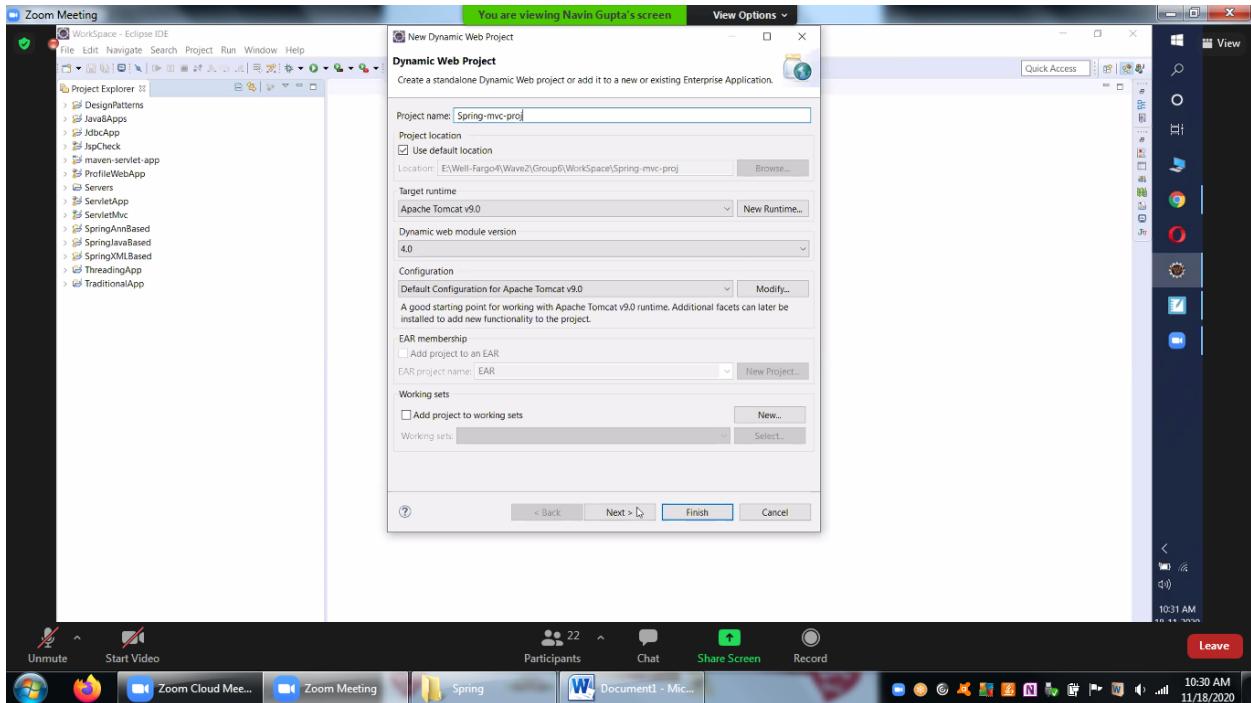
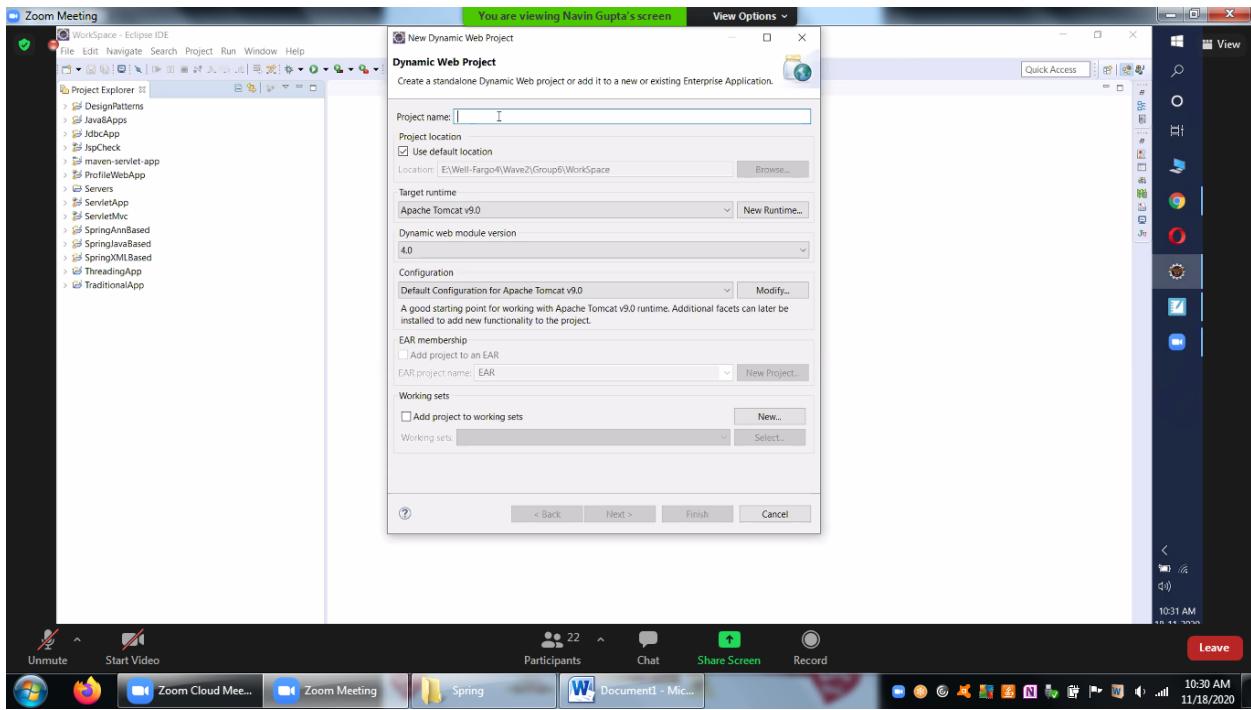
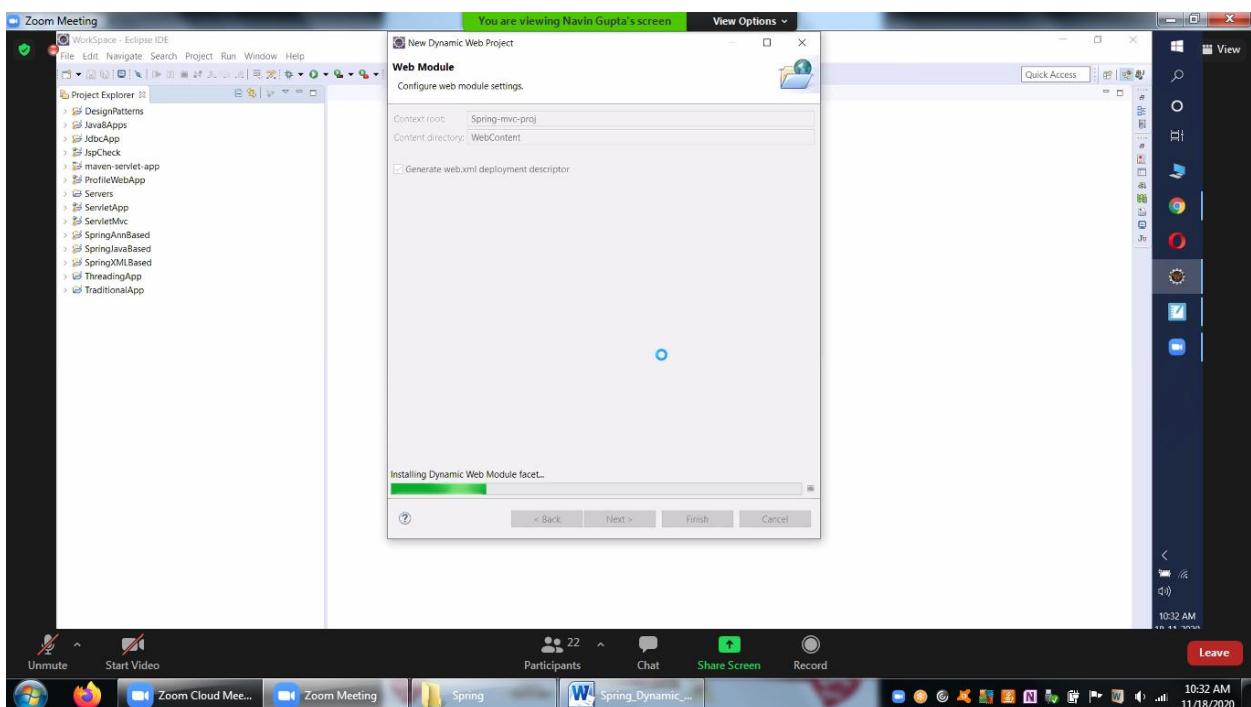
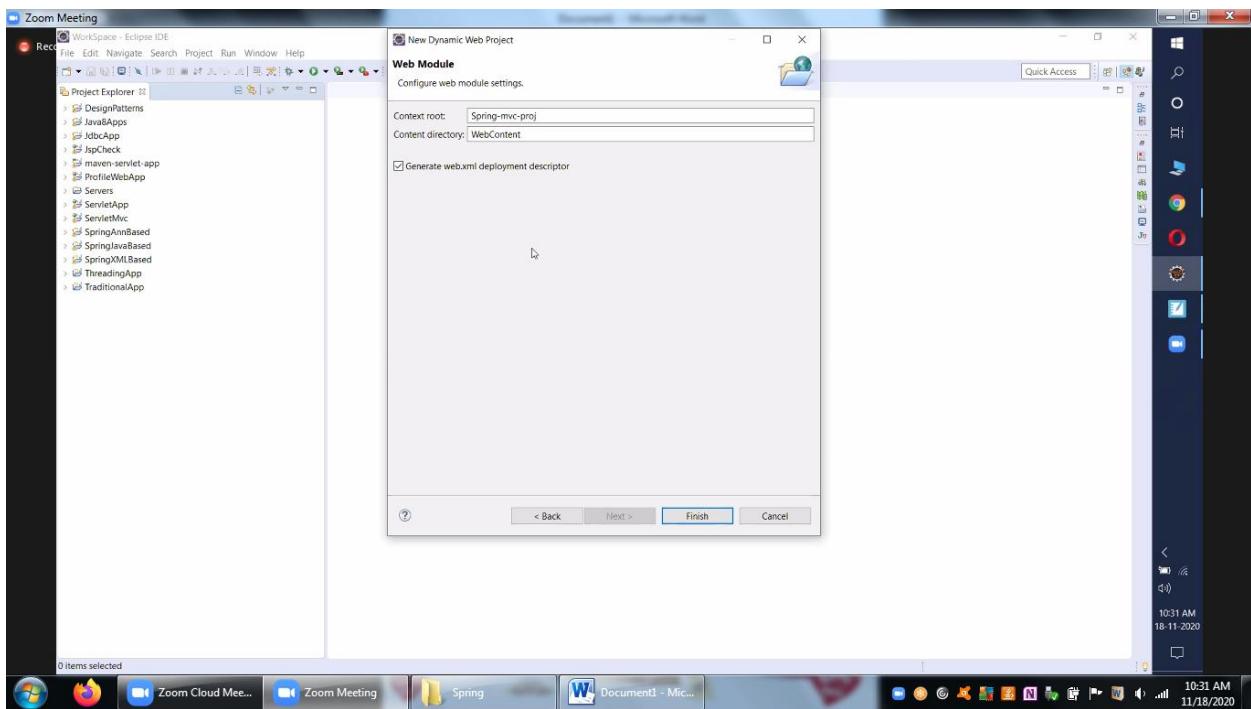
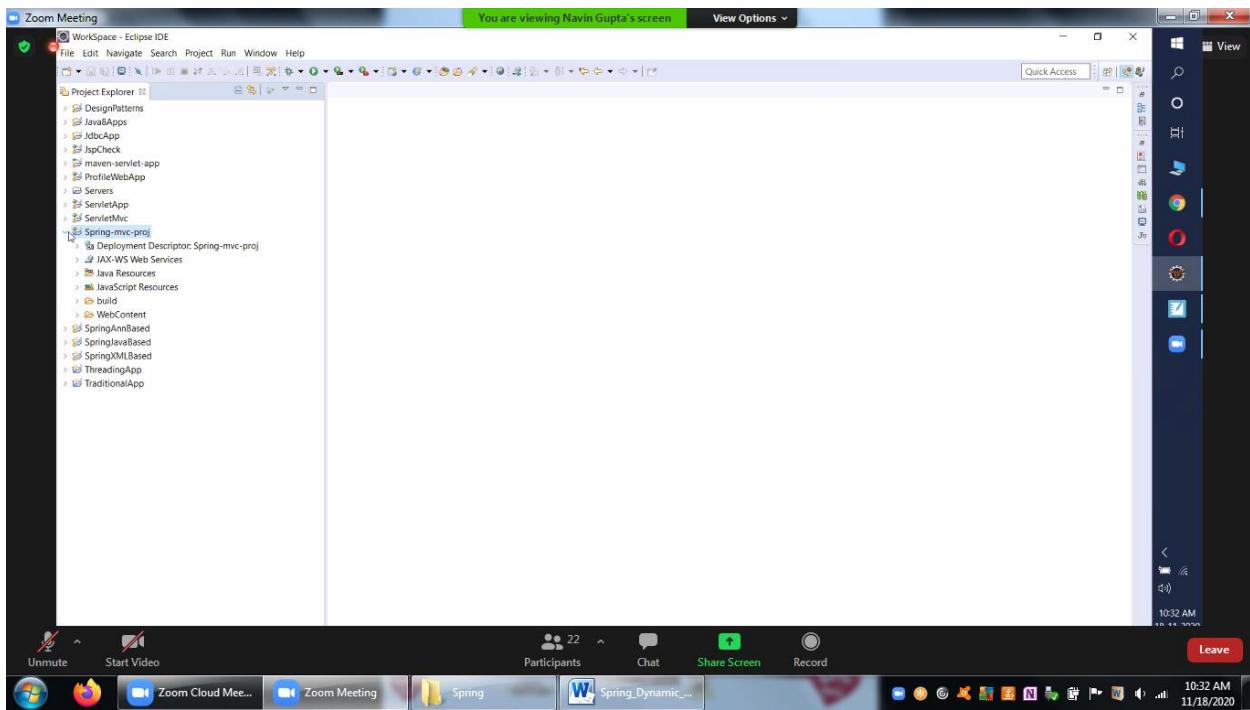


