

Ex. No: 3**Date: 04.04.2022****IMPLEMENTATION OF SOAP AND RESTFUL
WEB SERVICES IN JAVA****AIM:**

To implement SOAP and Restful web services in Java

PROCEDURE:**SOAP:**

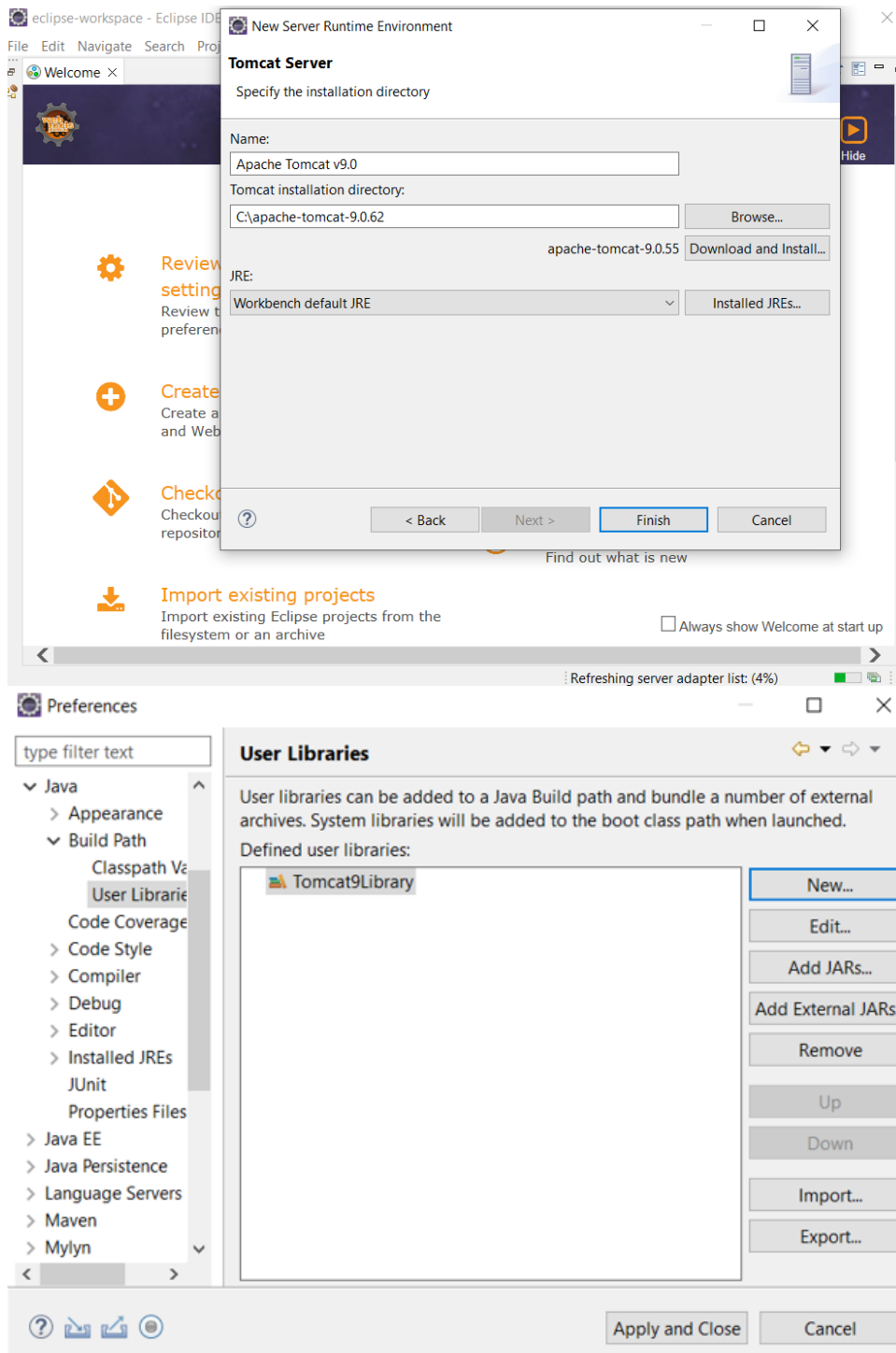
1. Installation of Eclipse IDE for Java EE Developers
2. Installation Tomcat Apache Web Server
3. Setup Tomcat in Eclipse IDE
4. Create new Dynamic web project
5. Create a web server

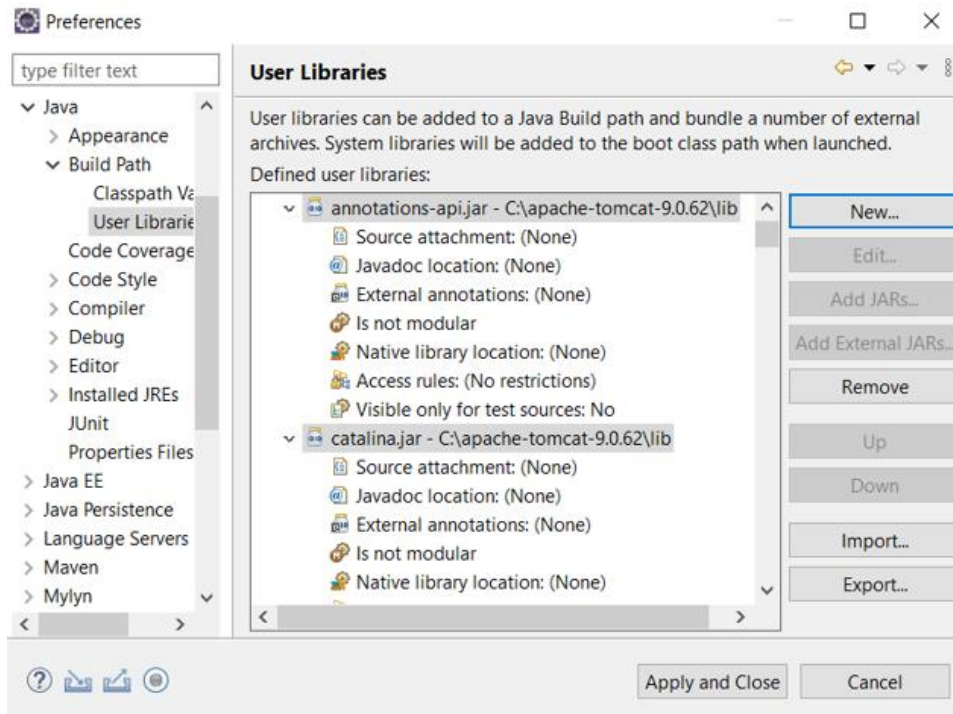
RESTFUL:

