Lesson 01 Demo 01

Develop a SOAP Web Service

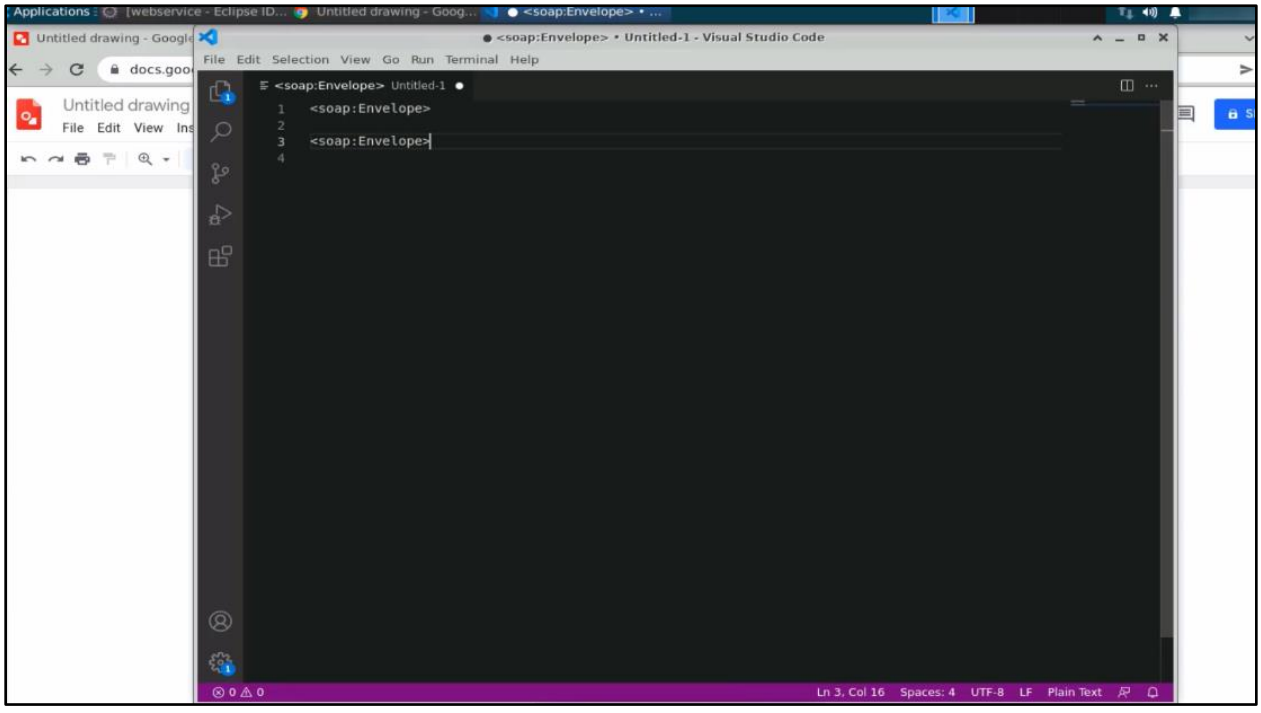
Objective: To develop a SOAP web service for user details through Java objects
Tools required: VS Code and Eclipse IDE
Pre-requisites: None

Steps to be followed:

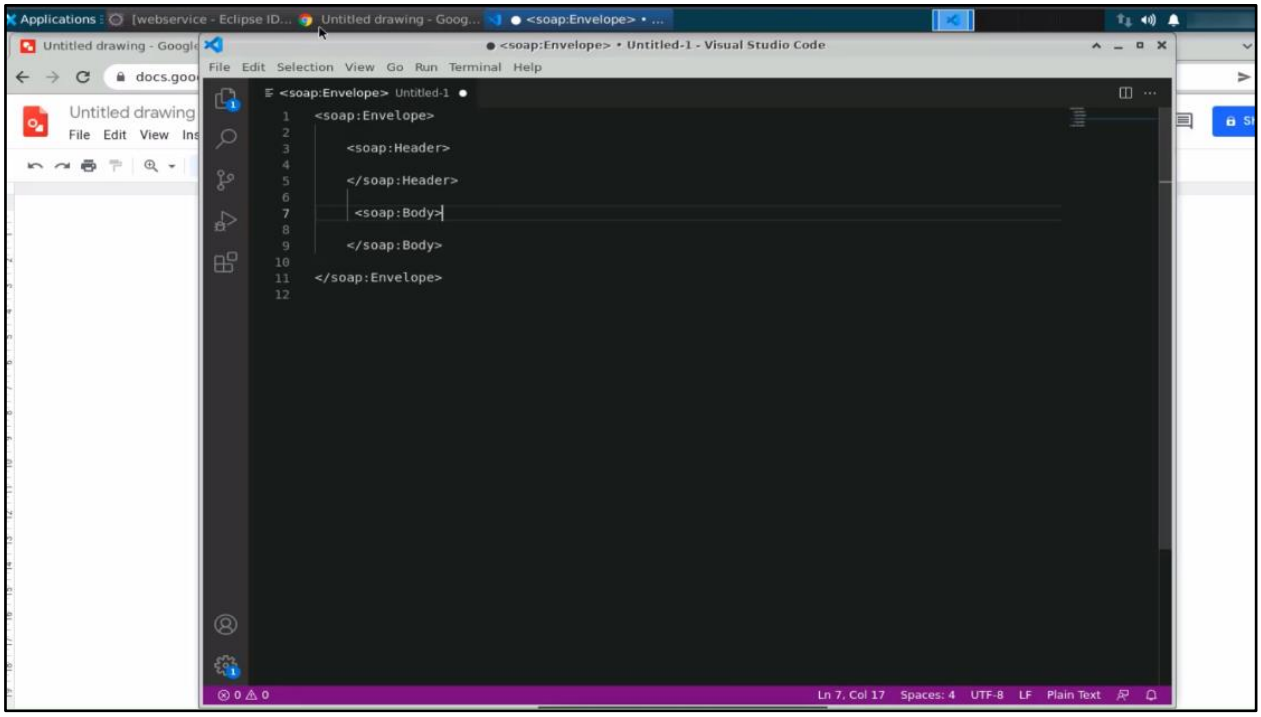
1. Creating model components
2. Creating user service components

Step 1: Creating model components

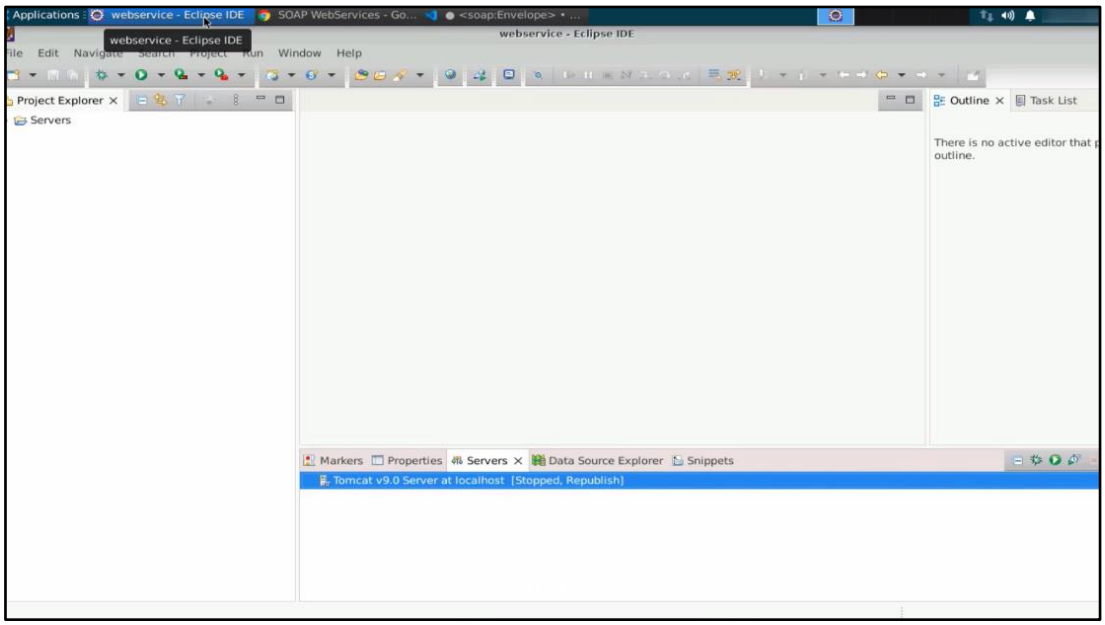
1.1 Launch **Visual Studio Code** from the desktop and create a file that contains a SOAP envelope as the root tag, as shown in the screenshot below:



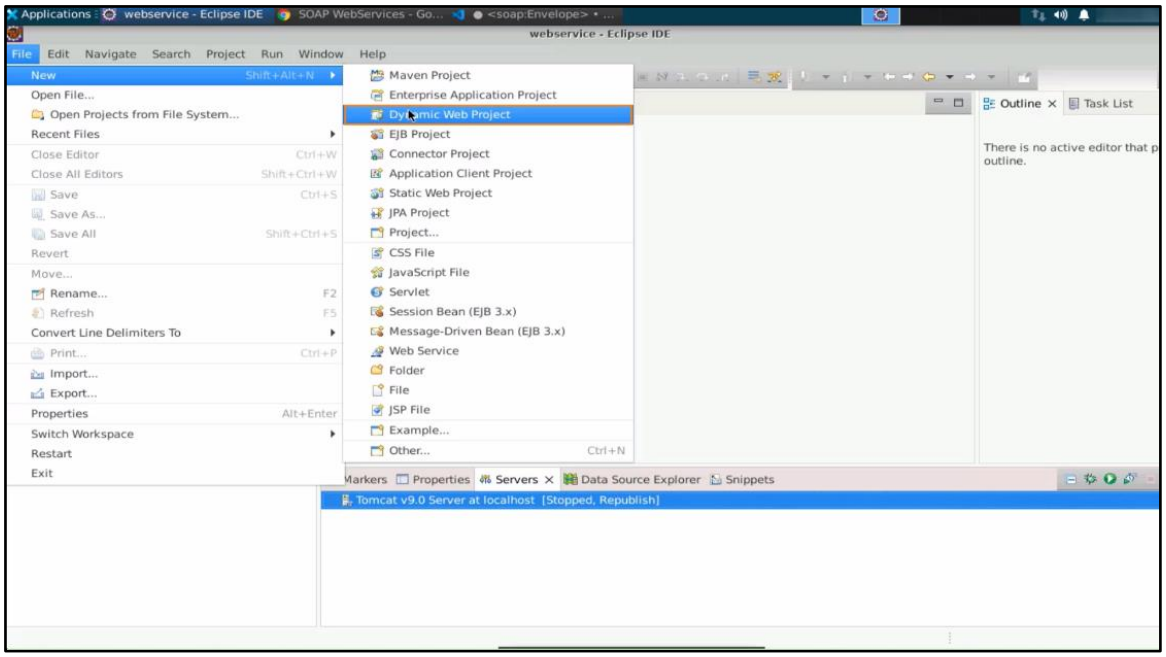
1.2 Add the **soap:Header** and **soap:Body** tags in the envelope as shown in the screenshot and save the file:



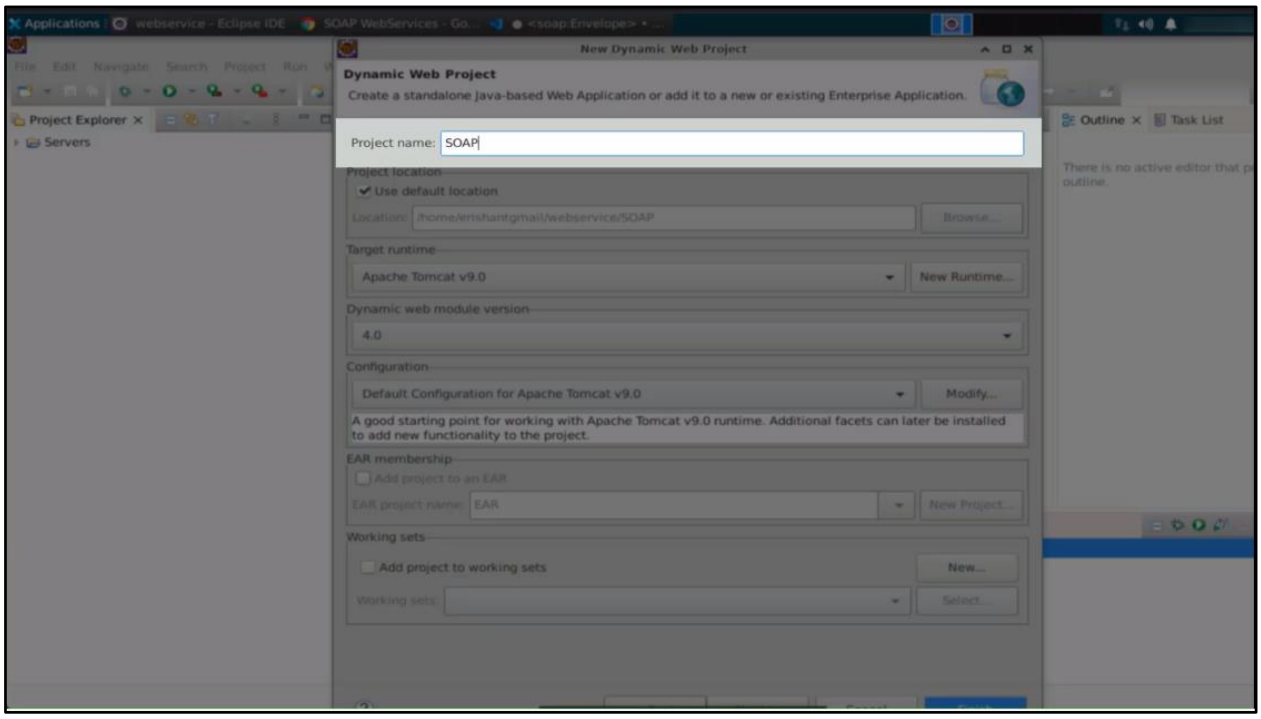
1.3 Open **Eclipse IDE** from the desktop of your lab



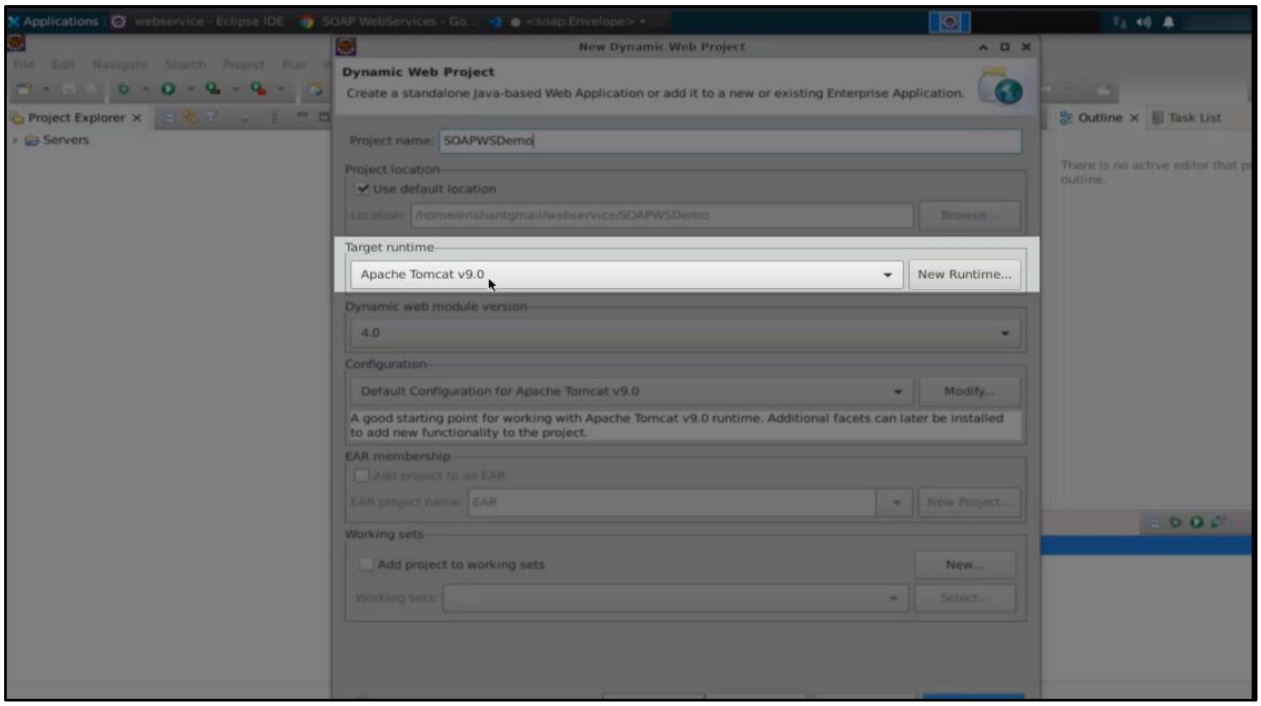
1.4 Click on **File** -> **New** -> **Dynamic Web Project,** as shown in the screenshot below:



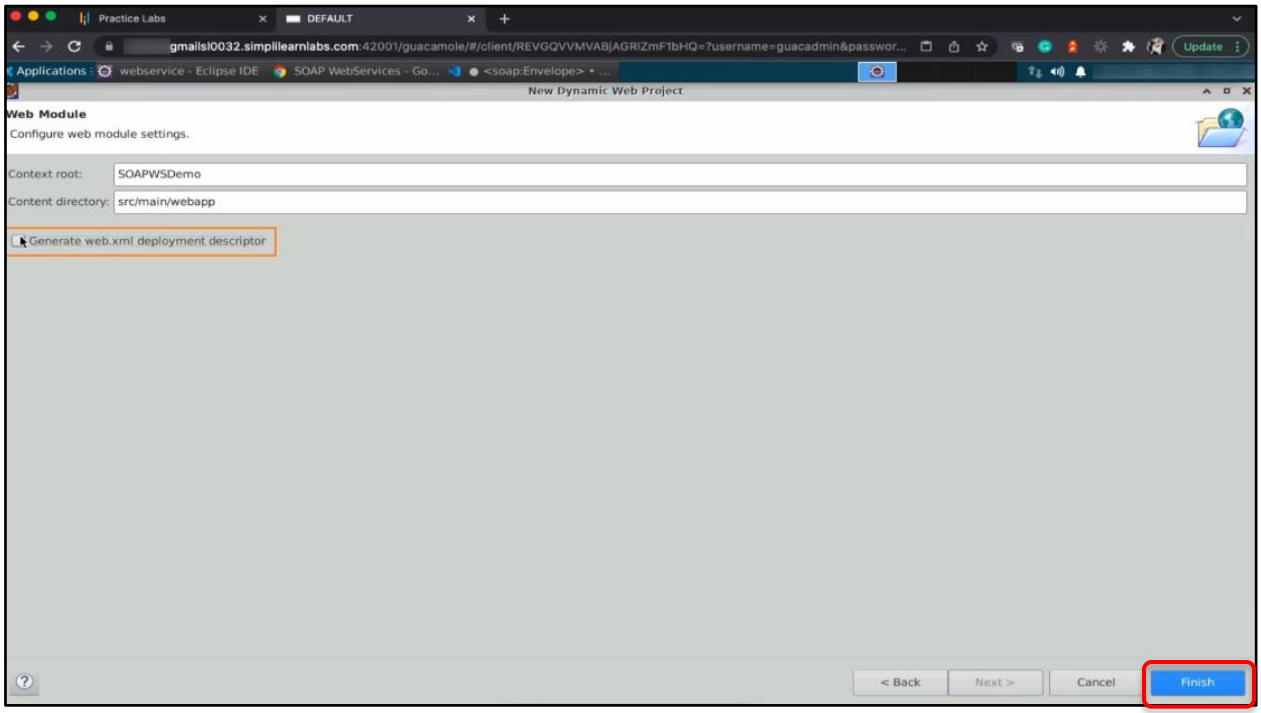
1.5 Provide a name of the project as **SOAP**



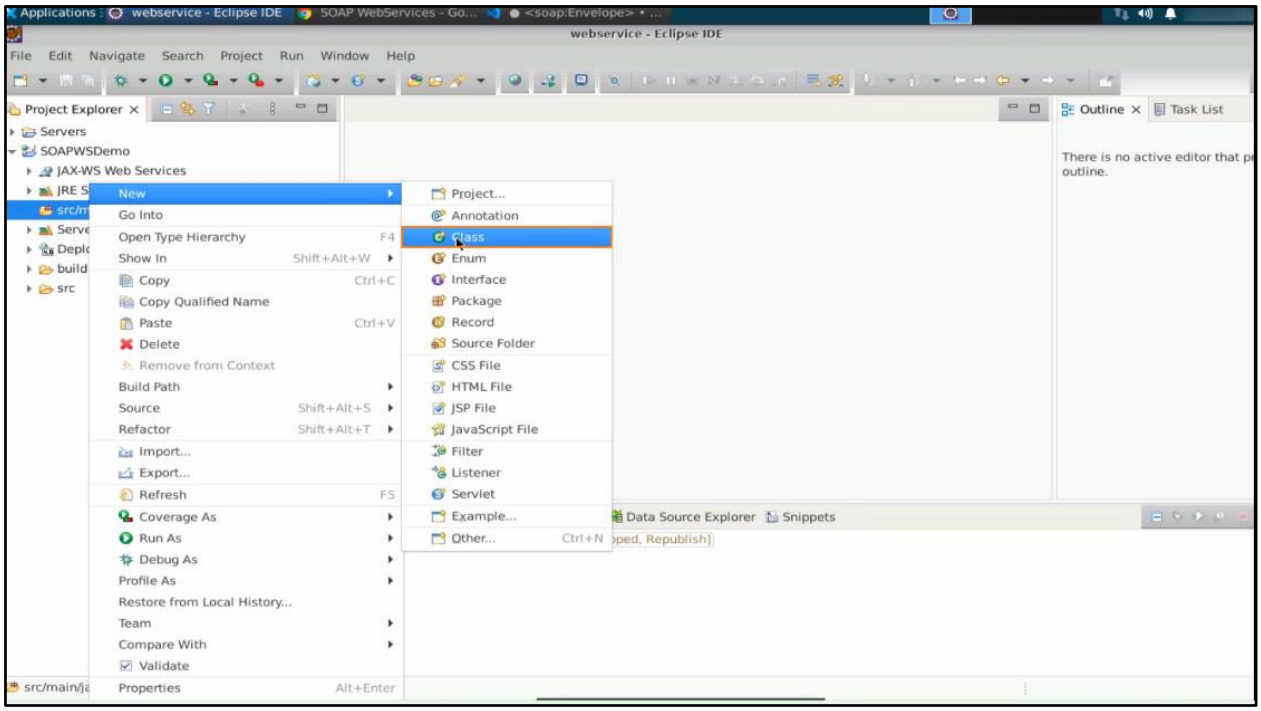
1.6 Select the Tomcat version as **Apache Tomcat v9.0** and click **Next**