## Creating Dynamic Project for Spring

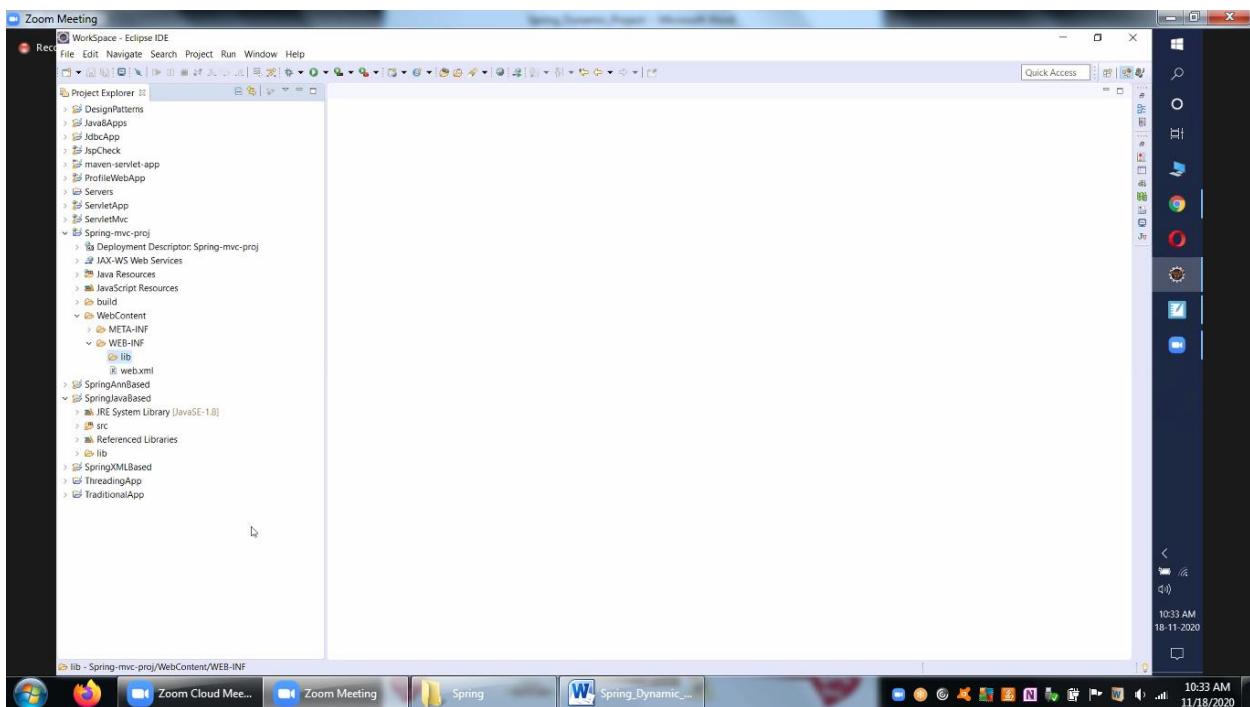


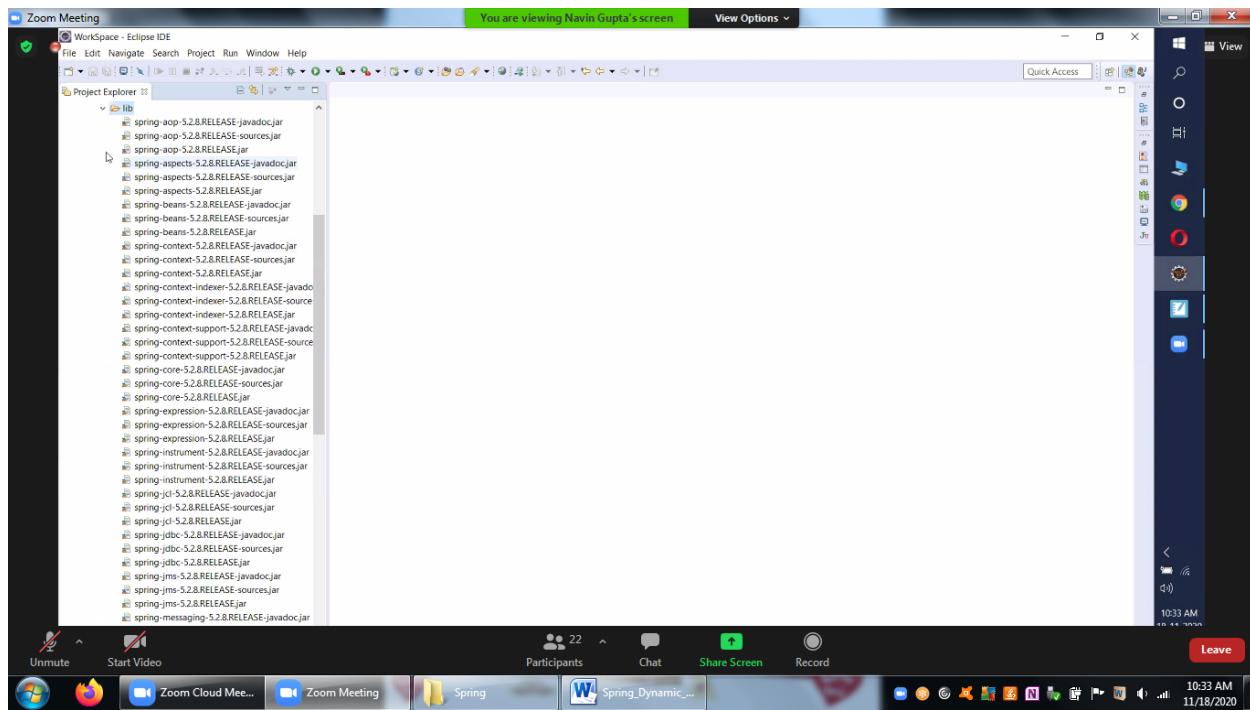
## Selecting the web.xml file mandatory