1. Install necessary npm packages
2. Install POSTMAN in Google chrome extension
3. Set "GET"
4. Enter URL and click send
5. Set "POST" and click send
6. Notified as 201 so it sends and receives the request

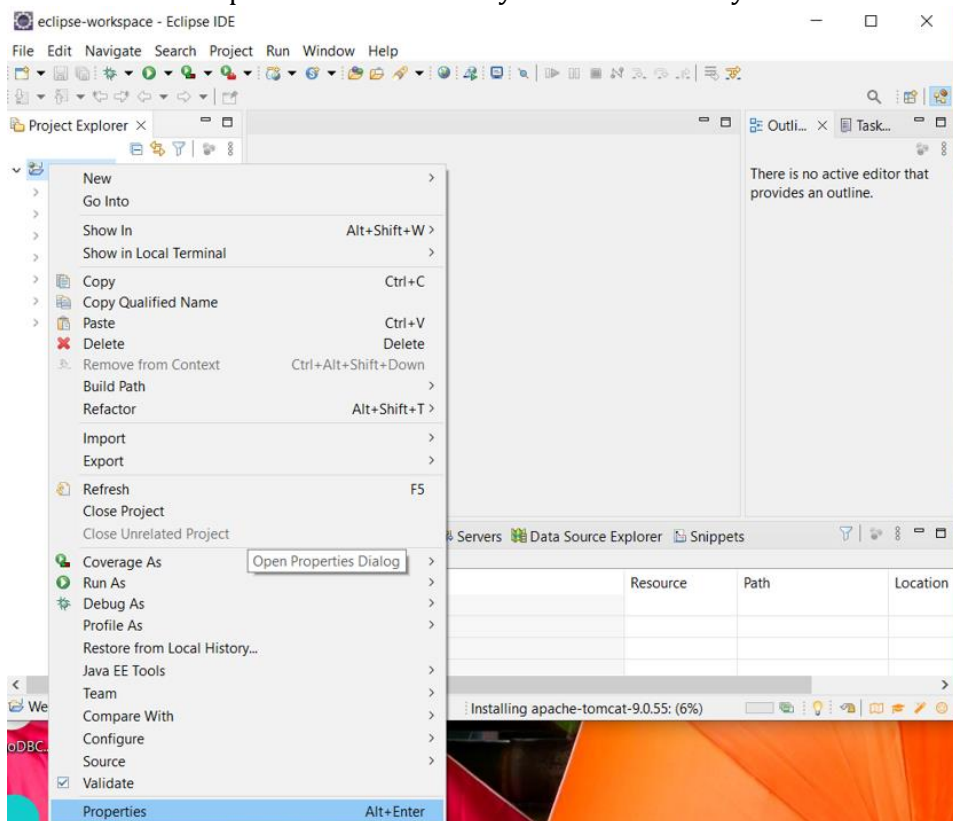
SOAP STEPS:

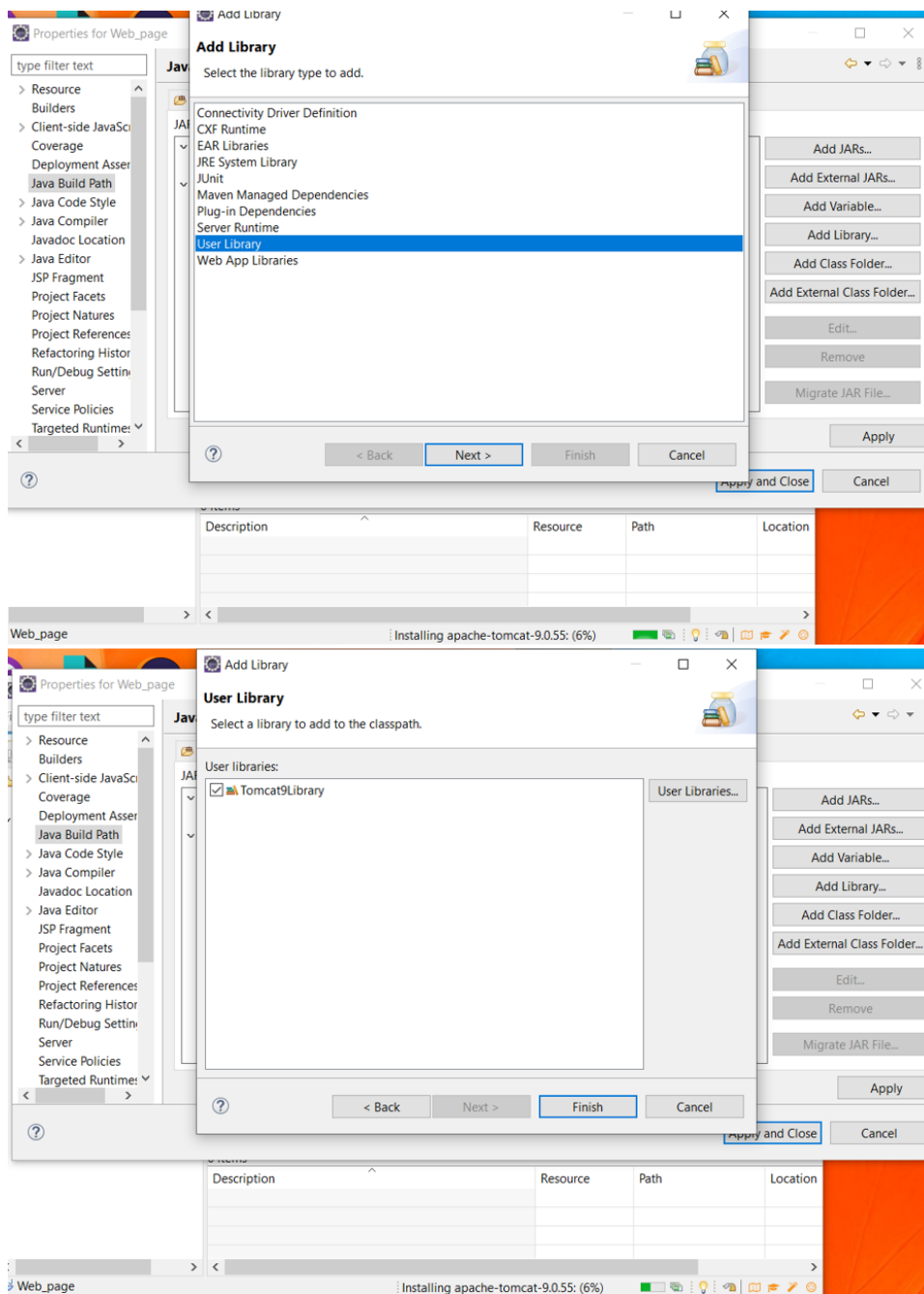
1. Select Windows -> Preferences -> Server Runtime Environment -> Add -> Select path of Tomcat Directory