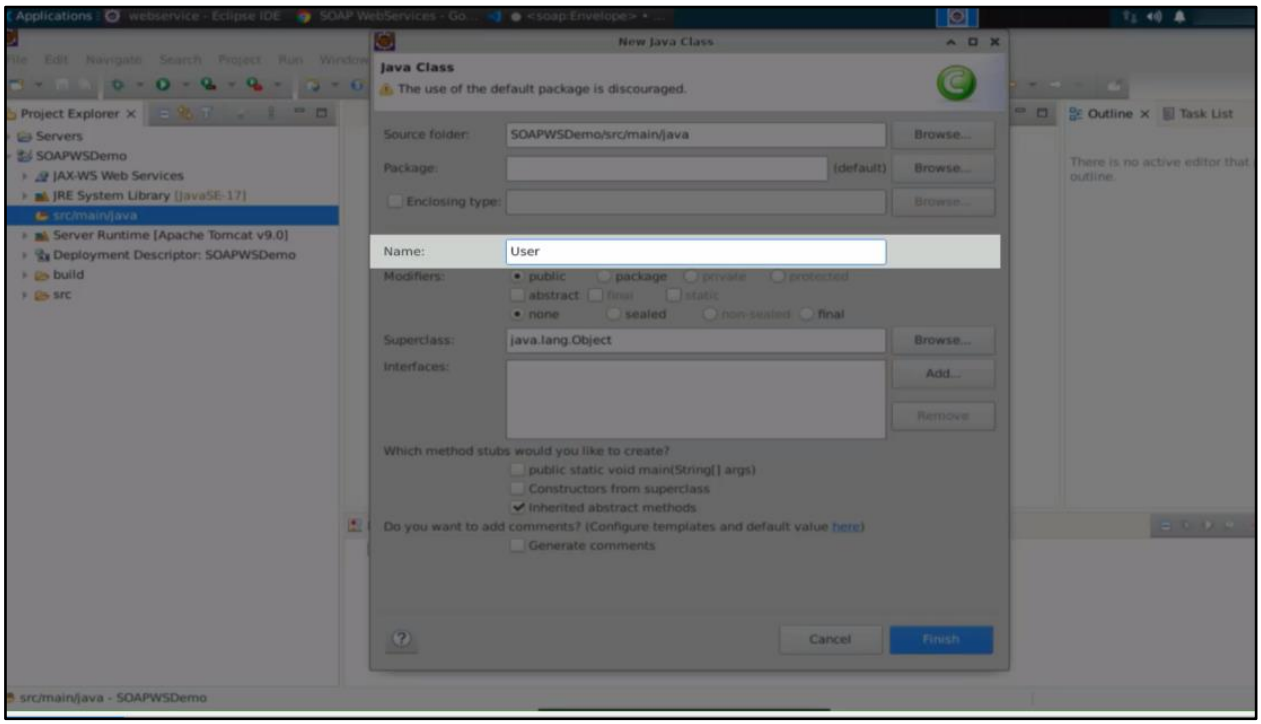
1.7 Check the **Generate web.xml deployment descriptor** box and click **Finish**



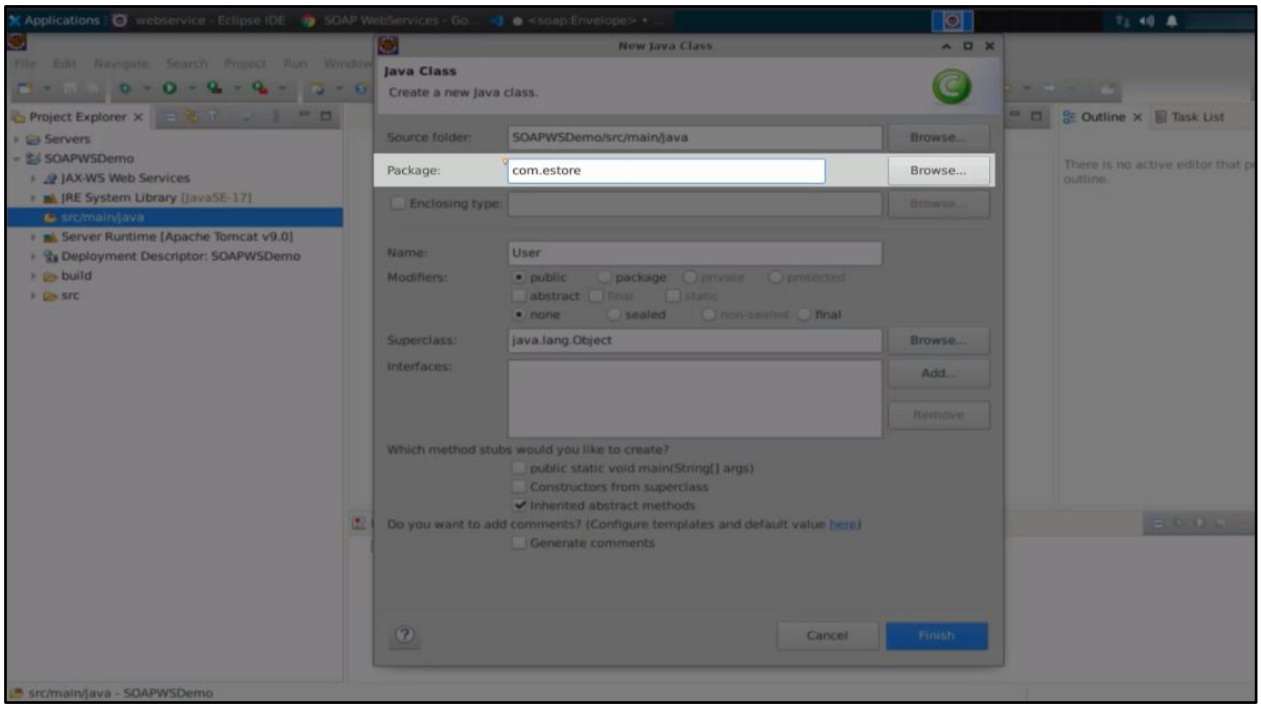
1.8 Right-click on the project and select **New -> Class,** as shown below:



1.9 Provide a **Name** to the class as **User**



1.10 Give the package name as **com.estore,** as shown below:

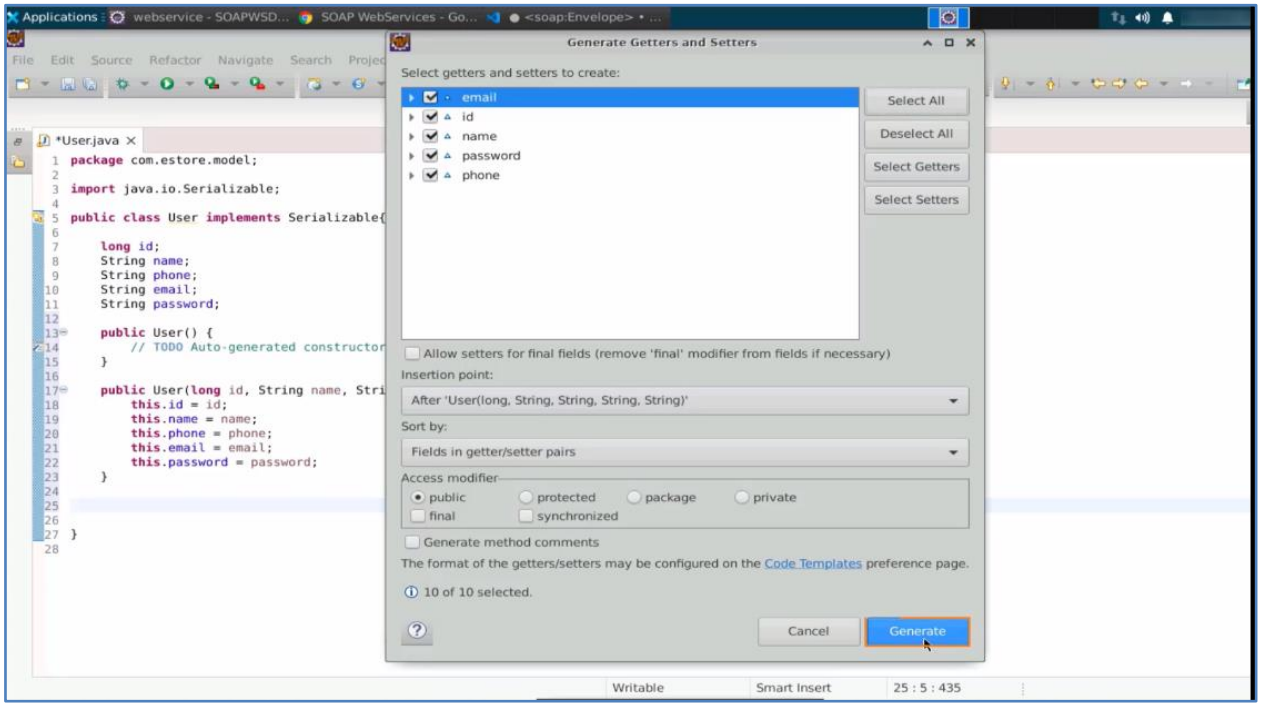


1.11 Write the code given below to the **User.java** file:

```
package com.estore.model;

import java.io.Serializable;

public class User implements Serializable{

    long id;
    String name;
    String phone;
    String email;
    String password;

    public User() {
            // TODO Auto-generated constructor stub
    }
```

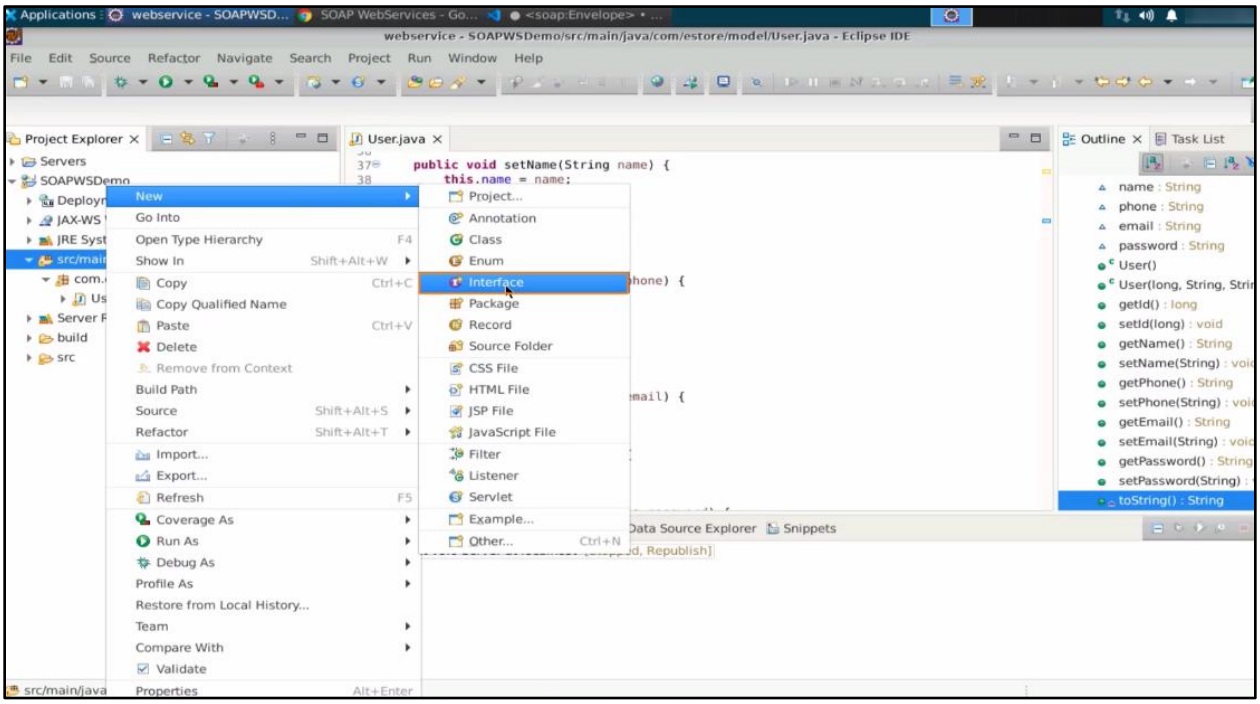
```
public User(long id, String name, String phone, String email, String password) {
        this.id = id;
        this.name = name;
        this.phone = phone;
        this.email = email;
        this.password = password;
}

public long getId() {
        return id;
}

public void setId(long id) {
        this.id = id;
}

public String getName() {
        return name;
}

public void setName(String name) {
        this.name = name;
}

public String getPhone() {
        return phone;
}

public void setPhone(String phone) {
        this.phone = phone;
}

public String getEmail() {
        return email;
}

public void setEmail(String email) {
        this.email = email;
```

```
    }

    public String getPassword() {
            return password;
    }

    public void setPassword(String password) {
            this.password = password;
    }

    @Override
    public String toString() {
            return "User [id=" + id + ", name=" + name + ", phone=" + phone + ", email=" +
email + ", password=" + password
                            + "]";
    }

}
```