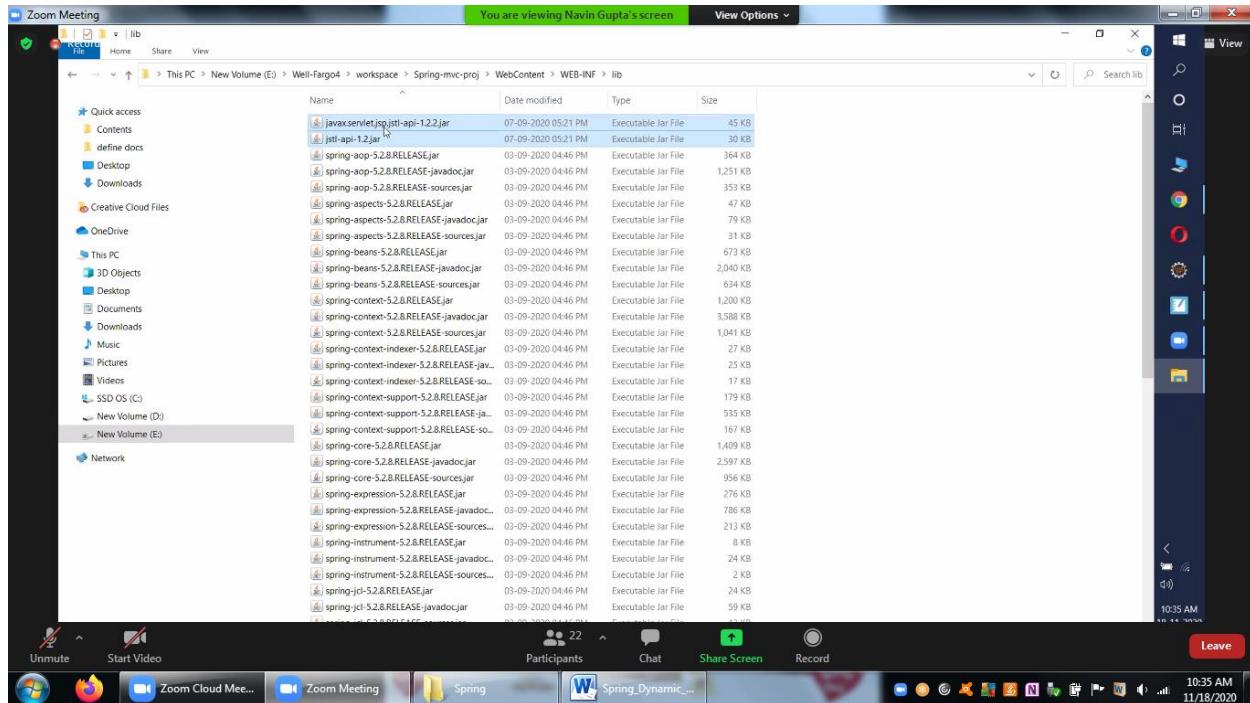


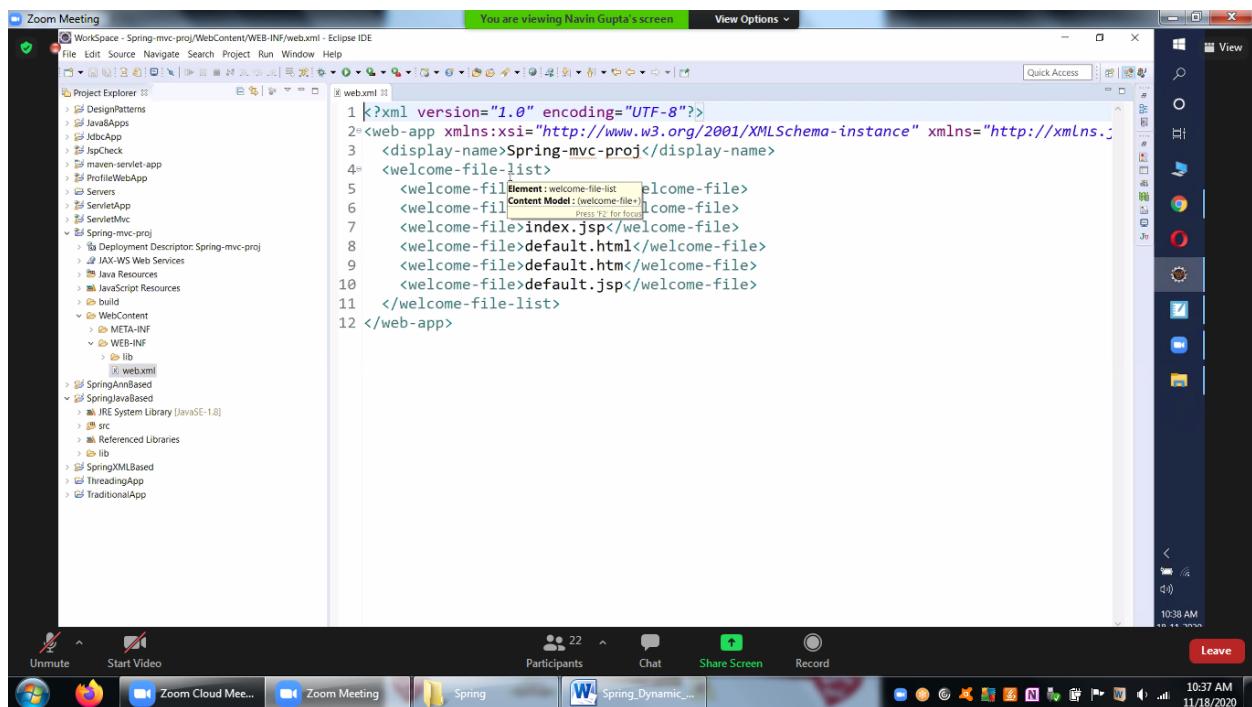
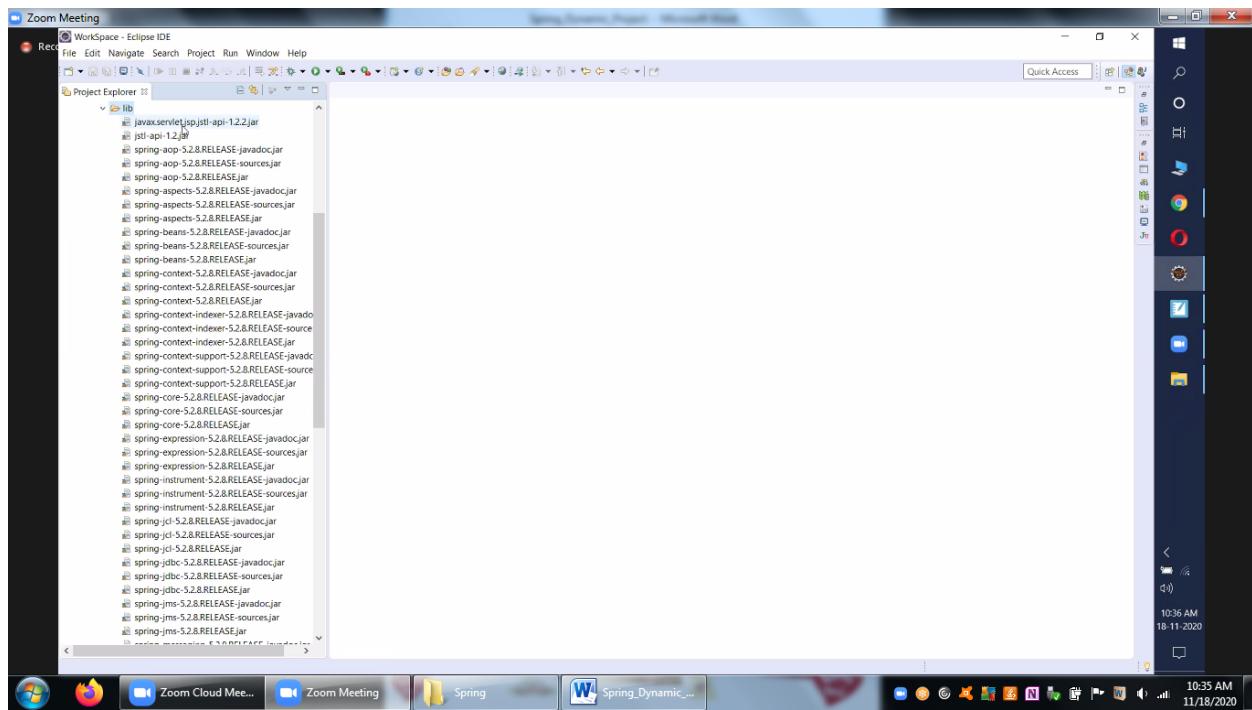
Copy/Add the Spring Jars to lib folder