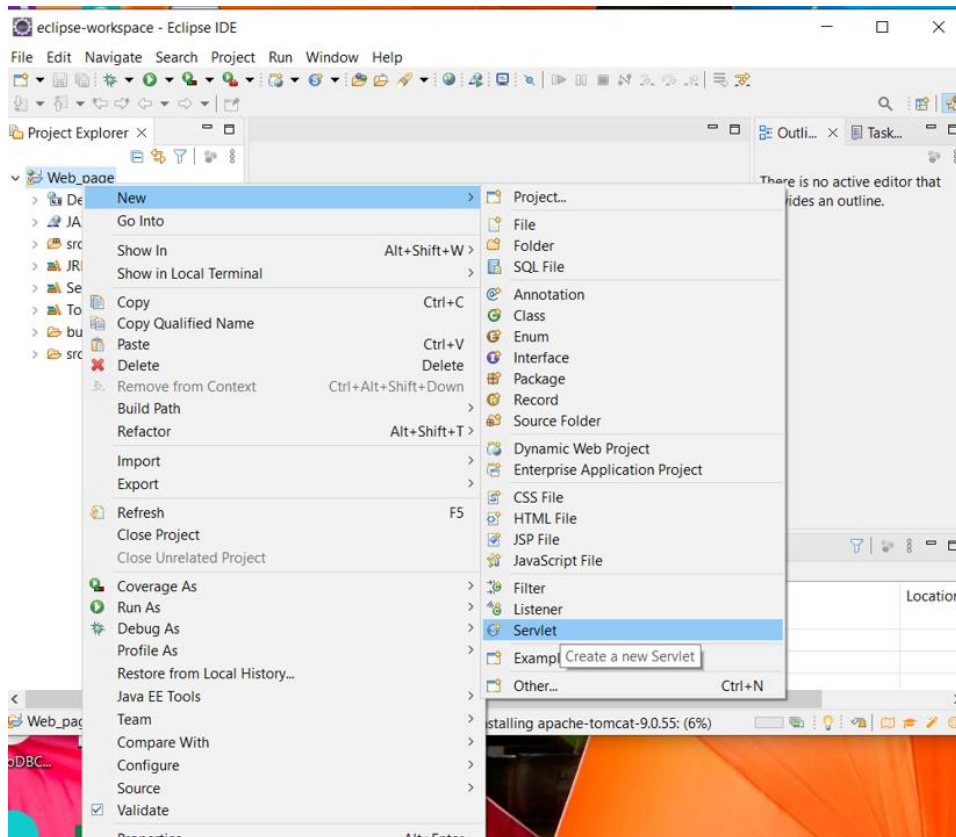


2. Select Properties -> User Library -> Tomcat Library -> Finish

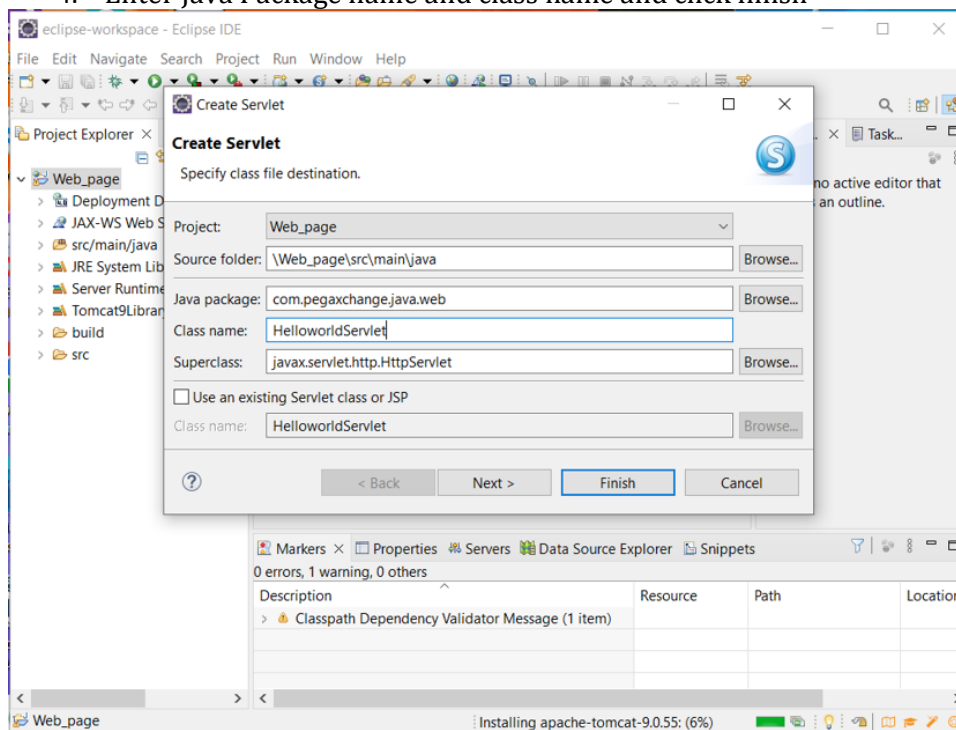




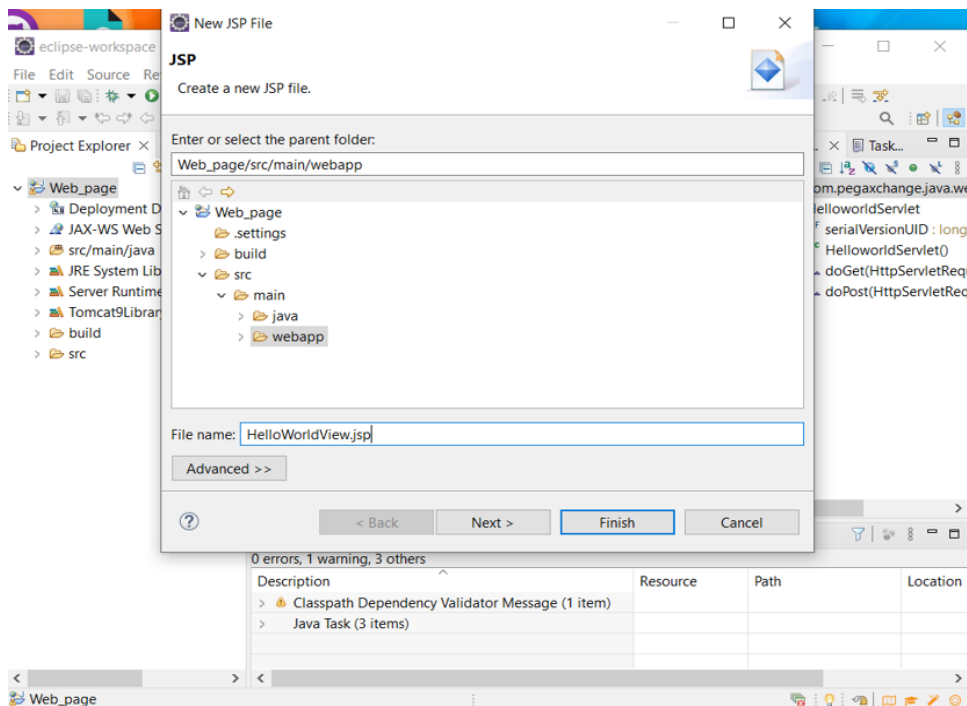
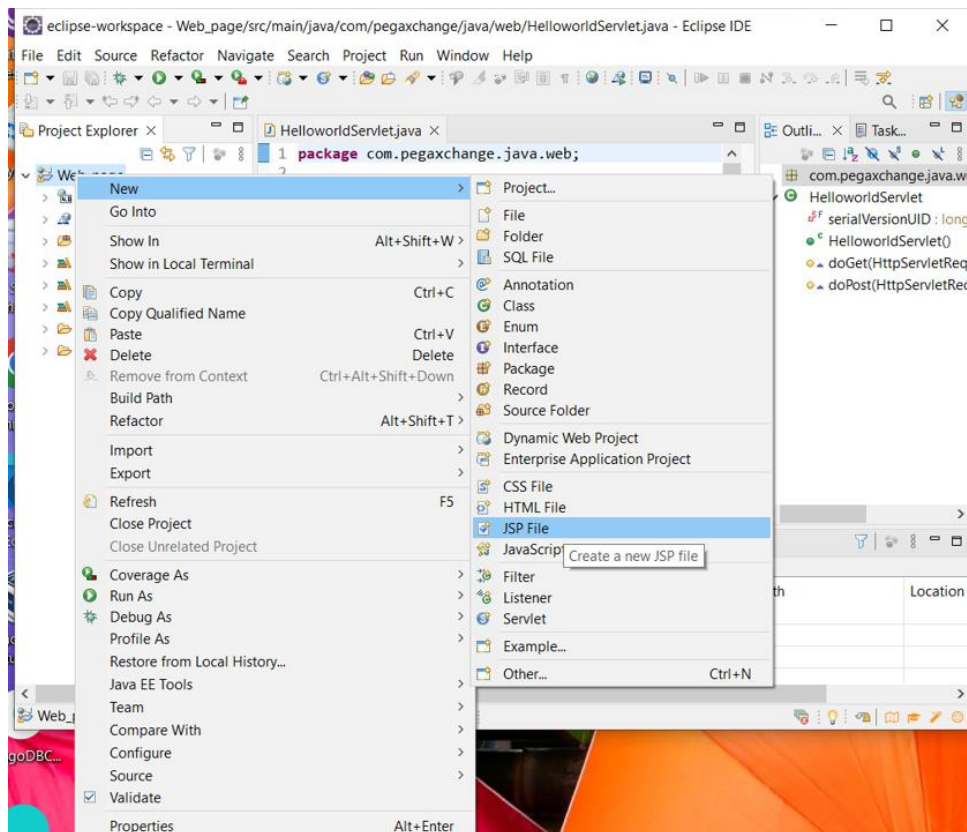
3. Right Click on the project name -> New -> Servlet

