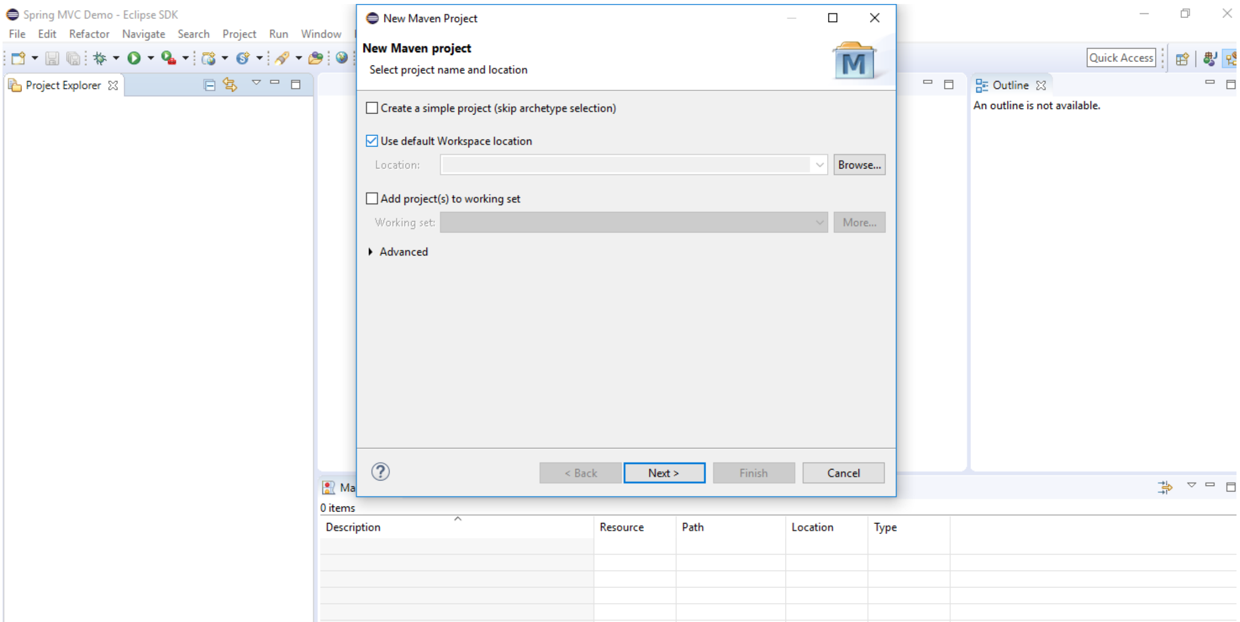
Spring MVC with Hibernate

1. Create **Maven Project** and choose use default workspace location
    a. Click Next



New Maven project

Select project name and location

☐ Create a simple project (skip archetype selection)

☑ Use default Workspace location

Location:

☐ Add project(s) to working set

Working set:

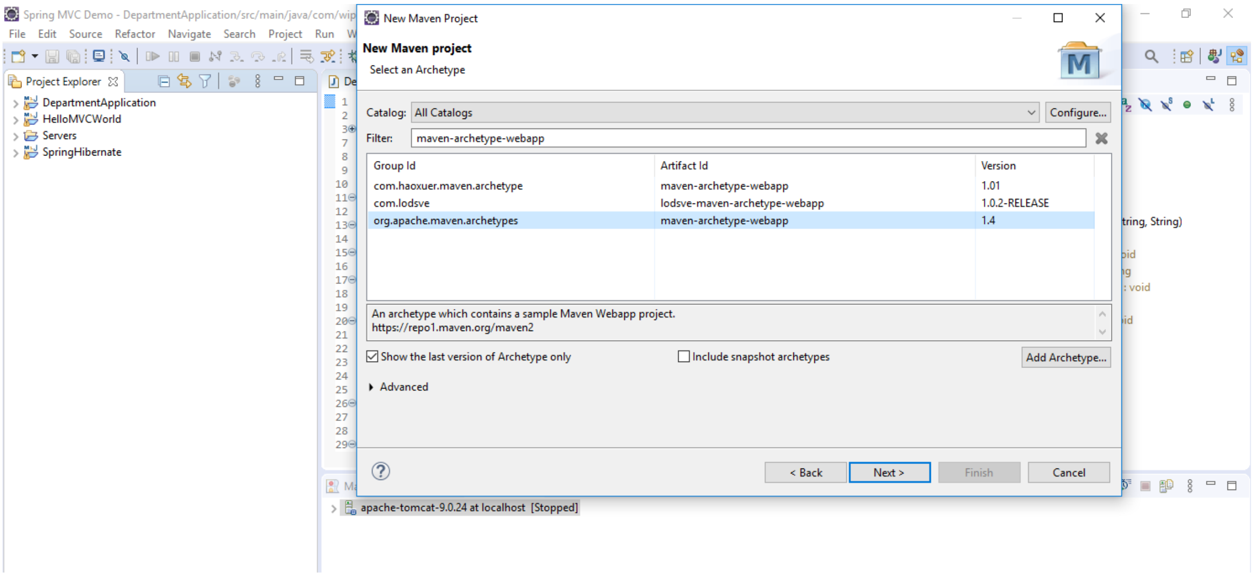
▸ Advanced

An outline is not available.