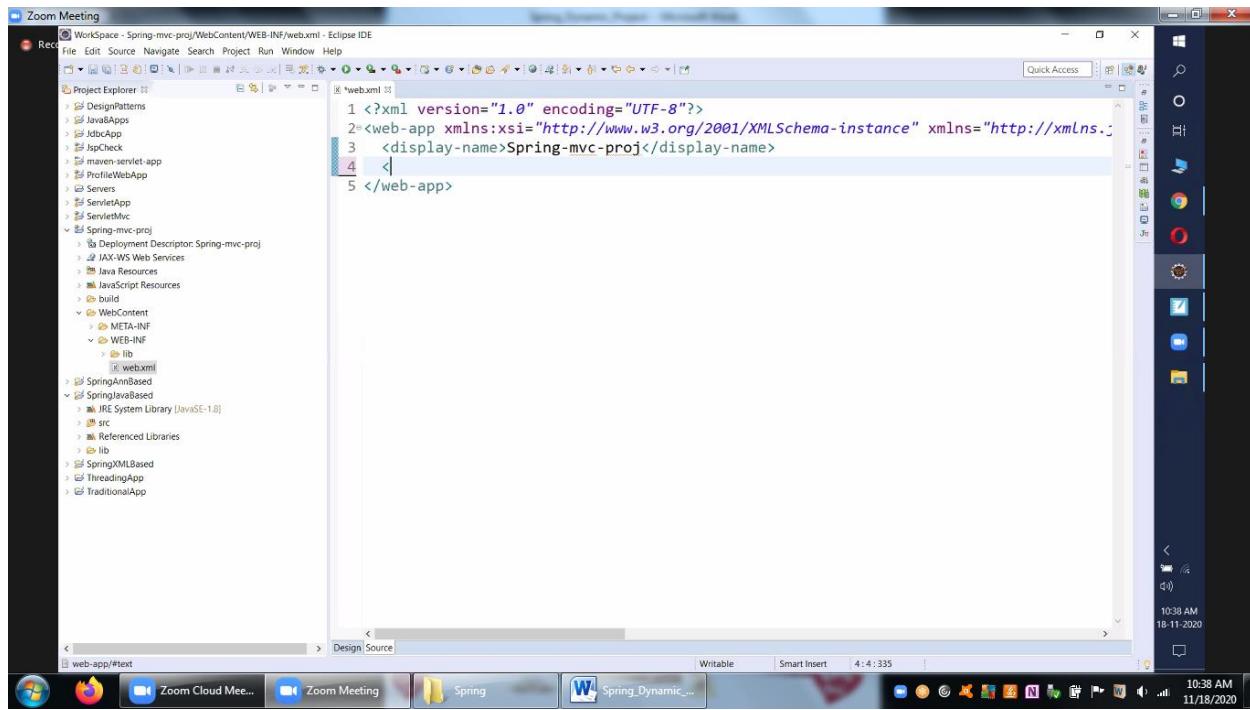


Support of JSP and JSTL also should be added to lib folder



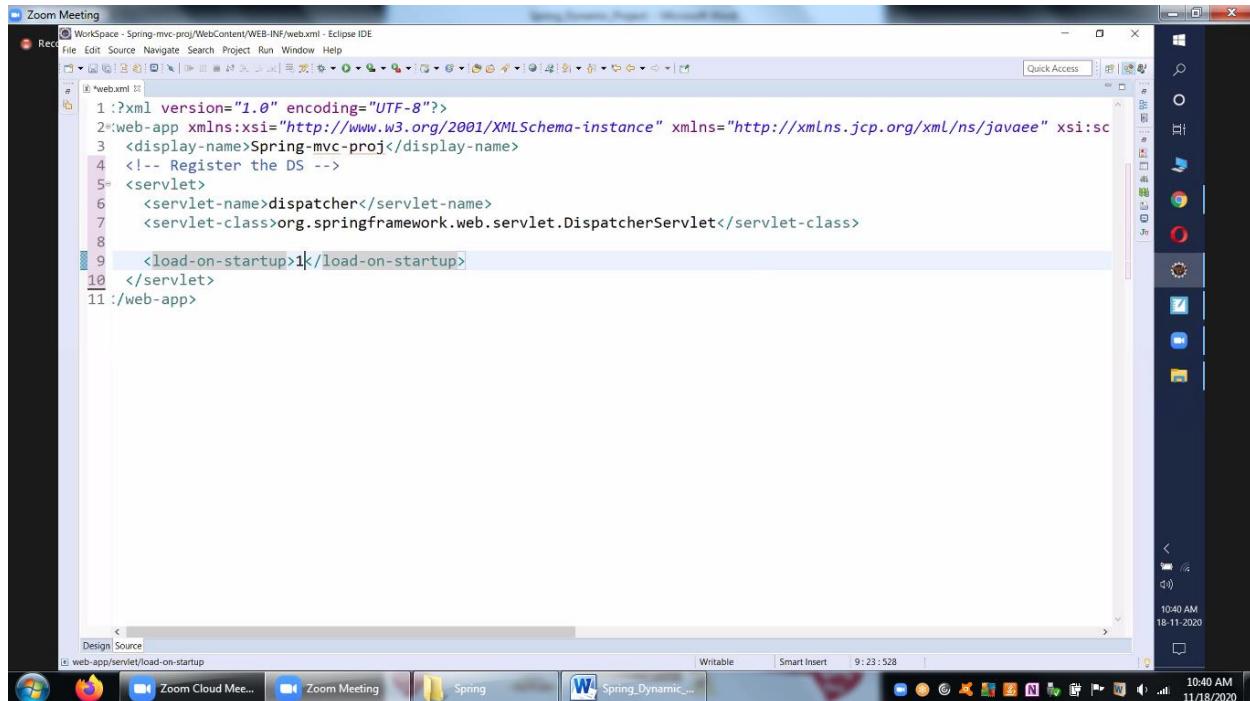


## Delete welcome file list from web.xml file



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://www.w3.org/2001/XMLSchema-instance" >
    <display-name>Spring-mvc-proj</display-name>
</web-app>
```

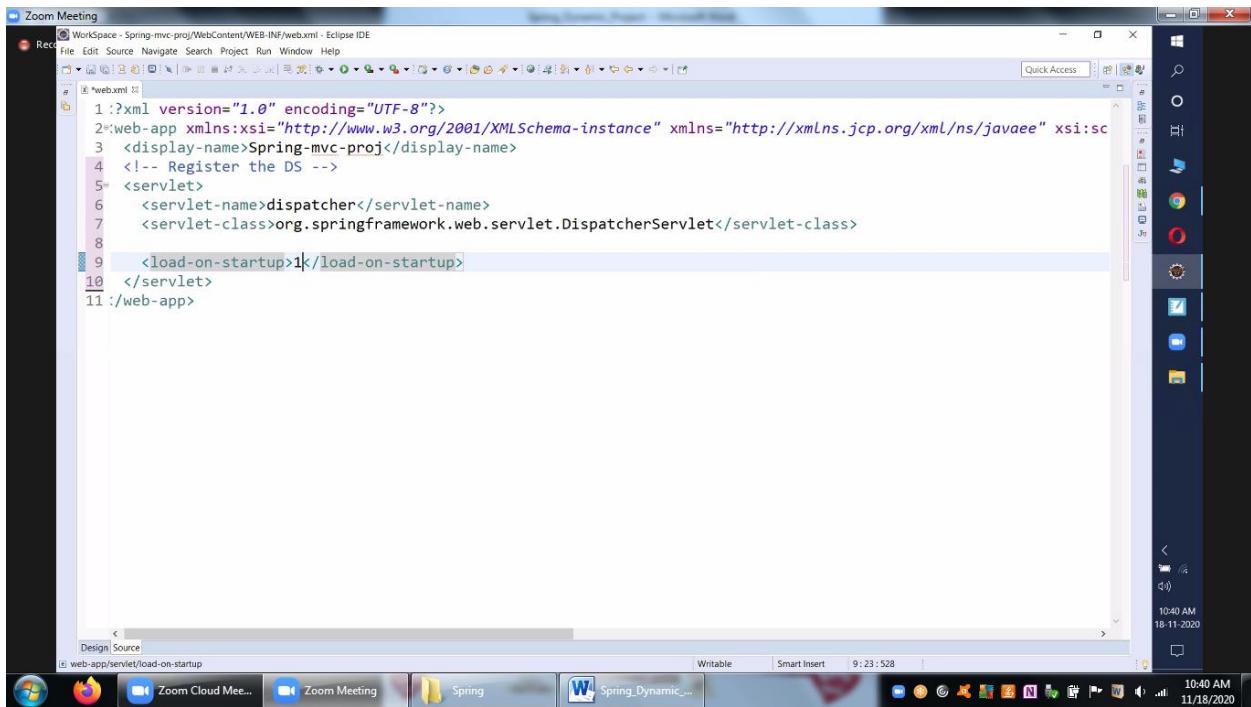
## Register the Dispatcher Servlet



```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://www.w3.org/2001/XMLSchema-instance" >
    <!-- Register the DS -->