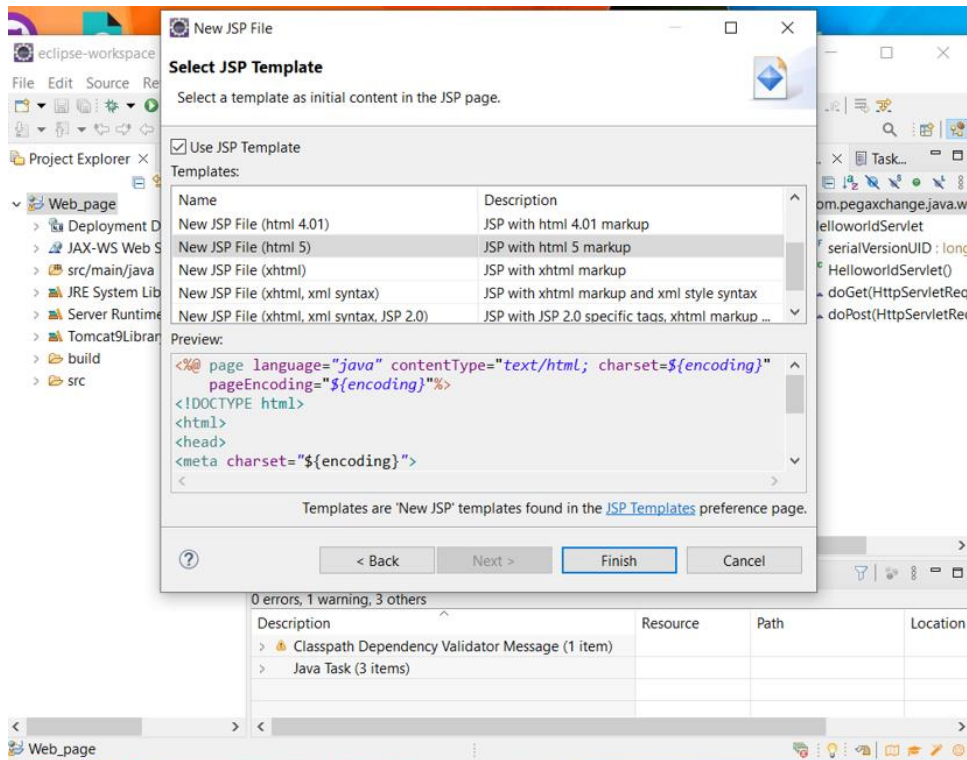


4. Enter Java Package name and class name and click finish

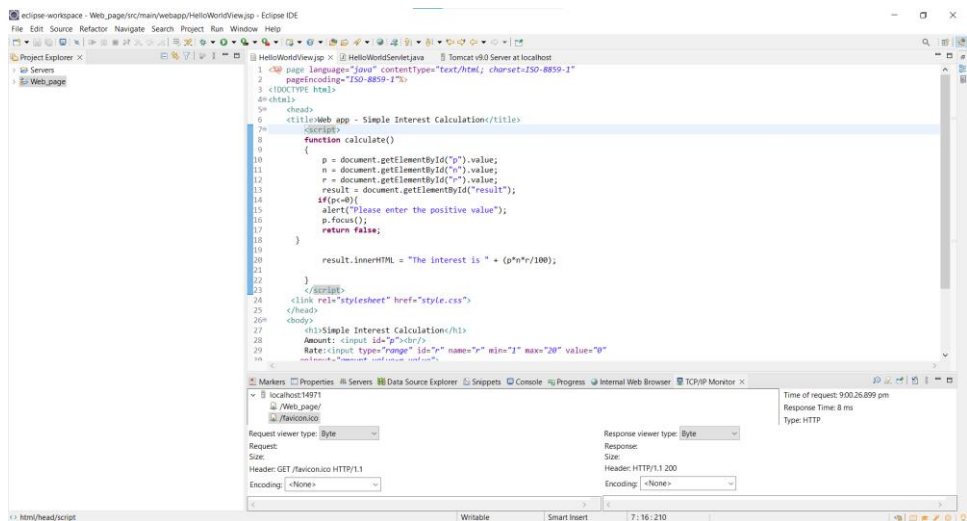


5. Right Click on the project -> New -> JSP File

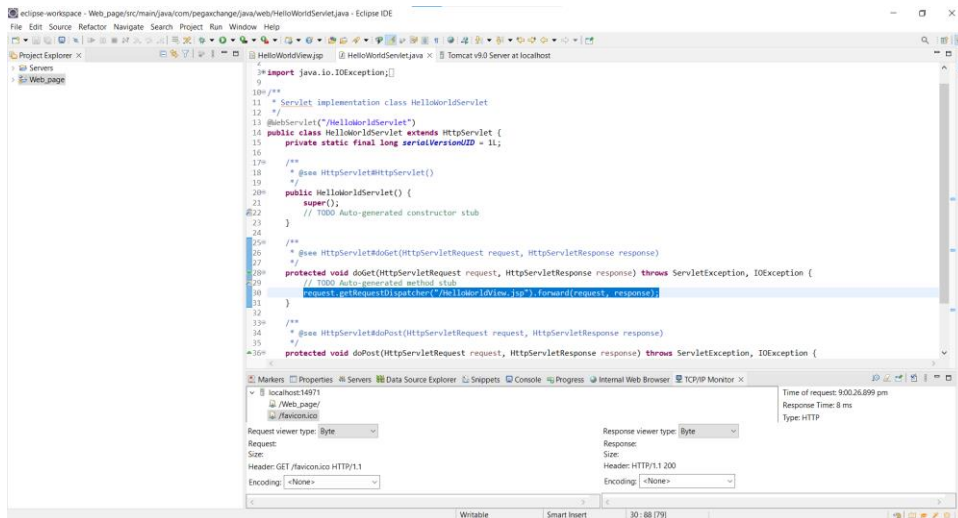




6. Change the default code of HelloWorldServlet for our application