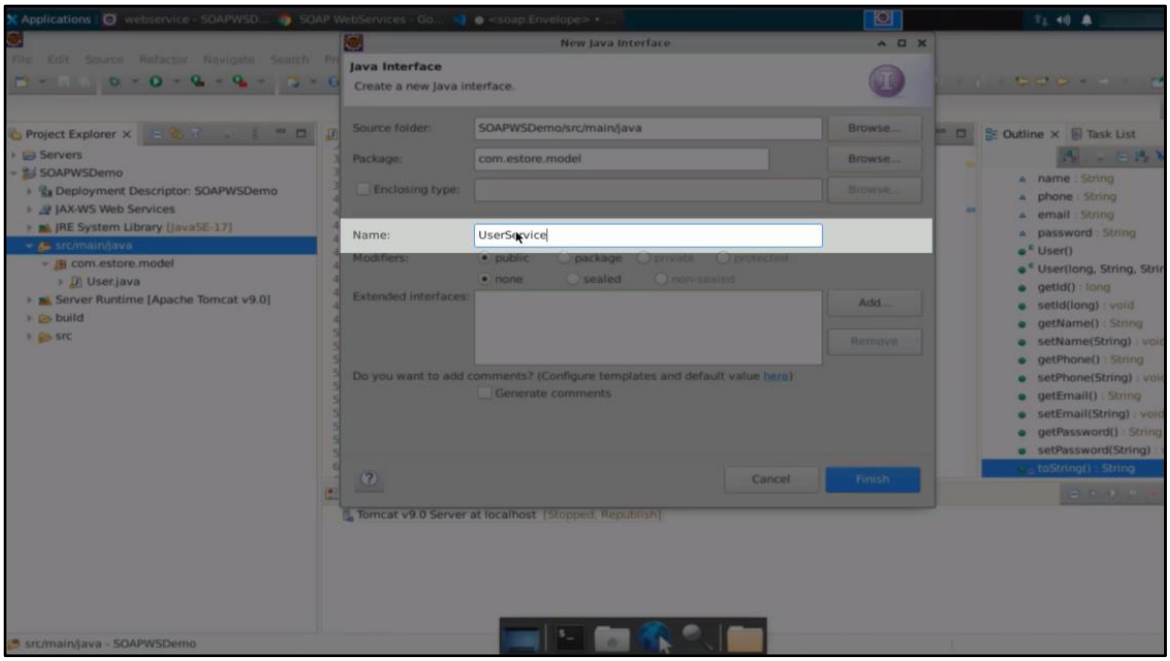


Step 2: Creating user service components

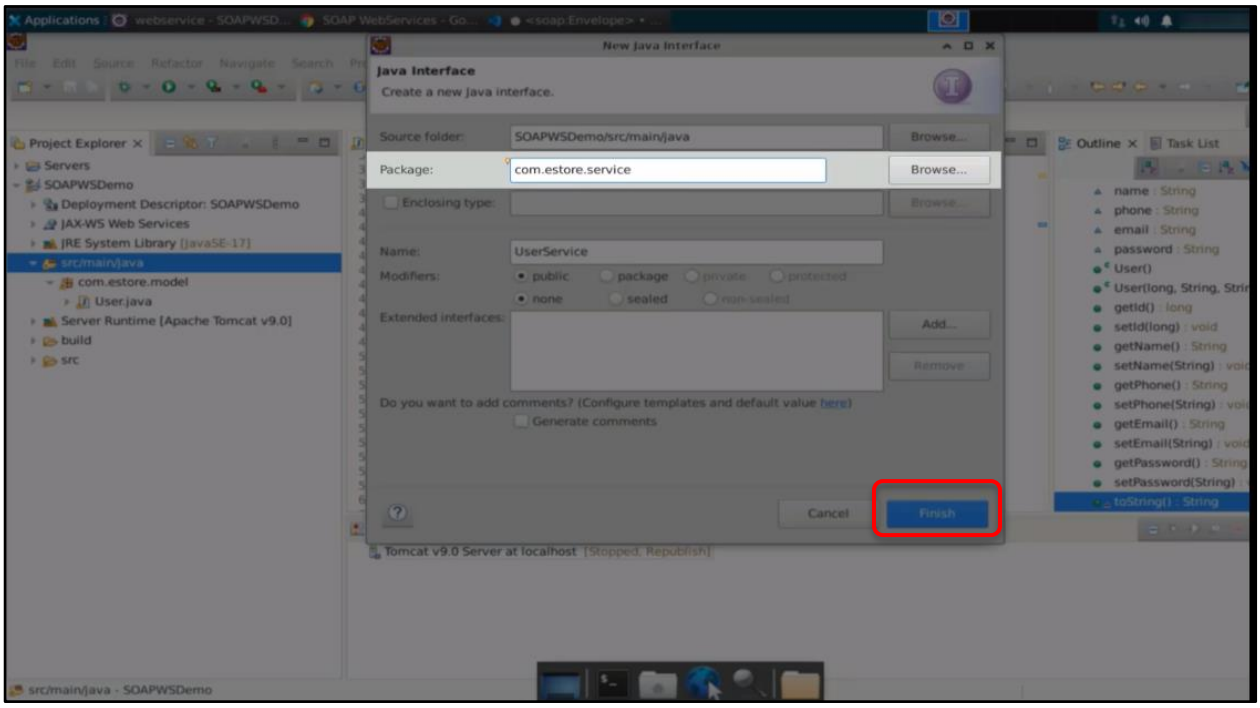
2.1 Right-click on the **src** folder and click **New -> Interface**