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
</web-app>
```

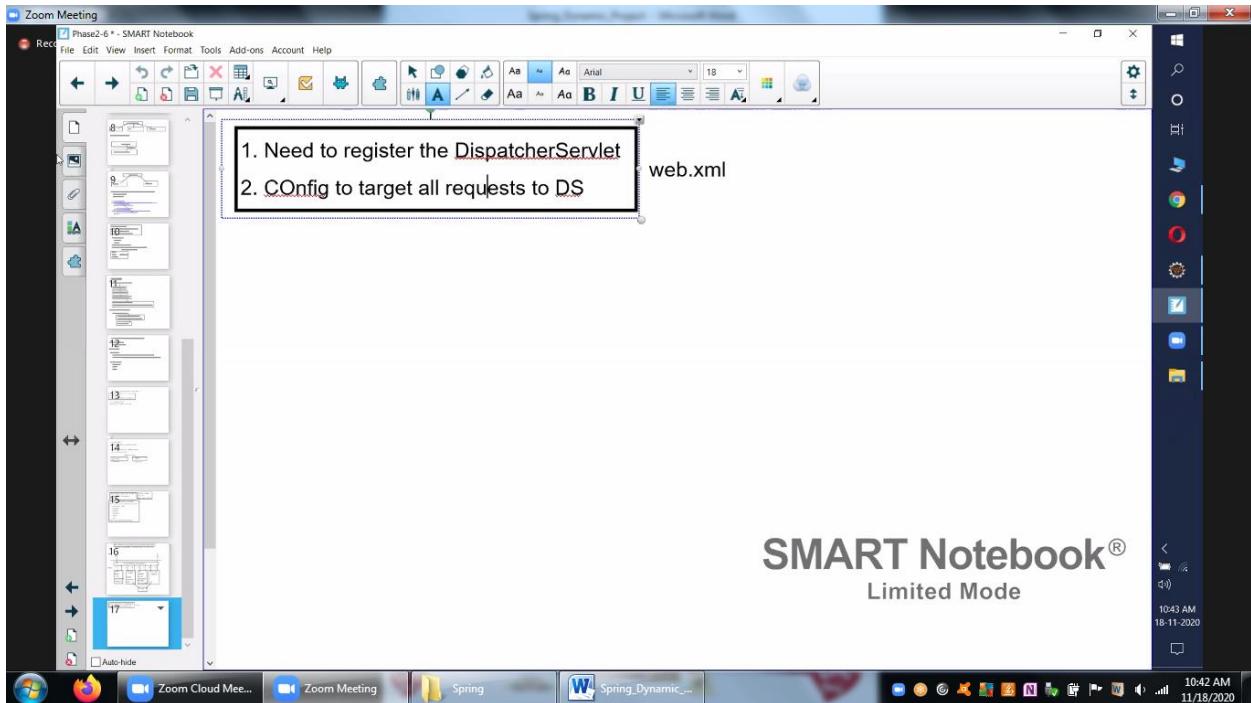
Target all urls to DS

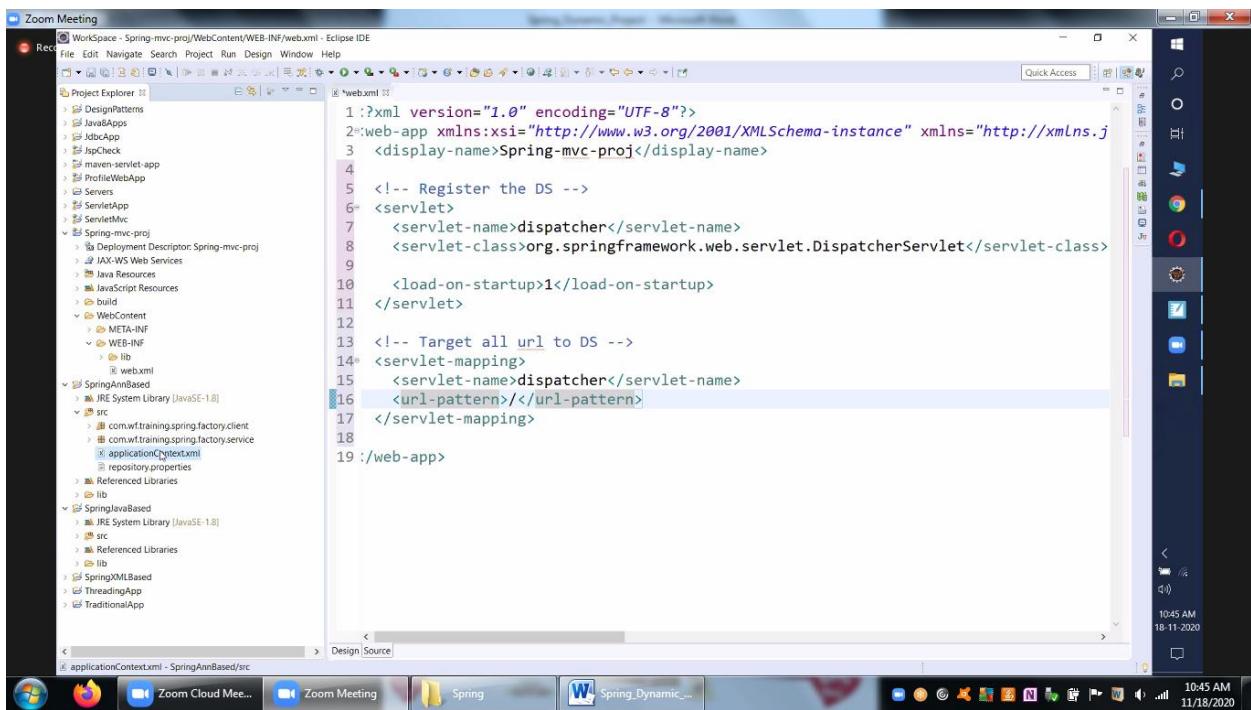
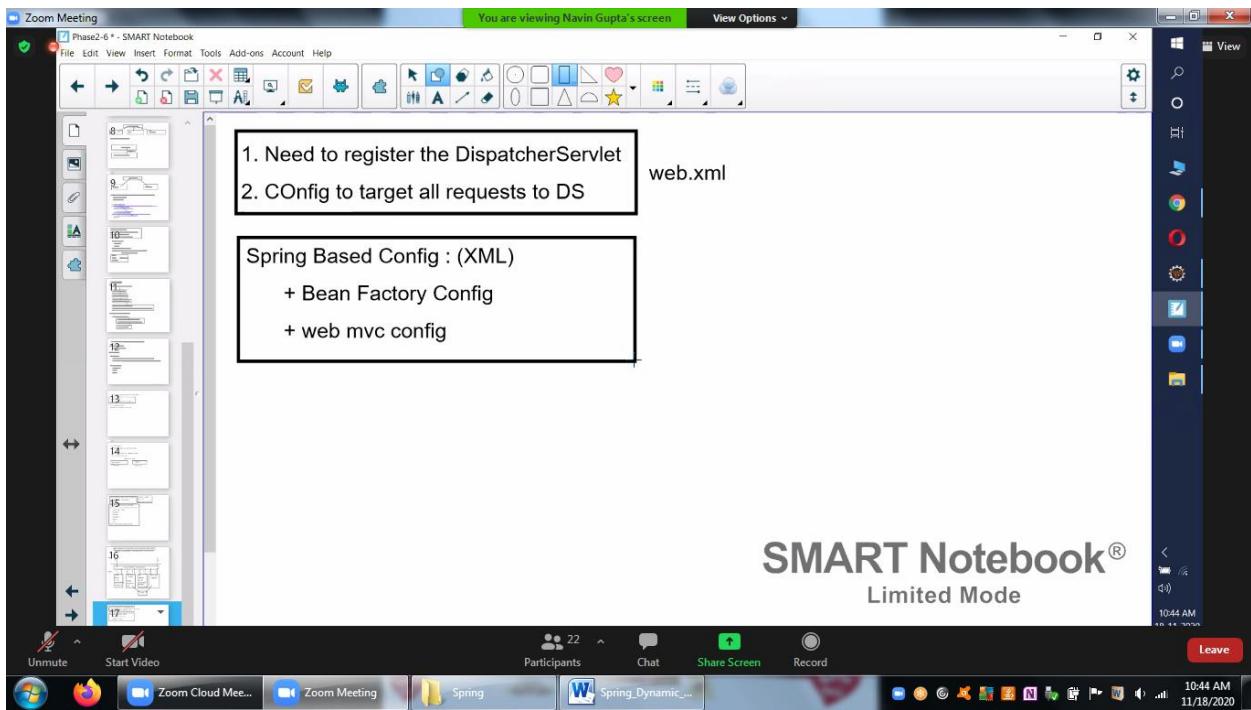
/ stands for every URL (All URL's will be targeted to Spring)



The screenshot shows the Eclipse IDE interface with the 'web.xml' file open in the editor. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://www.w3.org/2001/XMLSchema-instance" version="3.1">
    <!-- Register the DS -->
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
</web-app>
```