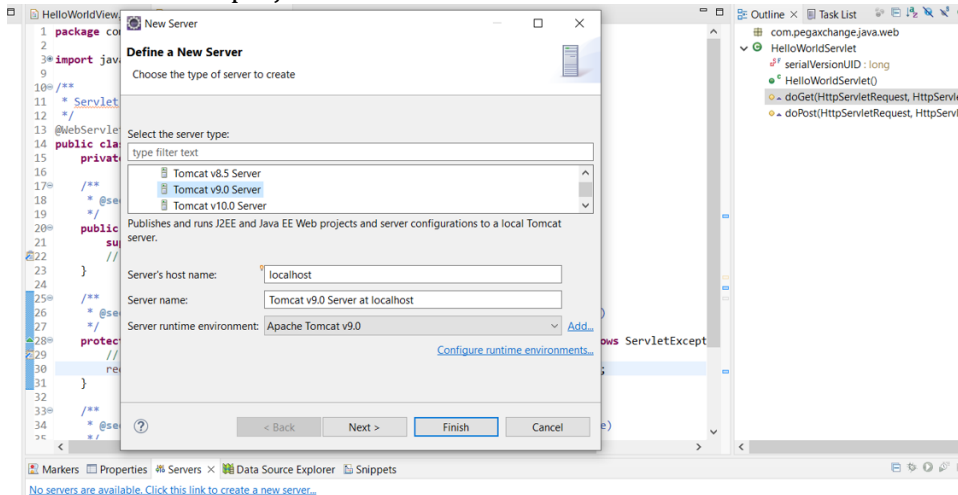


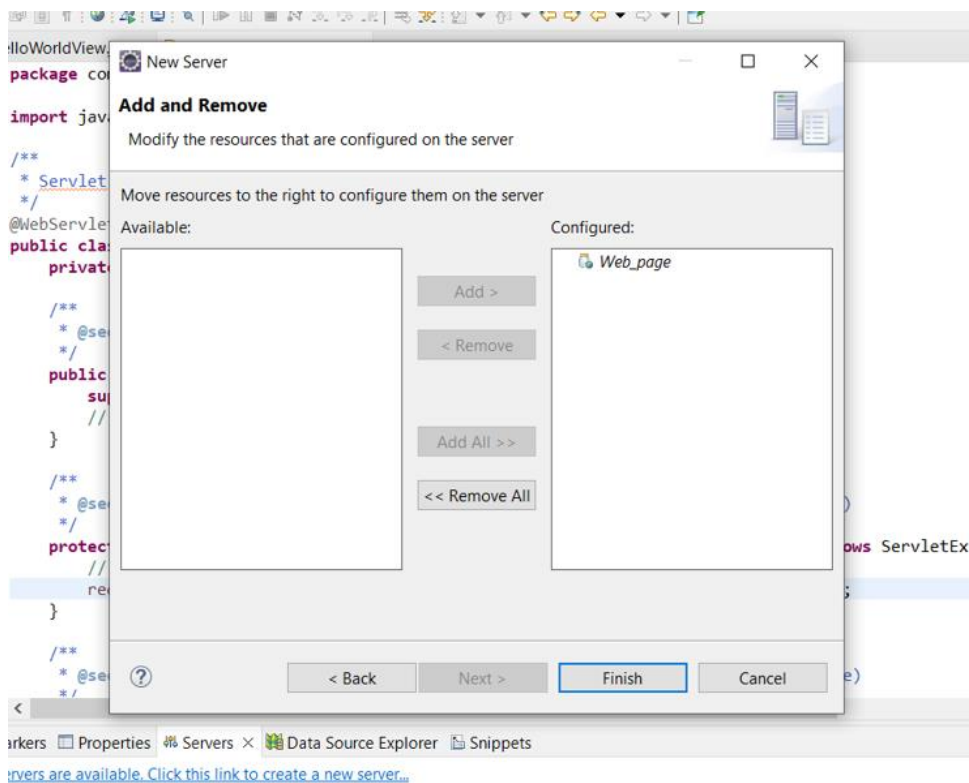
7. Now Change doGet method to request.getRequestDispatcher("/HelloWorldView.jsp").forward(request, response);



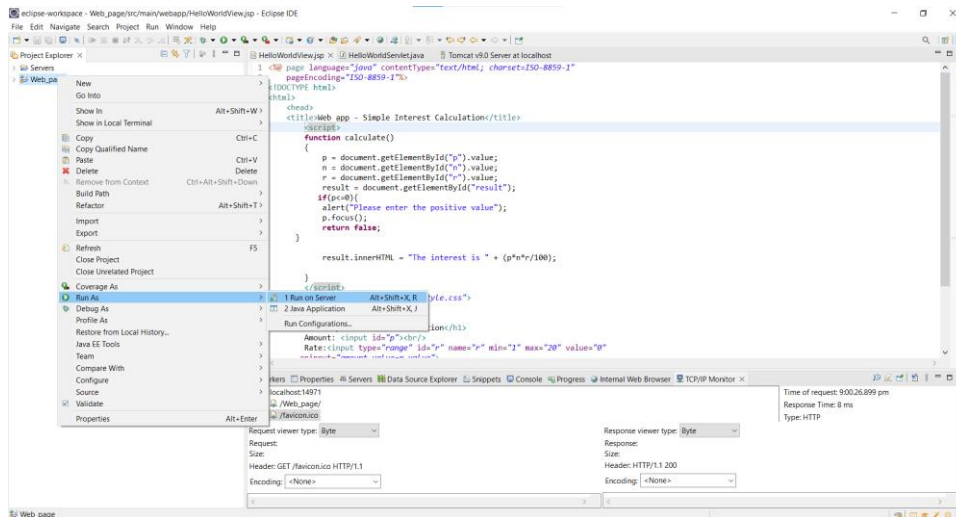
8. Right Click on the Project -> New -> Server -> Next