2.2 Provide the name of the Java Interface as **UserService** and click **Finish**

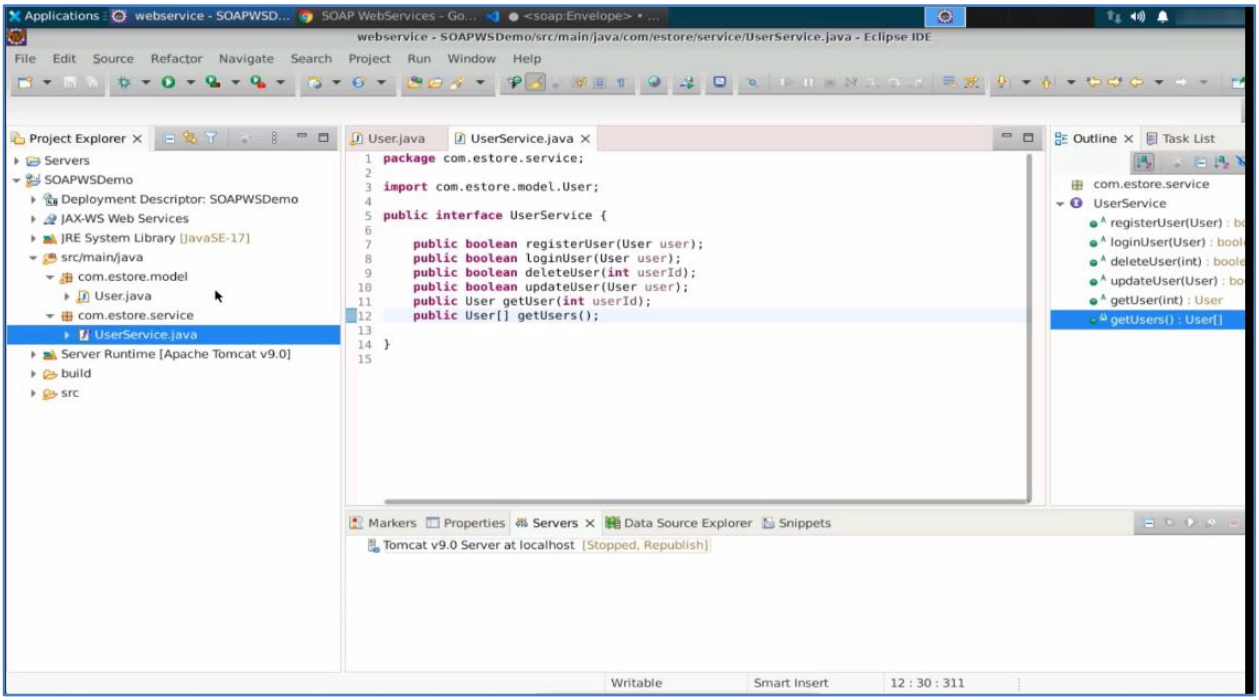


2.3 Provide the package name as **com.estore.service** and click **Finish**

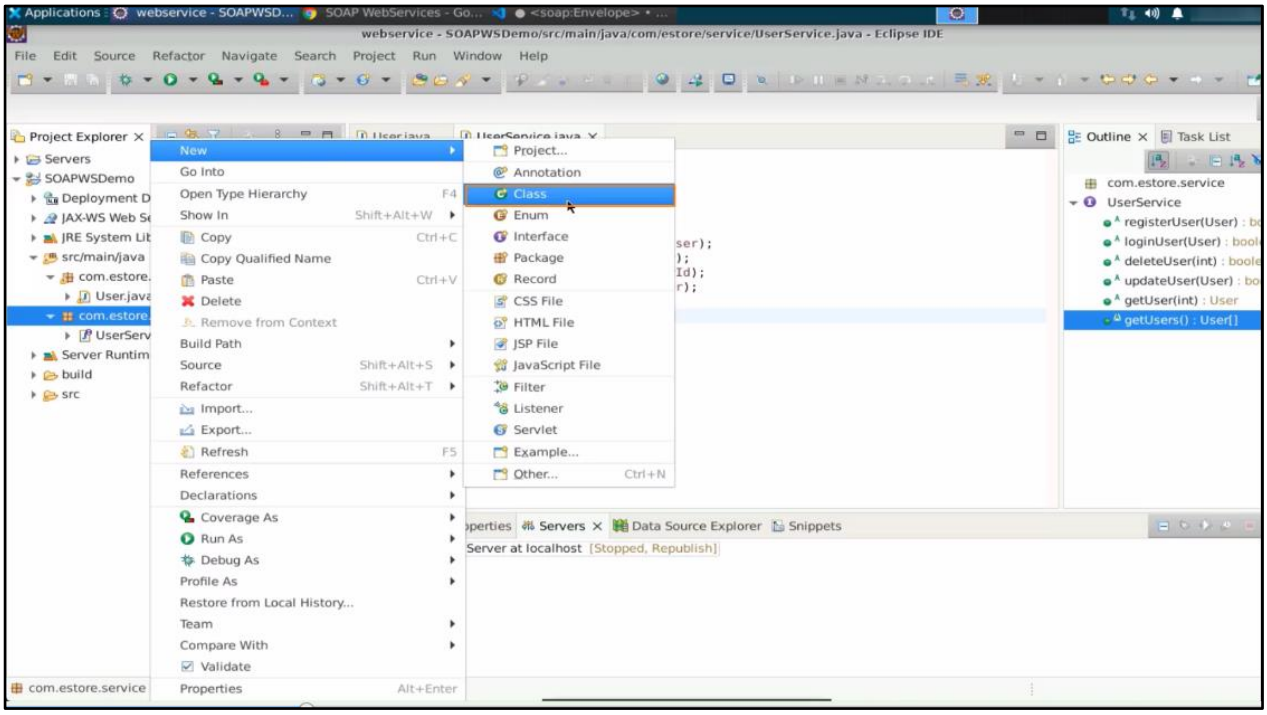


2.4 Add the code given below to the **UserService.java** file:

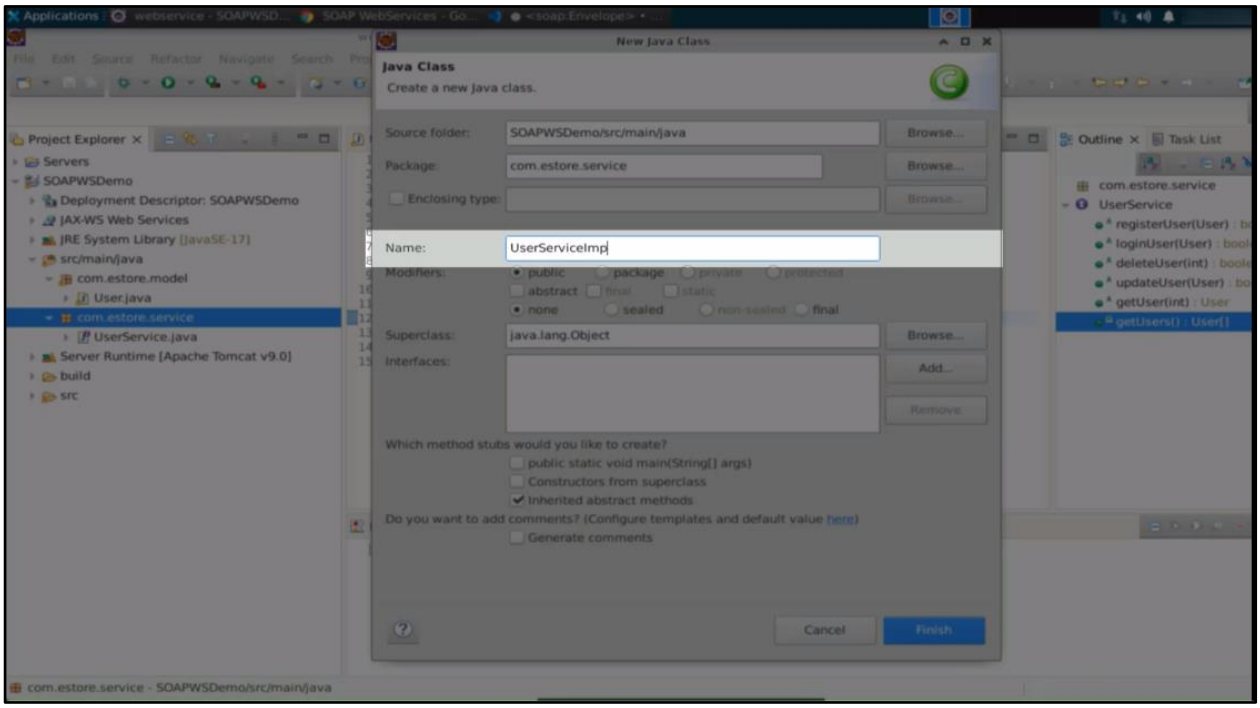
```
package com.estore.service;

import com.estore.model.User;

public interface UserService {

        public boolean registerUser(User user);
        public boolean loginUser(User user);
        public boolean deleteUser(String userEmail);
        public boolean updateUser(User user);
        public User getUser(String userEmail);
        public User[] getUsers();

}
```



2.5 Right-click on the package name and click **New -> Class**



2.6 Provide a name for the Java Class as **UserServiceImpl**

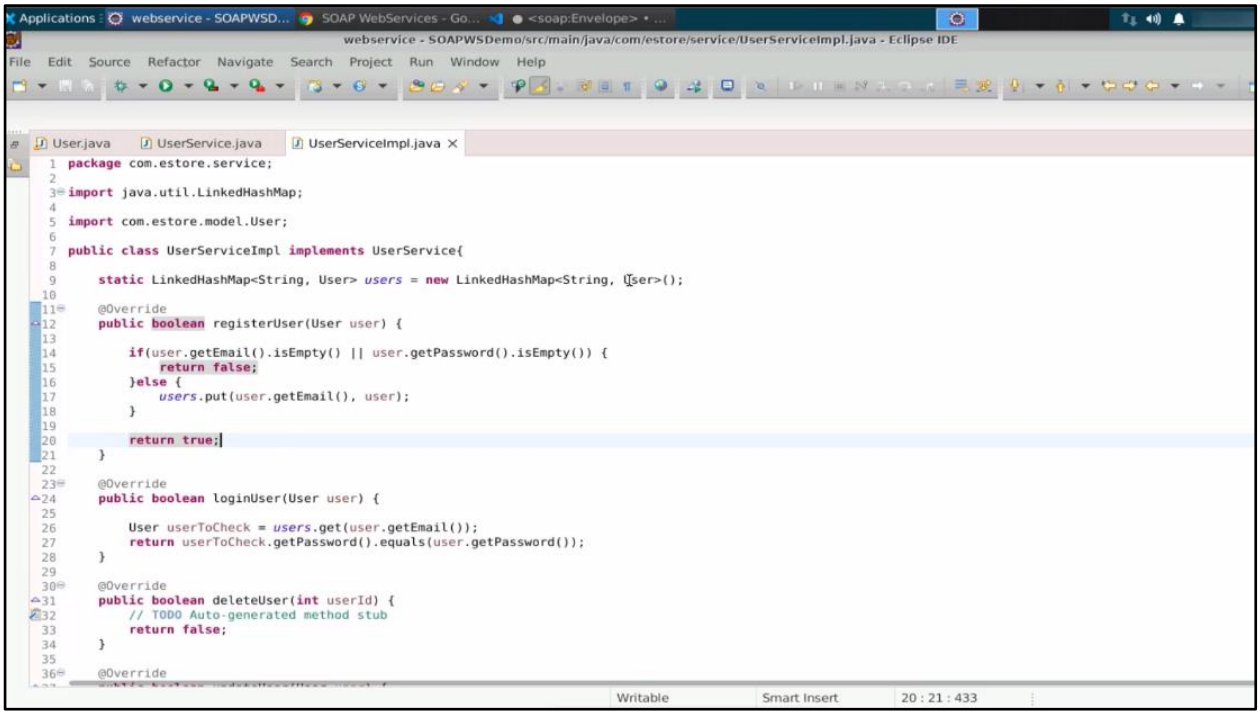


2.7 Add the code given below in **UserServiceImpl.java:**

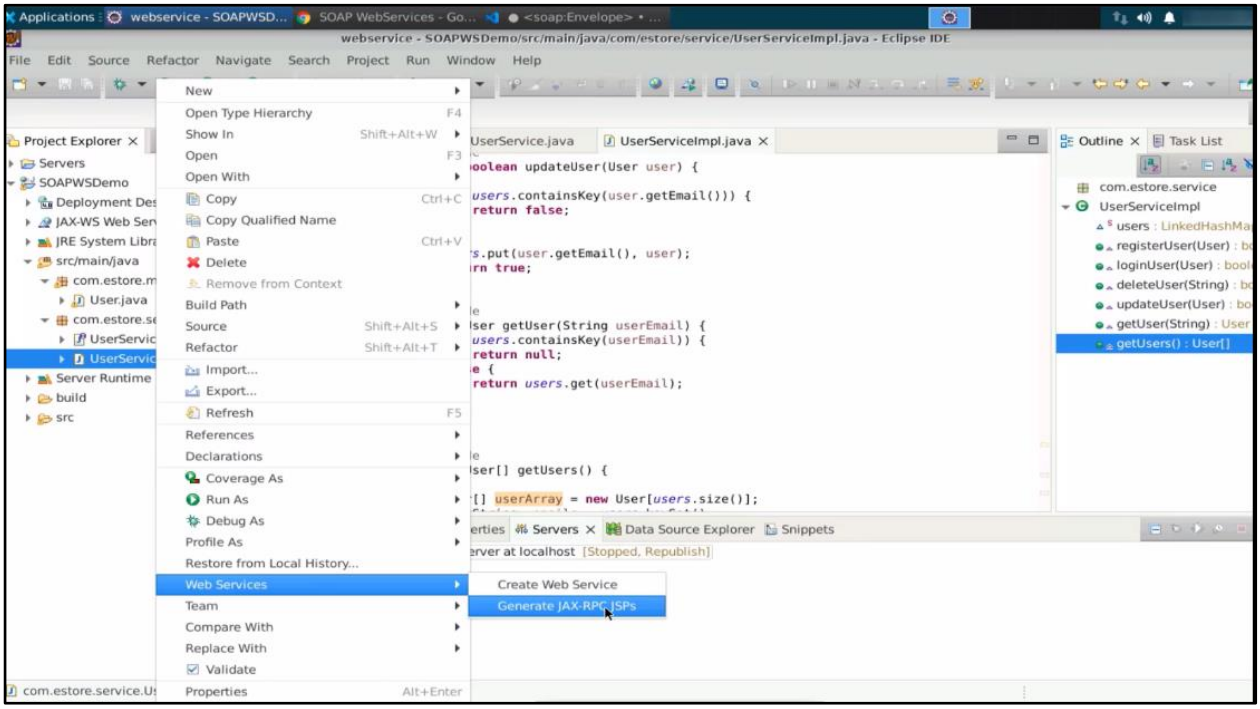
```
package com.estore.service;

import java.util.Iterator;
import java.util.LinkedHashMap;
import java.util.Set;

import com.estore.model.User;

public class UserServiceImpl implements UserService{

    static LinkedHashMap<String, User> users = new LinkedHashMap<String,
User>();

    @Override
    public boolean registerUser(User user) {
```

```
        if(user.getEmail().isEmpty() || user.getPassword().isEmpty()) {
                return false;
        }else {
                users.put(user.getEmail(), user);
        }

        return true;
}

@Override
public boolean loginUser(User user) {

        User userToCheck = users.get(user.getEmail());
        return userToCheck.getPassword().equals(user.getPassword());
}

@Override
public boolean deleteUser(String userEmail) {

        if(!users.containsKey(userEmail)) {
                return false;
        }
        users.remove(userEmail);
        return true;
}

@Override
public boolean updateUser(User user) {

        if(!users.containsKey(user.getEmail())) {
                return false;
        }

        users.put(user.getEmail(), user);
        return true;
}

@Override
```