Zoom Meeting

You are viewing Navin Gupta's screen

File Edit Navigate Search Project Run Design Window Help

Project Explorer

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.j
3   <display-name>Spring-mvc-proj</display-name>
4
5   <!-- Register the DS -->
6   <servlet>
7     <servlet-name>dispatcher</servlet-name>
8     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
9
10   <load-on-startup>1</load-on-startup>
11 </servlet>
12
13   <!-- Target all url to DS -->
14   <servlet-mapping>
15     <servlet-name>dispatcher</servlet-name>
16     <url-pattern>/</url-pattern>
17   </servlet-mapping>
18
19 </web-app>

```

Participants 22 Chat Share Screen Record

Unmute Start Video Zoom Cloud Meet... Zoom Meeting Spring Spring\_Dynamic\_...

10:45 AM 11/18/2020

Zoom Meeting

You are viewing Navin Gupta's screen

File Edit View Insert Format Tools Add-ons Account Help

2. Config to target all requests to DS

Spring Based Config : (XML)

- + Bean Factory Config
- + web mvc config

View Resolver :

eg : "index" (string) name of view page (controller)

prefix : location

suffix : View Template (extension)

```

<property name="prefix" value="/WEB-INF/views/" />
<property name="suffix" value=".jsp"/>

```

/WEB-INF/views/index.jsp

**SMART Notebook®**  
Limited Mode

Participants 22 Chat Share Screen Record

Unmute Start Video Zoom Cloud Meet... Zoom Meeting Spring Spring\_Dynamic\_...

11:01 AM 11/18/2020

Zoom Meeting

You are viewing Navin Gupta's screen

File Edit Source Navigate Project Run Window Help

Project Explorer

- DesignPatterns
- JavaApps
- JdbcApp
- JspCheck
- maven-servlet-app
- ProfileWebApp
- Servers
- ServiceApp
- ServletMvc
- Spring-mvc-proj
  - Deployment Descriptor: Spring-mvc-proj
  - JAX-WS Web Services
  - Java Resources
  - JavaScript Resources
  - build
  - WebContent
    - META-INF
    - WEB-INF
      - lib
      - views
        - applicationContext.xml
        - web.xml

applicationContext.xml

```
7 http://www.springframework.org/schema/context
8 http://www.springframework.org/schema/context/spring-context.xsd">
9
10 <!-- where to look for components -->
11 <context:component-scan base-package="com.wf.training.spring.factory.service"/>
12
13 <!-- View Resolver Config (web-mvc)-->
14 <!-- Expose the bean of an inbuilt class ViewResolver -->
15 <bean class="org.springframework.web.servlet.view.InternalViewResolver">
16   <!-- inject dependency -->
17   <property name="prefix" value="/WEB-INF/views/"/>
18   <property name="suffix" value=".jsp"/>
19
20 </bean> I
21 </beans>
22
23
24
25
26
27
28
29
30
31
32
33
34
```

Participants Chat Share Screen Record

Leave

Unmute Start Video

Zoom Cloud Meet... Zoom Meeting Spring Spring\_Dynamic\_... Untitled - Notepad

11:03 AM 11/18/2020

Changed the name of the xml file to match the Spring class as web.xml refers to Servlet

Zoom Meeting

You are viewing Navin Gupta's screen

File Edit Source Navigate Project Run Window Help

Project Explorer

- DesignPatterns
- JavaApps
- JdbcApp
- JspCheck
- maven-servlet-app
- ProfileWebApp
- Servers
- ServiceApp
- ServletMvc
- Spring-mvc-proj
  - Deployment Descriptor: Spring-mvc-proj
  - JAX-WS Web Services
  - Java Resources
  - JavaScript Resources
  - build
  - WebContent
    - META-INF
    - WEB-INF
      - lib
      - views
        - applicationContext.xml
        - web.xml

applicationContext.xml

