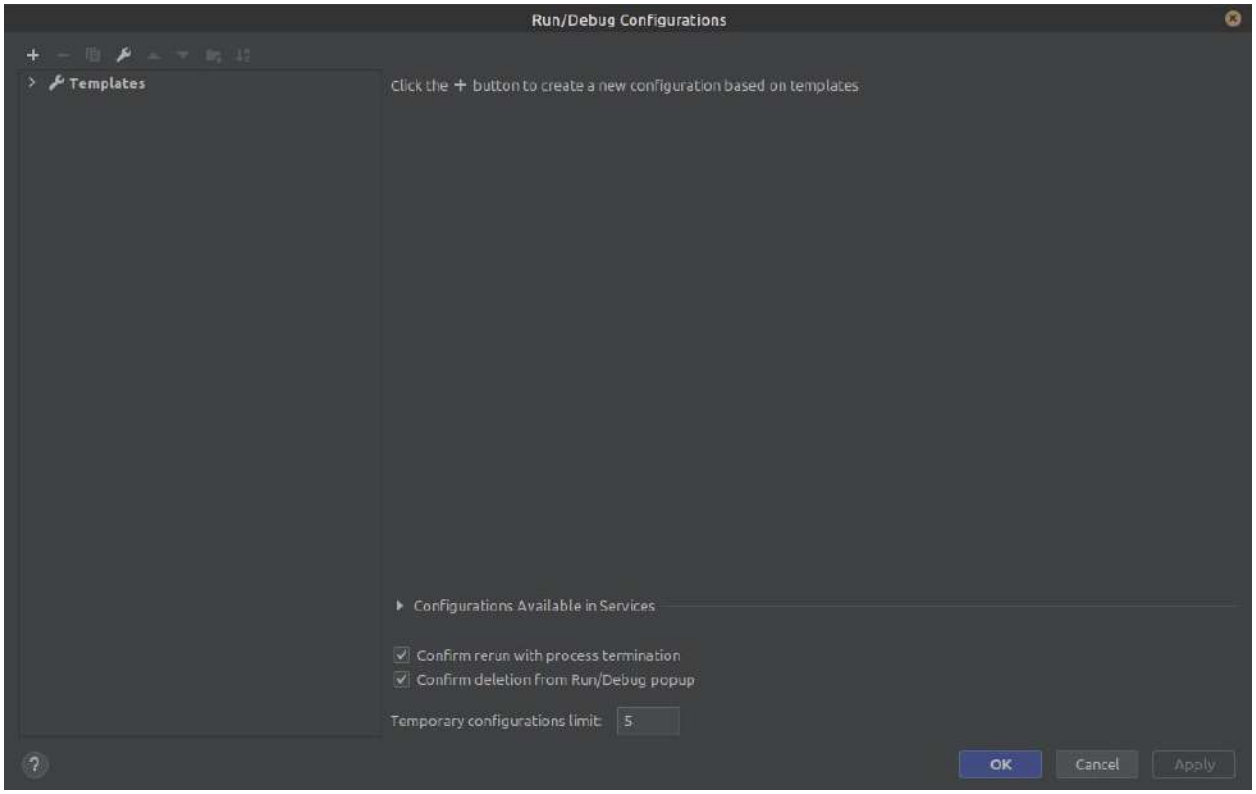

Figure 12: Configuring Intellij to run on Tomcat server

11. Click over that **+** button and choose for **TomcatServer (Local).** Click on that **Fix** button and choose for **<projectname>:war.** You can also do the same under Deployments tab, by clicking over **+ button** in Deployment tab and choosing for **artifact** and then selecting the suitable **war** file.