9. Select the project name under the section and add button





10. Click New -> Run as -> 1 Run on server



**CODE:****HelloWorldView.jsp:**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
    <head>
<title>Web app - Simple Interest Calculation</title>
    <script>
    function calculate()
    {
        p = document.getElementById("p").value;
        n = document.getElementById("n").value;
        r = document.getElementById("r").value;
        result = document.getElementById("result");
        if(p<=0){
            alert("Please enter the positive value");
            p.focus();
            return false;
        }
    }
}
```

```

        result.innerHTML = "The interest is " + (p*n*r/100);
    }
</script>
<link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Simple Interest Calculation</h1>
    Amount: <input id="p"><br/>
    Rate:<input type="range" id="r" name="r" min="1" max="20" value="0"
oninput="amount.value=r.value">
    <output id="amount" name="amount" for="r">0</output><br/>
    No. of Years: <input type="number" id="n" name="n" min="1" max="25"><br/>
    <button onclick="calculate()">Compute</button>
</div>
<p id="result"></p>
</div>
</body>
</html>

```

HelloWorldServlet.java:

```

package com.pegaxchange.java.web;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloWorldServlet
 */
@WebServlet("/HelloWorldServlet")
public class HelloWorldServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloWorldServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        request.getRequestDispatcher("/HelloWorldView.jsp").forward(request, response);}

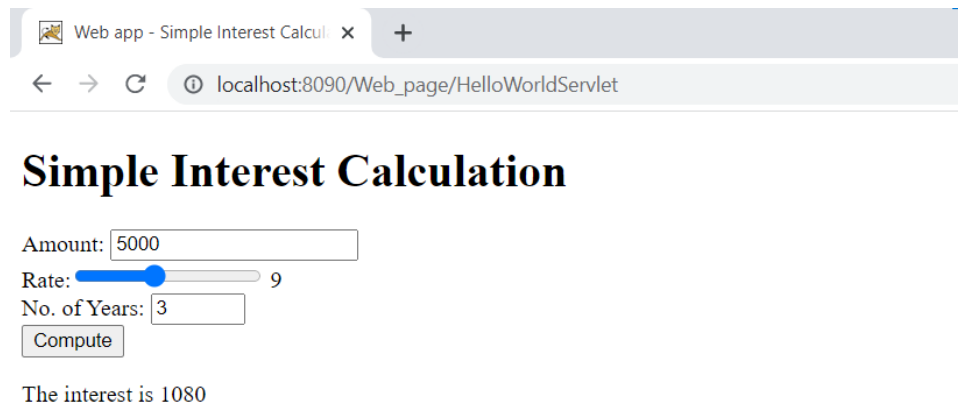
```

```

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
// TODO Auto-generated method stub
doGet(request, response);
}
}

```

SCREENSHOT:



Web app - Simple Interest Calcul x +

localhost:8090/Web_page/HelloWorldServlet

Simple Interest Calculation

Amount:

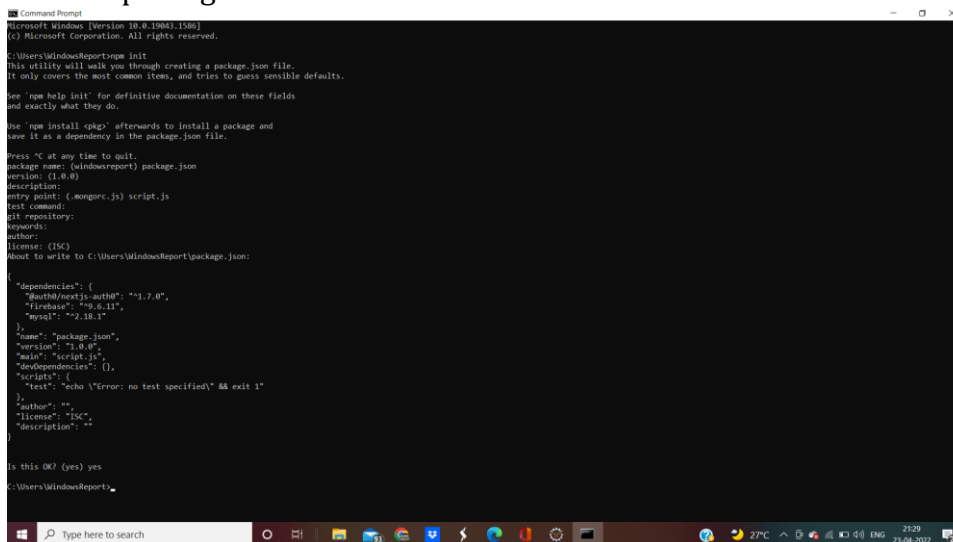
Rate:

No. of Years:

The interest is 1080

RESTFUL STEPS:

1. Create one folder and get it in cmd prompt
2. Install npm packages
 - a. npm init
 - b. npm i express
 - c. npm i -g nodemon