```
7 http://www.springframework.org/schema/context
8 http://www.springframework.org/schema/context/spring-context.xsd">
9
10 <!-- where to look for components -->
11 <context:component-scan base-package="com.wf.training.spring.factory.service"/>
12
13 <!-- View Resolver Config (web-mvc)-->
14 <!-- Expose the bean of an inbuilt class ViewResolver -->
15 <bean class="org.springframework.web.servlet.view.InternalViewResolver">
16   <!-- inject dependency -->
17   <property name="prefix" value="/WEB-INF/views/"/>
18   <property name="suffix" value=".jsp"/>
19
20 </bean> I
21 </beans>
22
23
24
25
26
27
28
29
30
31
32
33
34
```

Participants Chat Share Screen Record

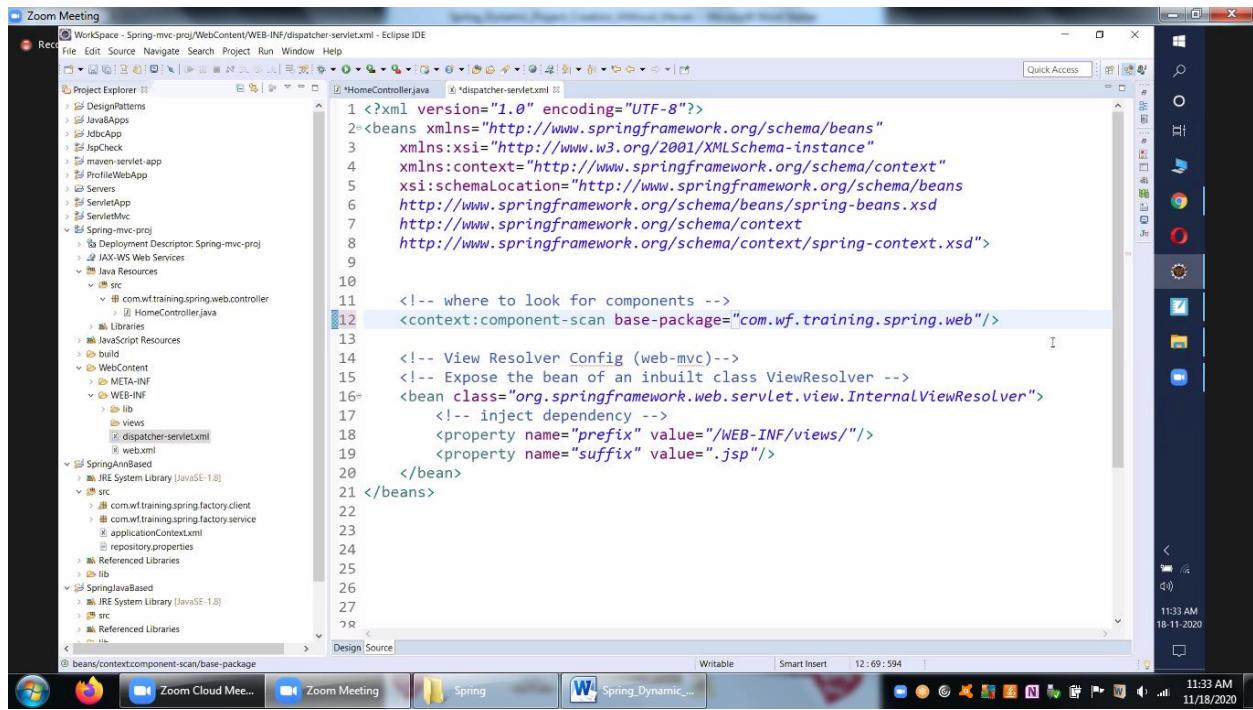
Leave

Unmute Start Video

Zoom Cloud Meet... Zoom Meeting Spring Spring\_Dynamic\_... Untitled - Notepad

11:03 AM 11/18/2020

## Change the package name in the xml file



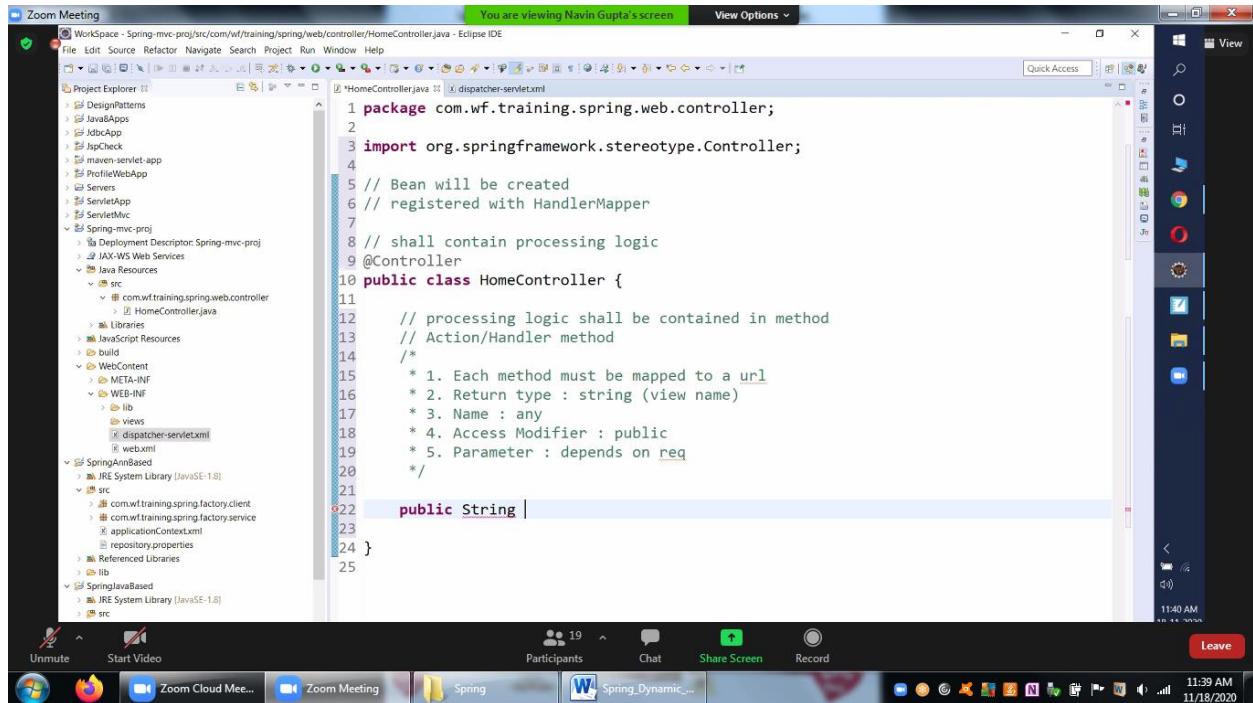
The screenshot shows the Eclipse IDE interface with the 'dispatcher-servlet.xml' file open in the editor. The XML code defines a Spring configuration for a web application. The 'base-package' attribute in the 'context:component-scan' tag is currently set to 'com.wf.training.spring.web'. This needs to be changed to match the package name of the POJO classes.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- where to look for components -->
    <context:component-scan base-package="com.wf.training.spring.web"/>

    <!-- View Resolver Config (web-mvc)-->
    <!-- Expose the bean of an inbuilt class ViewResolver -->
    <bean class="org.springframework.web.servlet.view.InternalViewResolver">
        <!-- inject dependency -->
        <property name="prefix" value="/WEB-INF/views/"/>
        <property name="suffix" value=".jsp"/>
    </bean>
</beans>
```

## Create POJO classes – HomeController.java



The screenshot shows the Eclipse IDE interface with the 'HomeController.java' file open in the editor. The code defines a Spring MVC controller named 'HomeController' that implements the 'Controller' stereotype. It contains a single method 'public String index()' which is annotated with '@RequestMapping' to map to the root URL ('/').

```
package com.wf.training.spring.web.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
@RequestMapping("/")
public class HomeController {

    @RequestMapping("/")
    public String index() {
        return "index";
    }
}
```

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

- Title Bar:** Zoom Meeting, WorkSpace - Spring-mvc-proj/src/com/wf/training/spring/web/controller/HomeController.java - Eclipse IDE
- Project Explorer:** Shows the project structure including DesignPatterns, JavaApps, JdbcApp, JspCheck, maven-servlet-app, ProfileWebApp, Servers, ServletApp, ServletMvc, and Spring-mvc-proj.
- Code Editor:** Displays the HomeController.java code. The code defines a public class HomeController with a single method home() that returns "index".
- Bottom Status Bar:** Shows the date and time as 11:43 AM 11/18/2020.

```
6 // Bean will be created
7 // registered with HandlerMapper
8
9 // shall contain processing logic
10 @Controller
11 public class HomeController {
12
13     // processing logic shall be contained in method
14     // Action/Handler method
15     /*
16      * 1. Each method must be mapped to a url
17      * 2. Return type : string (view name)
18      * 3. Name : any
19      * 4. Access Modifier : public
20      * 5. Parameter : depends on req
21     */
22
23     // to respond to root URL (home/welcome file)
24     @RequestMapping("/")
25     public String home() {
26         // add business logic
27         |
28         // respond back with a view page name
29         return "index";
30     }
31
32 }
```

Create JSP page(s) in views folder

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

- Title Bar:** Zoom Meeting, WorkSpace - Spring-mvc-proj/WebContent/WEB-INF/views/index.jsp - Eclipse IDE
- Project Explorer:** Shows the project structure including DesignPatterns, JavaApps, JdbcApp, JspCheck, maven-servlet-app, ProfileWebApp, Servers, ServletApp, ServletMvc, and Spring-mvc-proj.
- Code Editor:** Displays the index.jsp code, which contains a simple JSP page with an H1 tag.
- Bottom Status Bar:** Shows the date and time as 11:45 AM 11/18/2020.

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>Hello All, Spring Web is responding...</h1>
11 </body>
12 </html>
```

## Run the code

WorkSpace - Spring-mvc-proj/src/com/wf/training/spring/web/controller/HomeController.java - Eclipse IDE

```
6 // Bean will be created
7 // registered with HandlerMapper
8
9 // shall contain processing logic
10 @Controller
11 public class HomeController {
```

// processing logic shall be contained in method  
// Action/Handler method

/\*  
 \* 1. Each method must be mapped to a url  
 \* 2. Return type : string (view name)  
 \* 3. Name : any  
 \* 4. Access Modifier : public  
 \* 5. Parameter : depends on req  
 \*/

// to respond to root URL (home/welcome file)

```
@RequestMapping("/")
public String home() {
    // add business logic
    return "index";
}
```

Right-click context menu options for the selected 'Run As' item:

- 1 Run on Server
- 2 Java Application

Java EE Tools

Properties

WorkSpace - Spring-mvc-proj/WebContent/WEB-INF/dispatcher-servlet.xml - Eclipse IDE