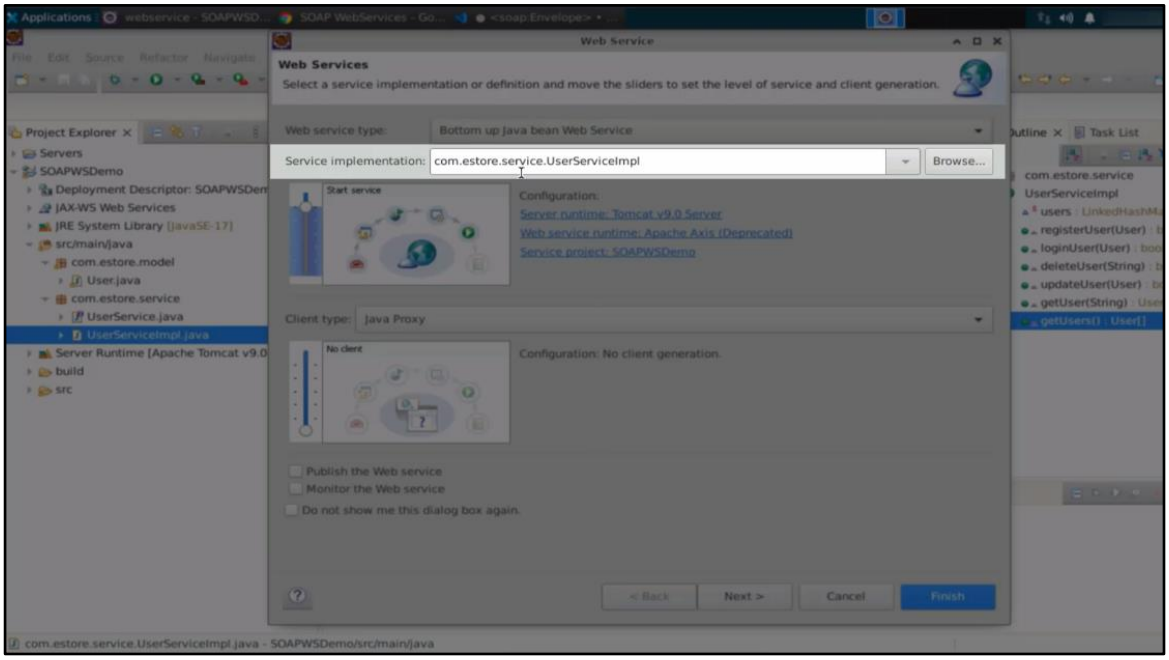
```
    public User getUser(String userEmail) {
        if(!users.containsKey(userEmail)) {
            return null;
        }else {
            return users.get(userEmail);
        }

    }

    @Override
    public User[] getUsers() {

        User[] userArray = new User[users.size()];
        Set<String> emails = users.keySet();

        int idx = 0;
        Iterator<String> itr = emails.iterator();
        while(itr.hasNext()) {
            String email = itr.next();
            userArray[idx] = users.get(email);
            idx++;
        }

        return userArray;
    }

}
```



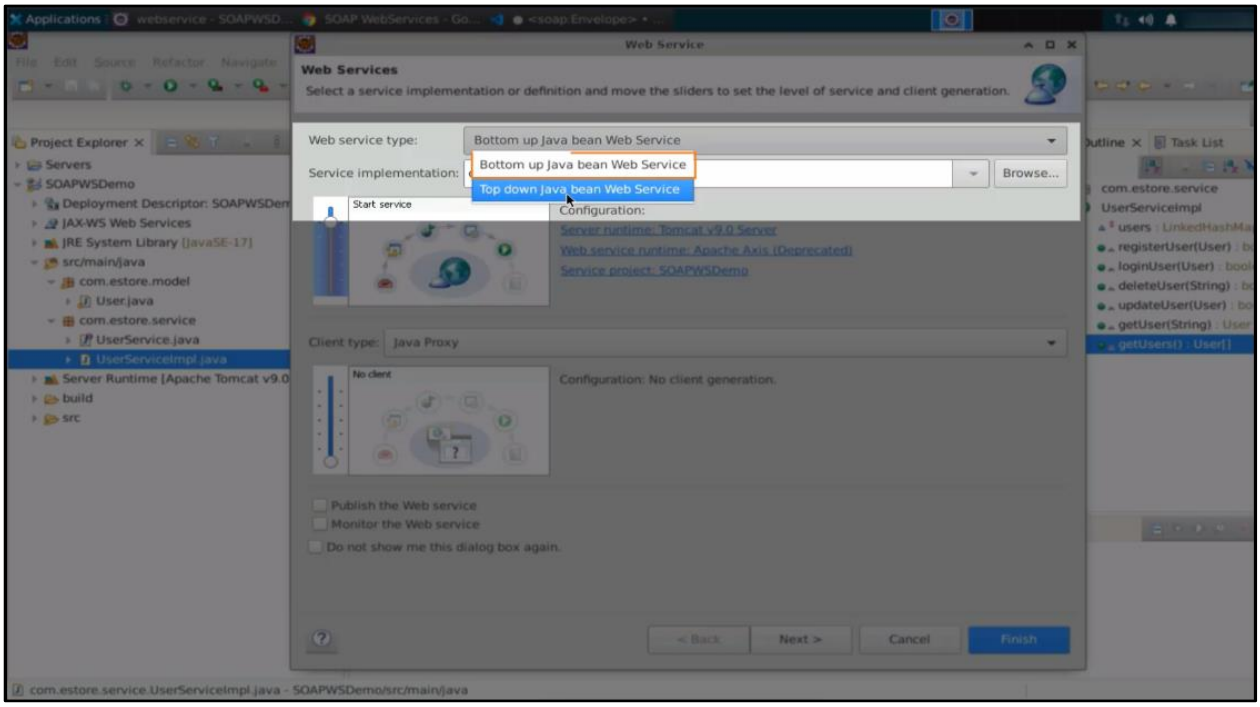
2.8 Right-click on the service file and select **Web Services -> Generate JAX RPC JSPs**



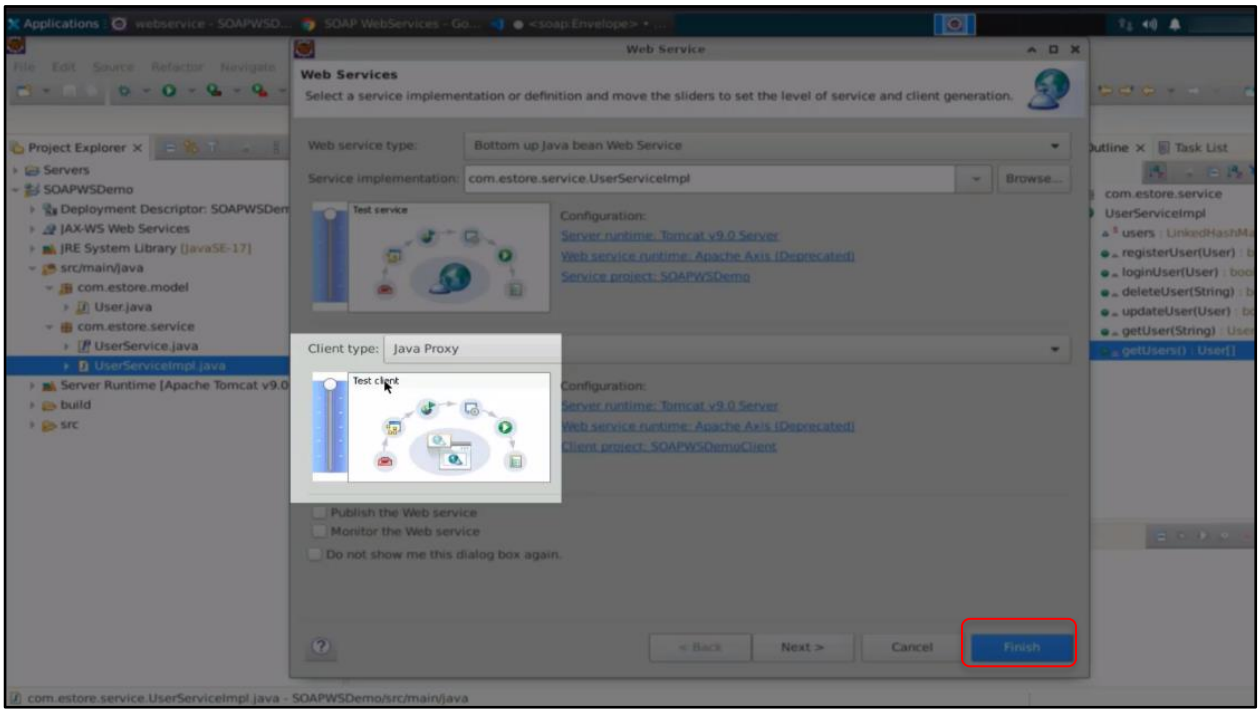
2.9 Select the **UserServiceImpl.java** file under **Service Implementation** and name it



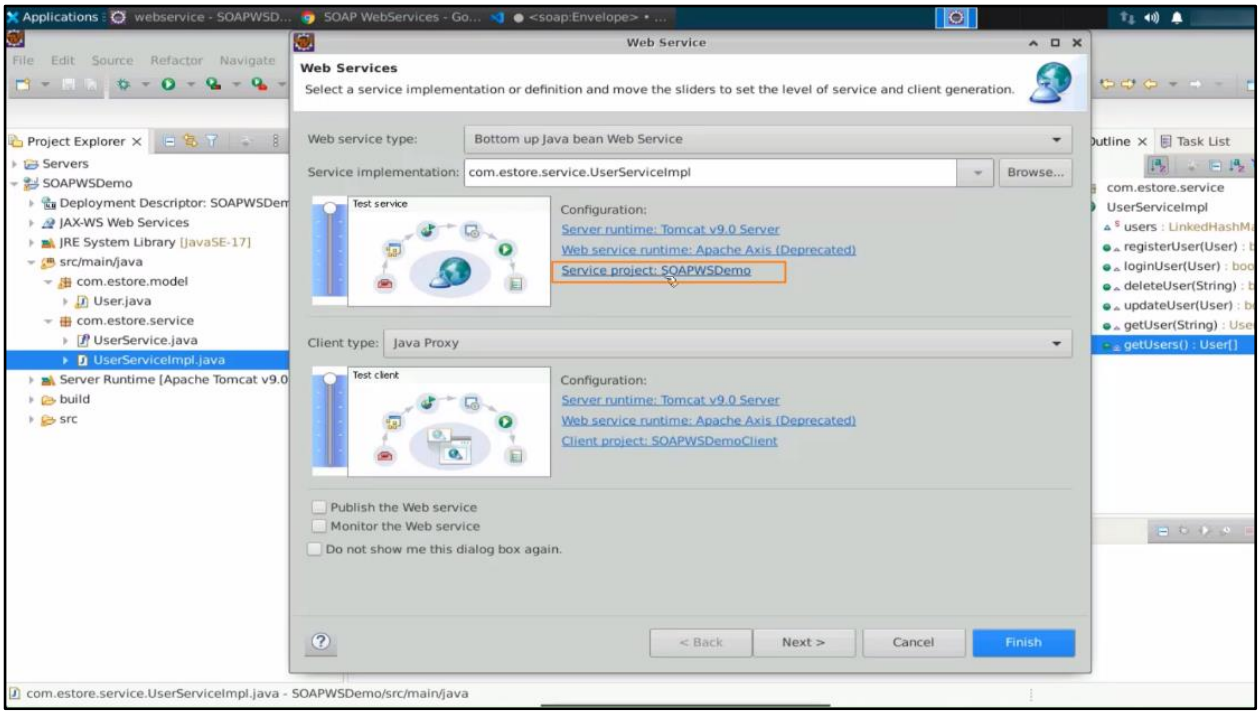
2.10 Under **Web service type,** select any one of the options as shown below:



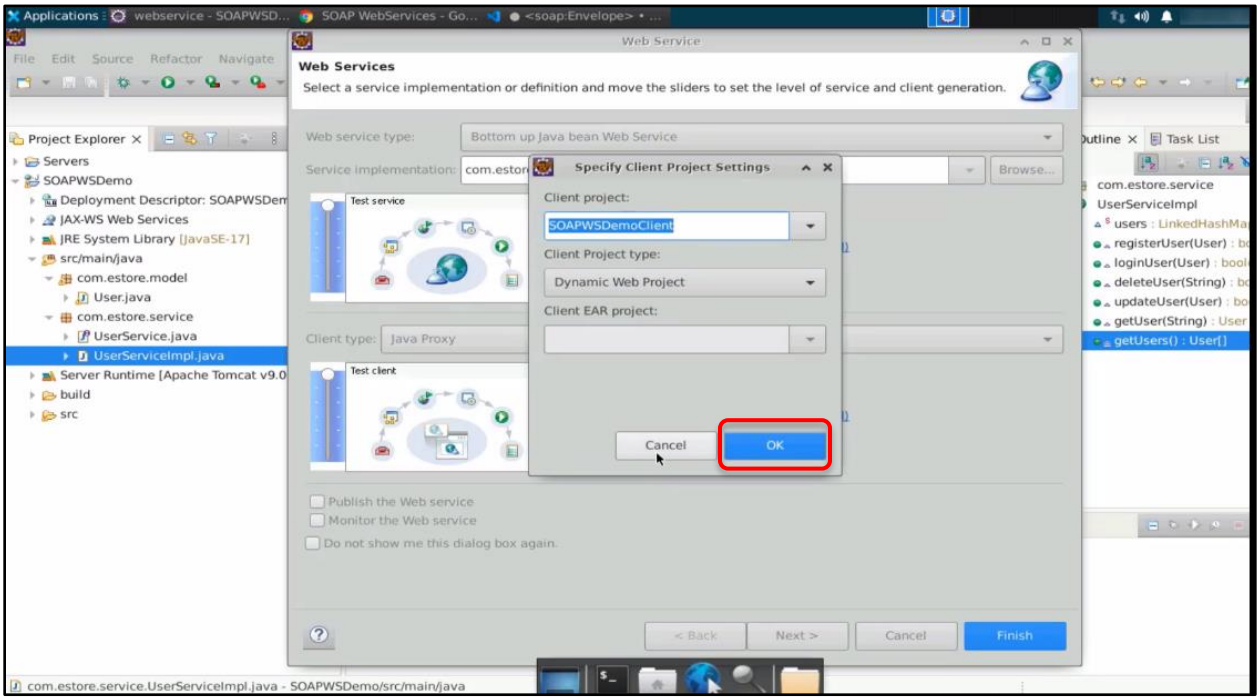
2.11 Select **Client type** as **Java Proxy** and click **Finish**



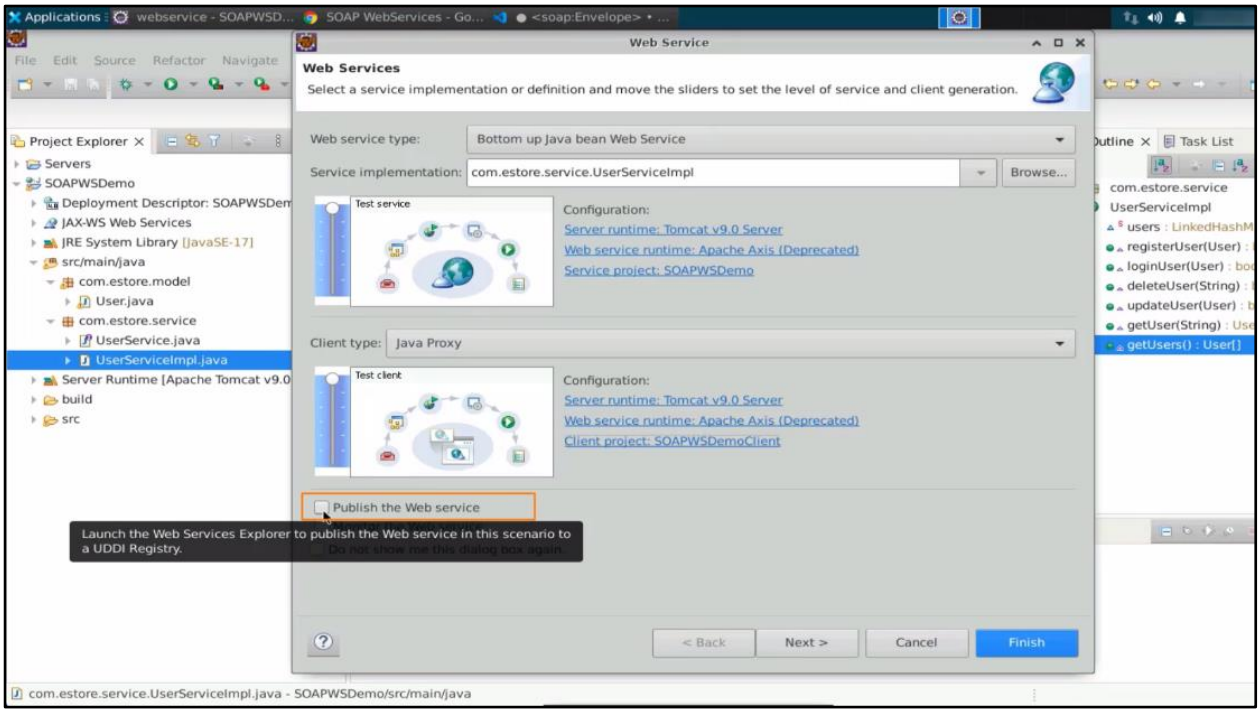
The Service Project is ready, as shown below:



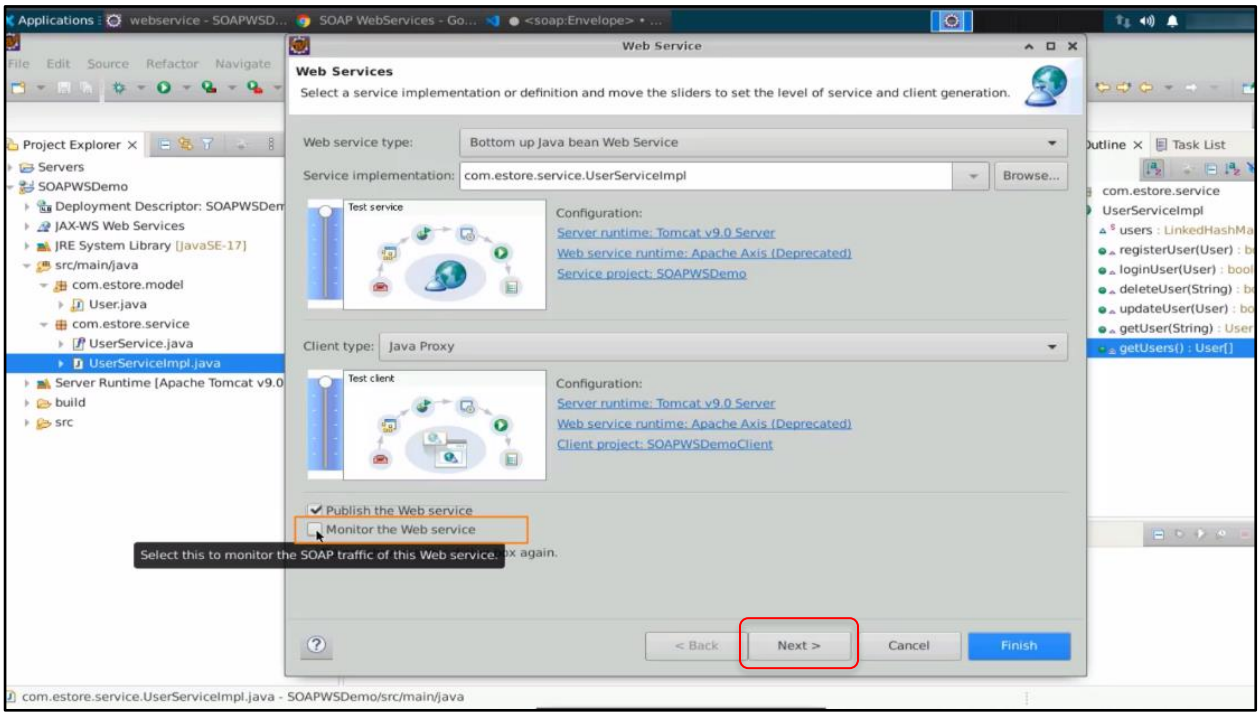
2.12 Click **Client Project** and provide the name of **SOAPWSDemoClient.** Now, click **OK**



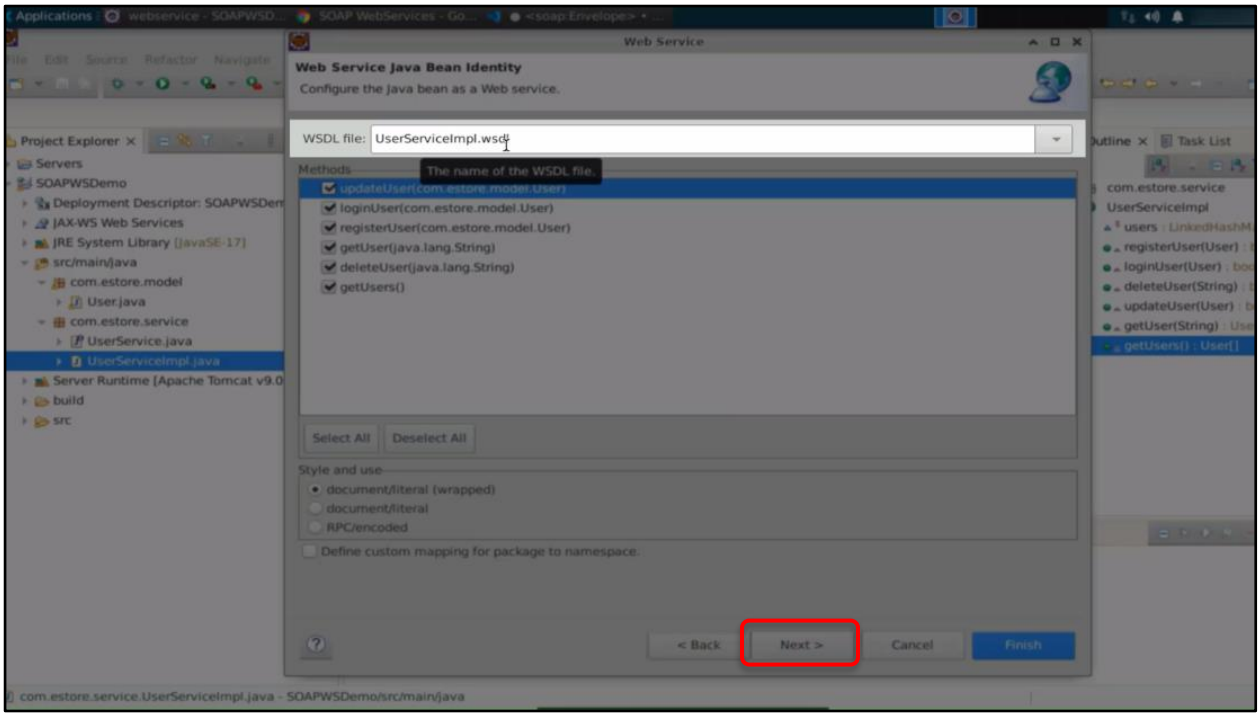
2.13 Select **Publish the Web service** option as shown below:



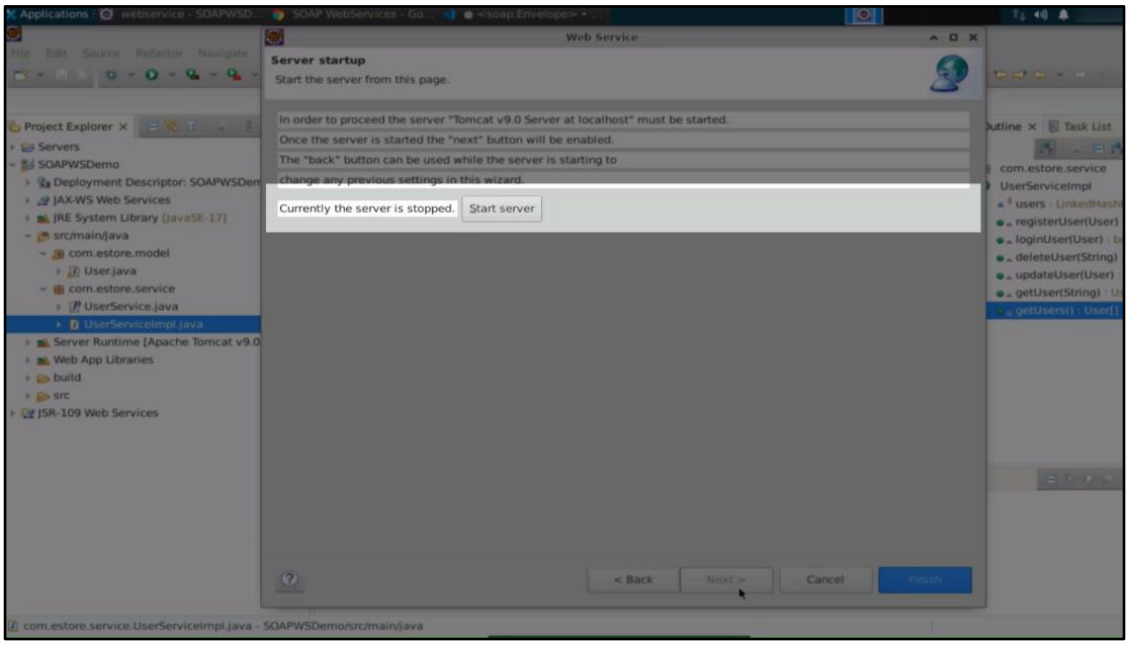
2.14 Select **Monitor the Web service** and click **Next**



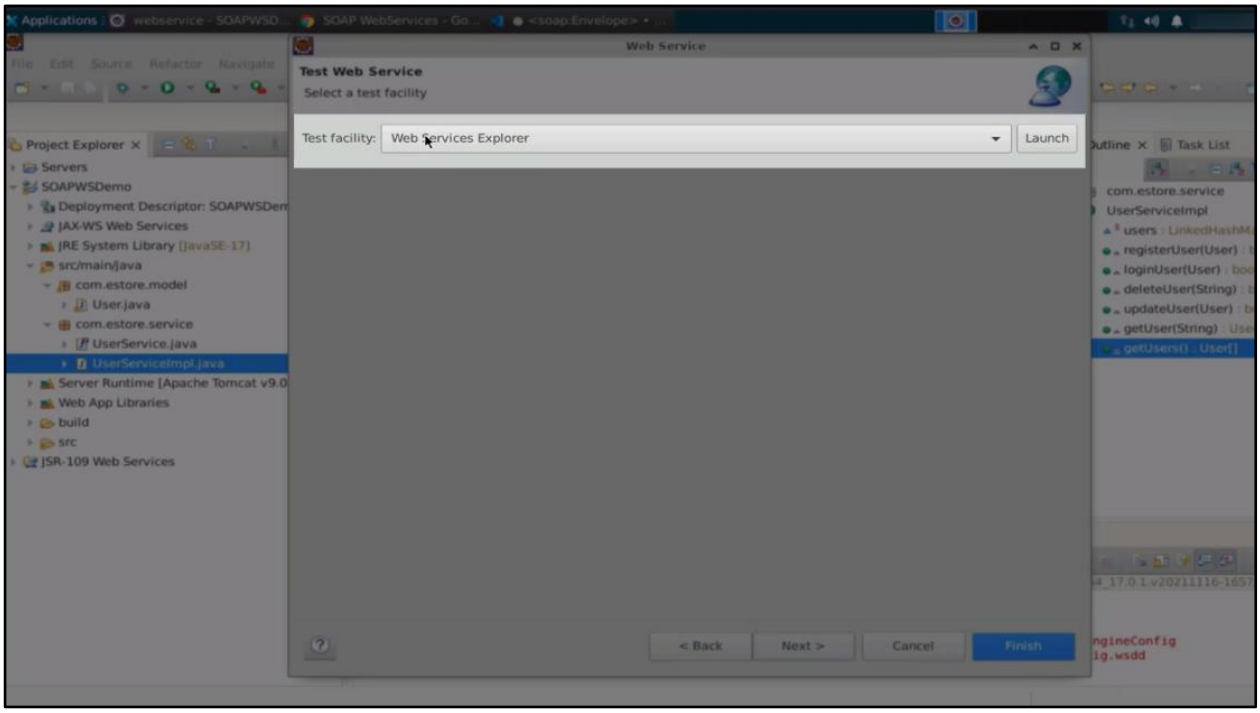
2.15 Write the name for the WSDL file as **UserServiceImpl.wsdl** and click **Next**

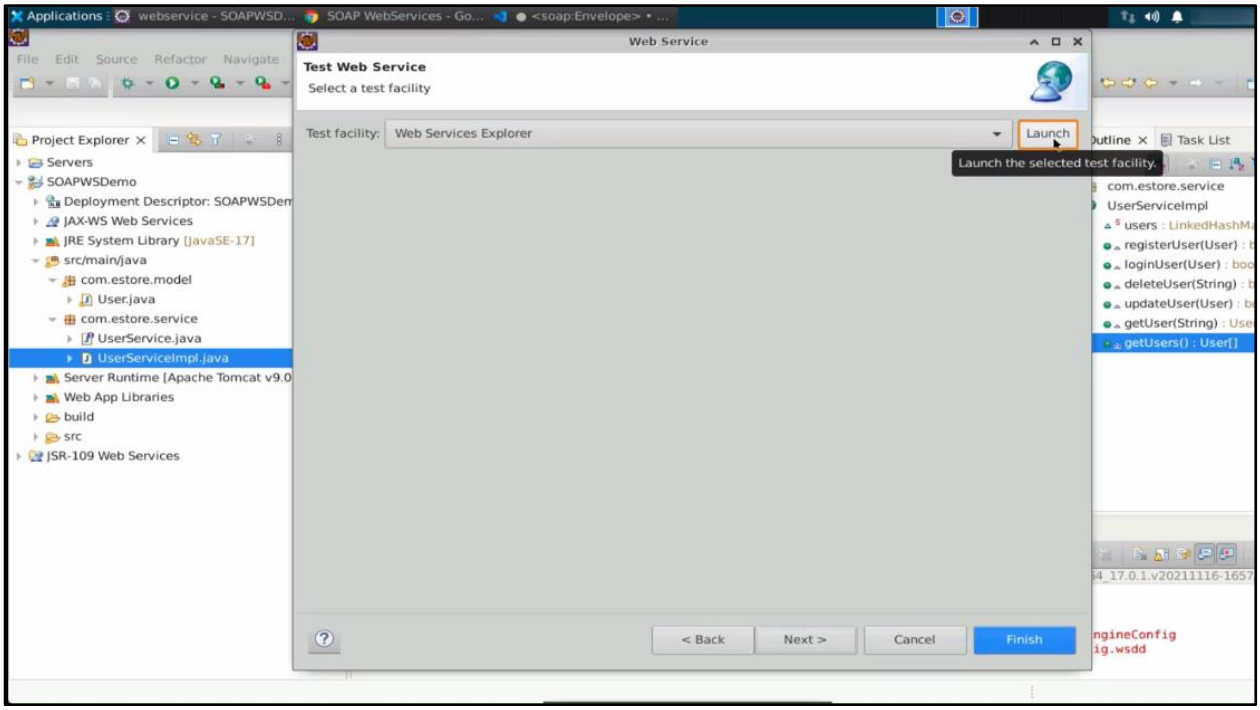


2.16 Click on **Start Server**

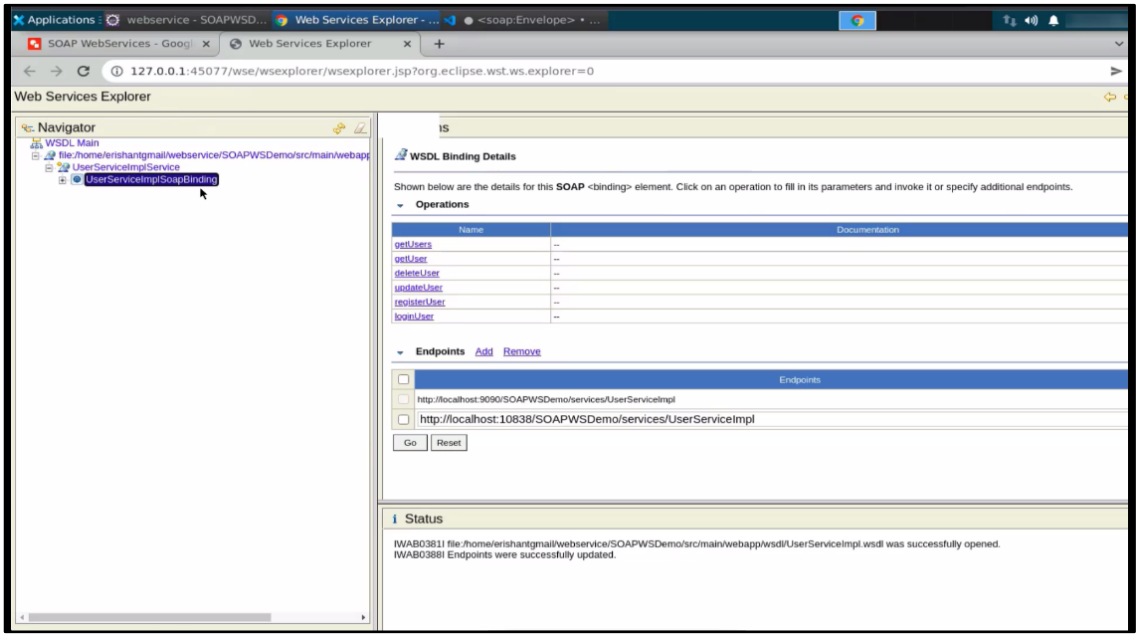


2.17 Select **Web Services Explorer** and click **Launch**





2.18 Open the **Web Service Explorer** to see your service working as shown below:



Hence, we have successfully created and consumed a SOAP web service in the Java code.