```

Microsoft Windows [version 10.0.19043.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\WindowsReport>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

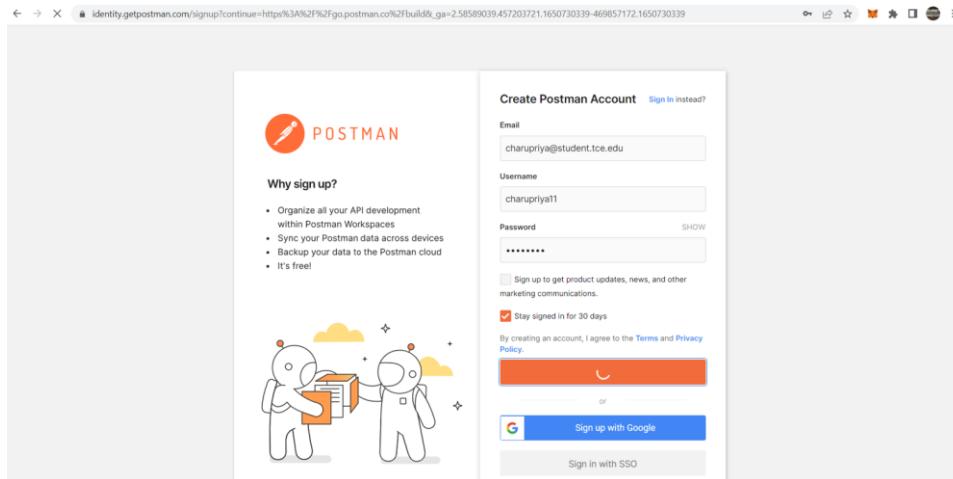
Press ^C at any time to quit.
package name: (WindowsReport) package.json
version: (1.0.0)
description:
entry point: (.mainrc.js) script.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\WindowsReport\package.json:

{
  "dependencies": {
    "bcryptjs": "^2.9.0",
    "firebase": "^9.6.11",
    "mysql": "^2.18.1"
  },
  "name": "package.json",
  "version": "1.0.0",
  "main": "script.js",
  "devDependencies": {},
  "scripts": {
    "test": "echo \"Error: no test specified\" & exit 1"
  },
  "author": "",
  "license": "ISC",
  "description": ""
}

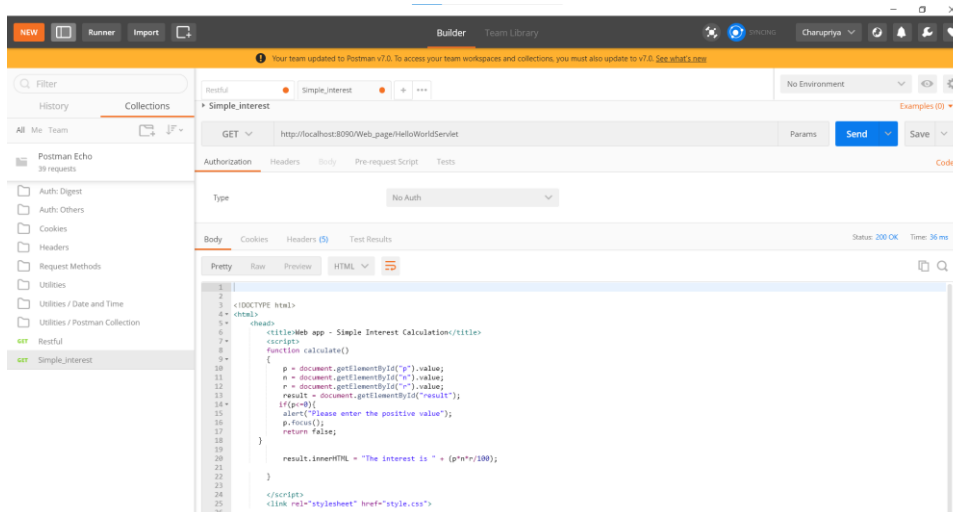
Is this OK? (yes) yes
C:\Users\WindowsReport>

```

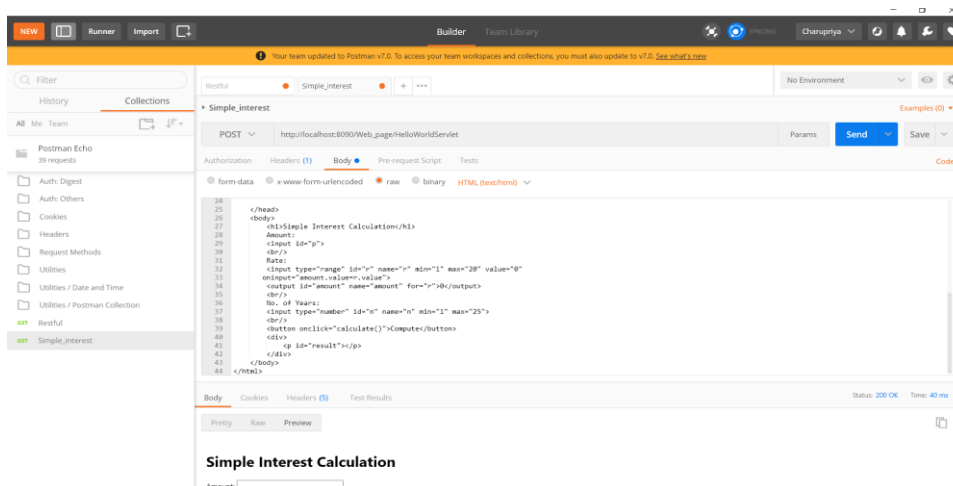
3. Install Postman in Chrome Extension



4. Click Get request and enter URL



5. Select POST and click body -> raw -> JSON



RESULT:

Thus, implementation of Soap and Restful web services in JAVA is verified