```
7 http://www.springframework.org/schema/context
8 http://www.springframework.org/schema/context/spring-context.xsd"
9
10
11 <!-- where to look for components -->
12 <context:component-scan base-package="com.wf.training.spring.web"/>
13
14 <!-- View Resolver Config (web-mvc)-->
15 <!-- Expose the bean of an inbuilt class ViewResolver -->
16 <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
17     <!-- inject dependency -->
18     <property name="prefix" value="/WEB-INF/views/"/>
19     <property name="suffix" value=".jsp"/>
20 </bean>
21 </beans>
22
23
24
25
26
27
28
29
30
31
32
33
34
```

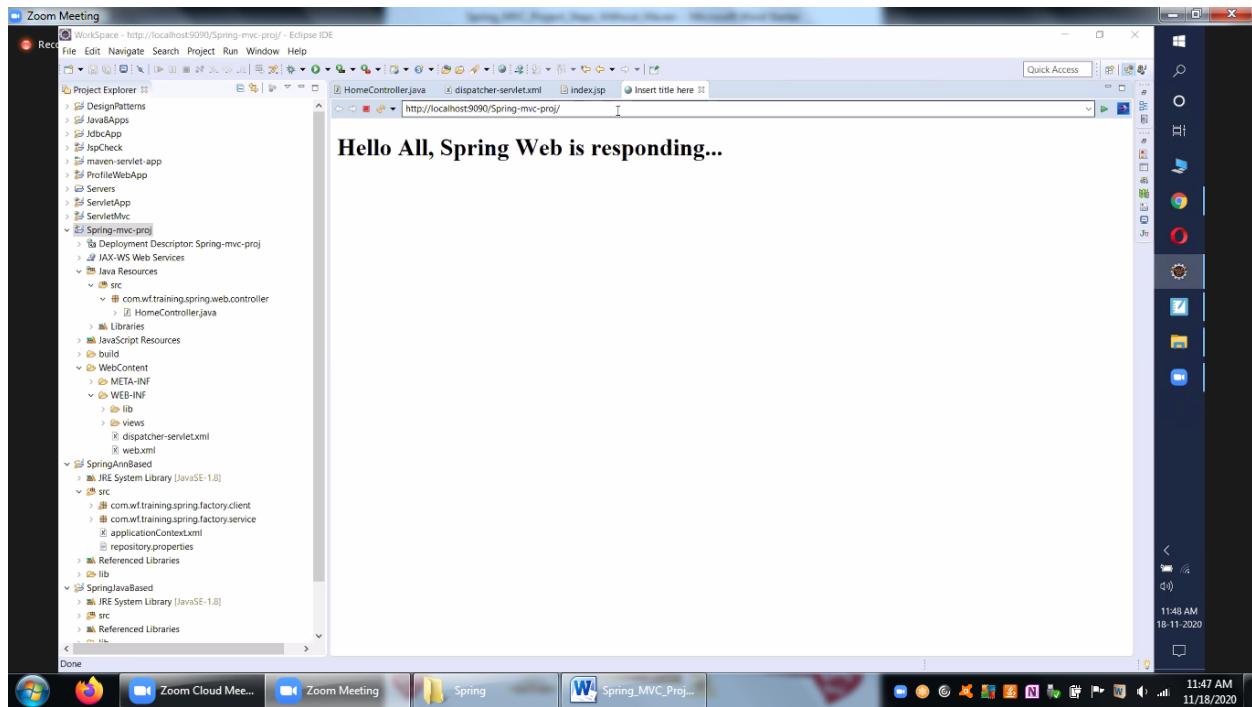
Right-click context menu options for the selected 'InternalResourceViewResolver' bean definition:

- 1 Run on Server
- 2 Java Application

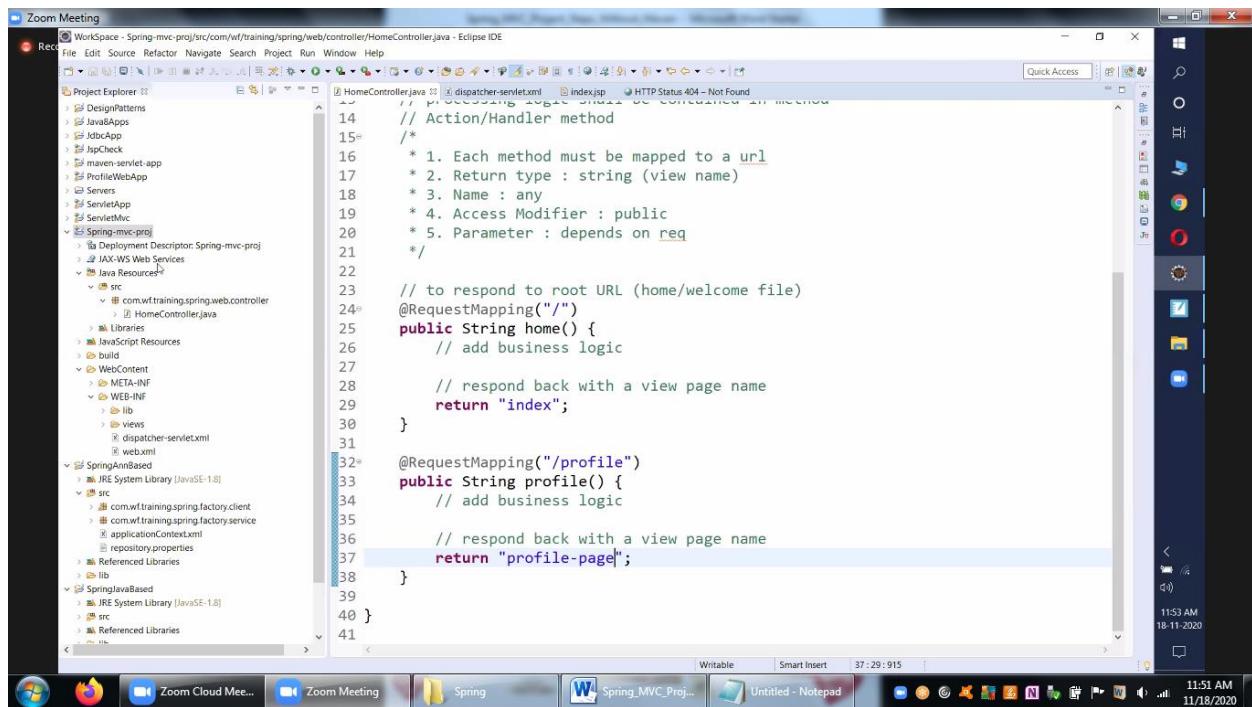
Java EE Tools

Properties

The first page of JSP execution is successful (Single URL Mapping)



Adding one more handler method

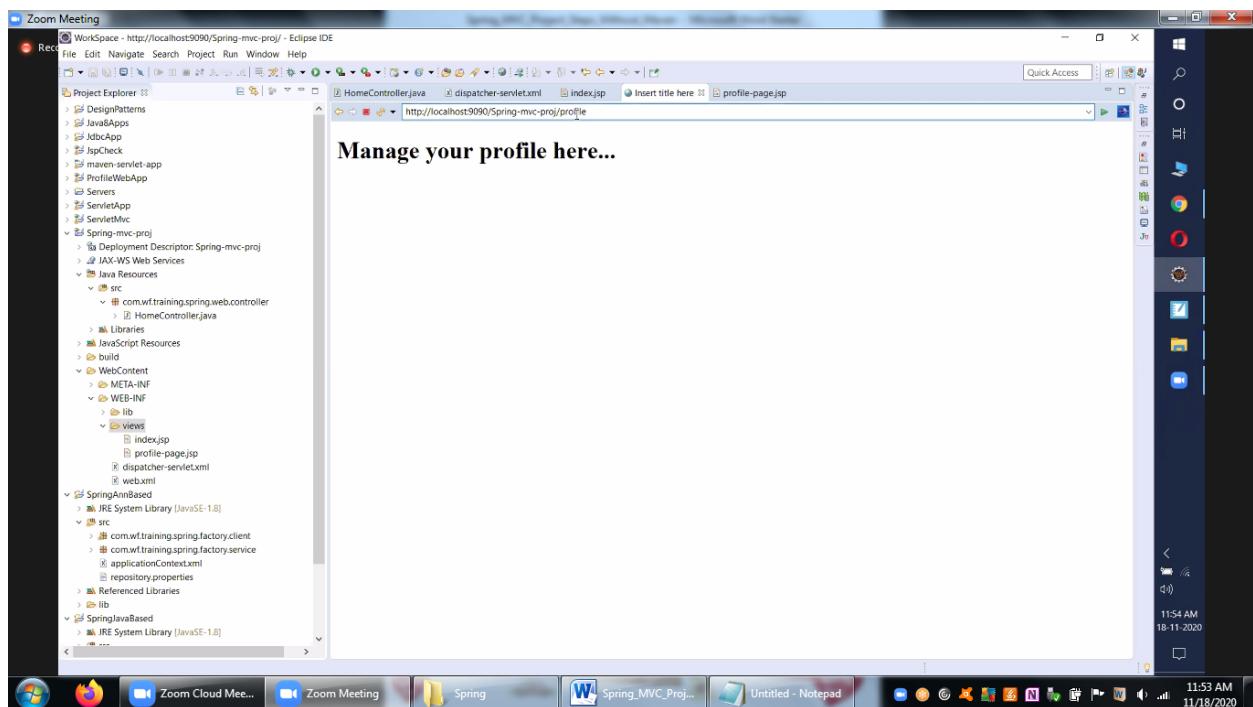


The screenshot shows the Eclipse IDE interface. The code editor displays the following JSP code:

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2     pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>Manage your profile here...</h1>
11 </body>
12 </html>
```

The Project Explorer view on the left shows the project structure, including the Spring-mvc-proj folder, its subfolders (src, WebContent, META-INF), and files (HomeController.java, dispatcher-servlet.xml, index.jsp). The status bar at the bottom indicates the date and time as 11:52 AM 11/18/2020.

We can use as link from index page as well



Instead of creating the servlet class, we are creating a simple POJO classes for request and response.

If we face any issues verify the below settings, we should the below folders and jars in the java build path