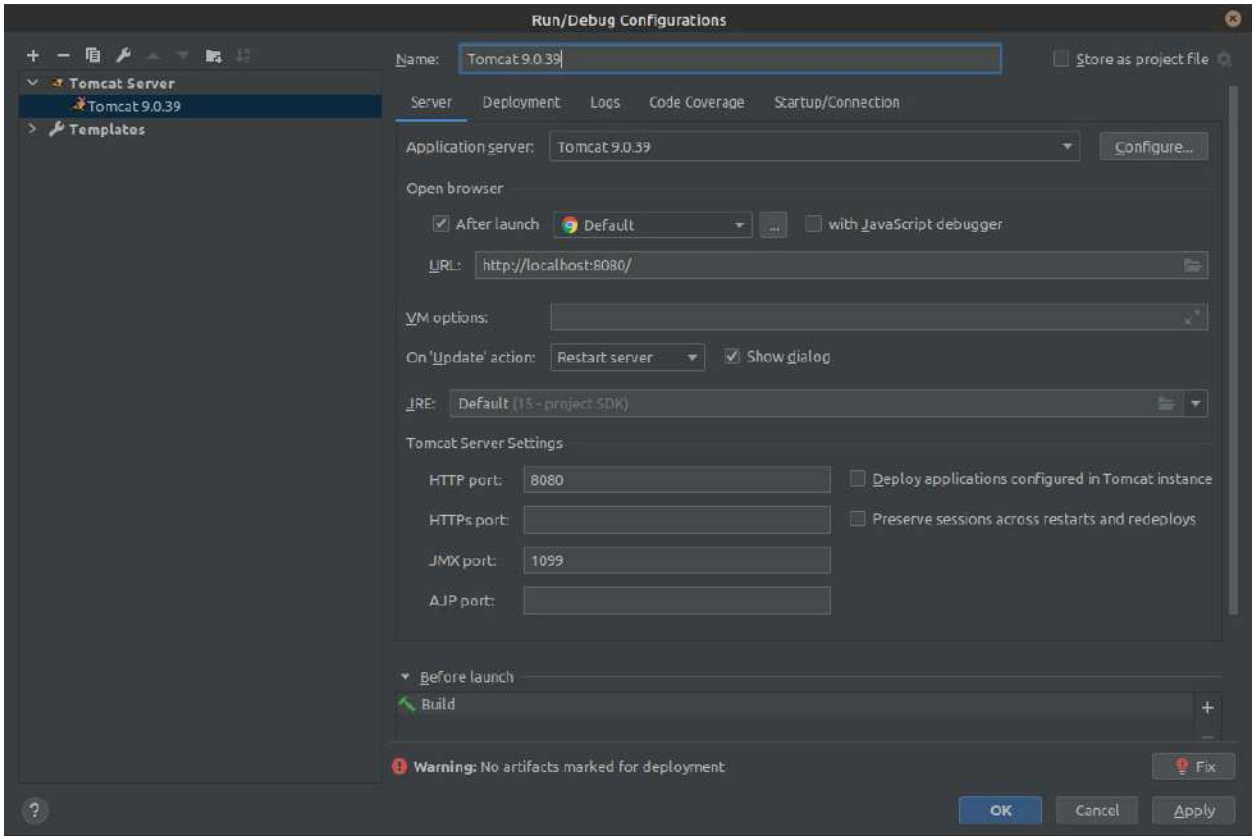

Figure 13: Configuring Tomcat server to run the war file

12. Finally you can run your tomcat server and view your webpages. By default tomcat runs on port 8080. You can view your webpage over address http://localhost:8080/<Project Name_<war or war_exploded>>

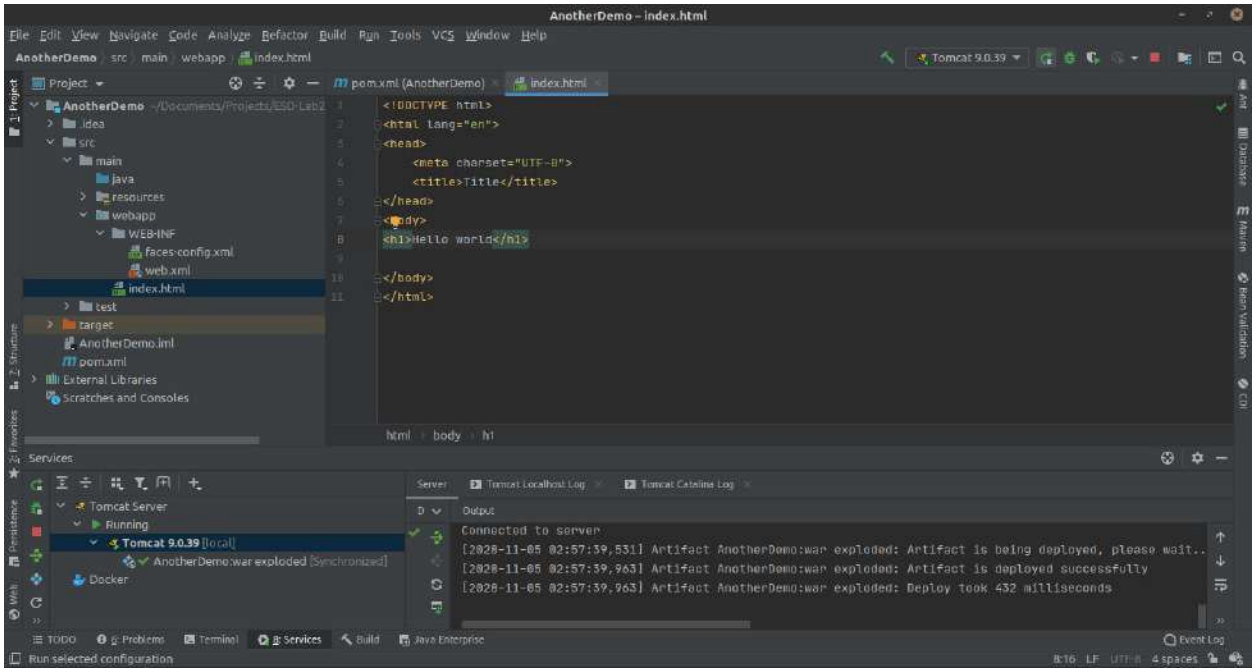

Figure 14: Running tomcat server and deploying war files over it.

Hello World

Figure 15: Tomcat server running and your webpages.

Various CSS Frameworks

1. Bootstrap
2. Foundation
3. Bulma
4. UIkit
5. Semantic UI
6. Susy
7. Materialize
8. Pure
9. Skeleton
10. Milligram

Various JS Frameworks

1. Angular
2. React
3. Vue.js
4. Ember.js
5. Meteor

Reference / Help Resources

1. https://learn.shayhowe.com/advanced-html-css/
2. https://javascript.info/
3. https://getbootstrap.com/
4. https://www.jetbrains.com/help/idea/creating-and-running-your-first-restful-web-service.html?section=Tomcat#run_config