0 items

Description	Resource	Path	Location	Type

2. Under Archetype choose **maven-archetype-webapp** from below given group Id and version
    a. Group Id : **org.apache.maven.archetypes**
    b. Version can be any latest version. Here It is taken 1.4
        i. You can also pick a specific version by checking the ***Include snapshot archetypes***
    c. Click Next



New Maven project

Select an Archetype

Catalog: All Catalogs

Filter: maven-archetype-webapp

Group Id	Artifact Id	Version
com.haoxuer.maven.archetype	maven-archetype-webapp	1.01
com.lodsve	lodsve-maven-archetype-webapp	1.0.2-RELEASE
org.apache.maven.archetypes	maven-archetype-webapp	1.4

An archetype which contains a sample Maven Webapp project.
https://repo1.maven.org/maven2

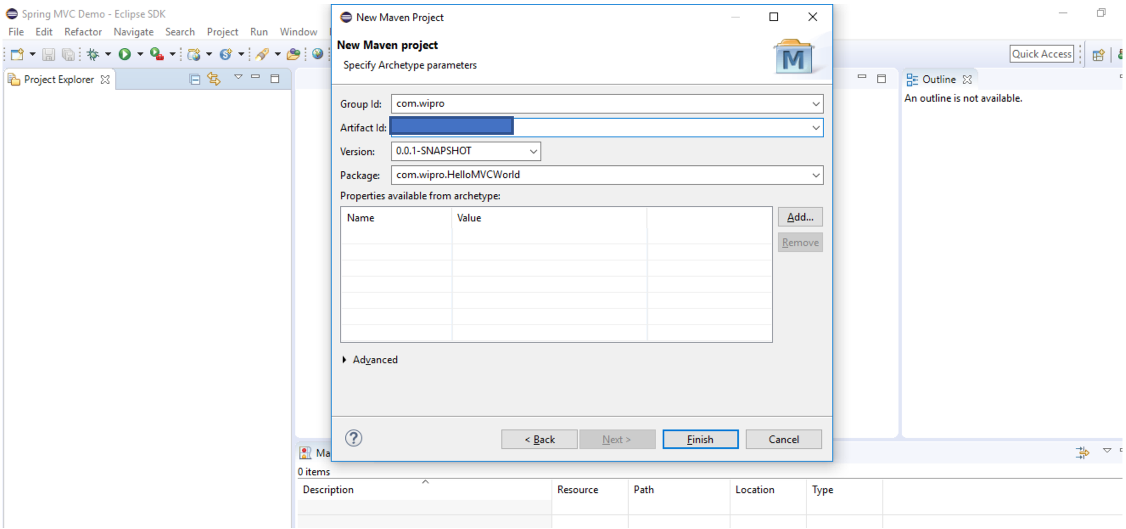
☑ Show the last version of Archetype only

☐ Include snapshot archetypes

▸ Advanced

apache-tomcat-9.0.24 at localhost [Stopped]

3. Fill the Maven Project details
    a. Group ID :: com.wipro
    b. Artifact ID :: DepartmentApplication
    c. Click finish



New Maven project

Specify Archetype parameters

Group Id: com.wipro

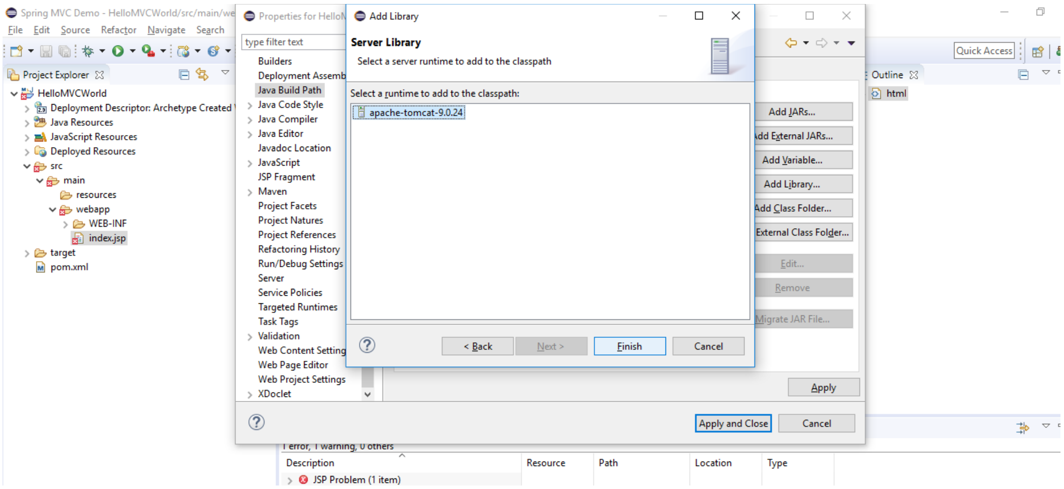
Artifact Id: [redacted]

Version: 0.0.1-SNAPSHOT

Package: com.wipro.HelloMVCWorld

Properties available from archetype:

4. This generates Spring Web Project. With the initial Error
    a. This is because the build path is not aware of servlets and jsp APIs.
    b. Add Tomcat server
        i. Click on window menu -> preferences -> server -> Runtime Environment
            1. Search the unzipped folder of your version of Appache Tomcat
    c. right click on project -> Properties -> Java Build Path -> Add Library
    d. Server Runtime -> Apache Tomcat -> Finish



Server Library

Select a server runtime to add to the classpath

Select a runtime to add to the classpath:

apache-tomcat-9.0.24

1 error, 1 warning, 0 others

JSP Problem (1 item)

5. Set the spring and Hibernate dependencies.
    a. Open POM file
    b. Add the following spring dependencies under the tag <dependencies>

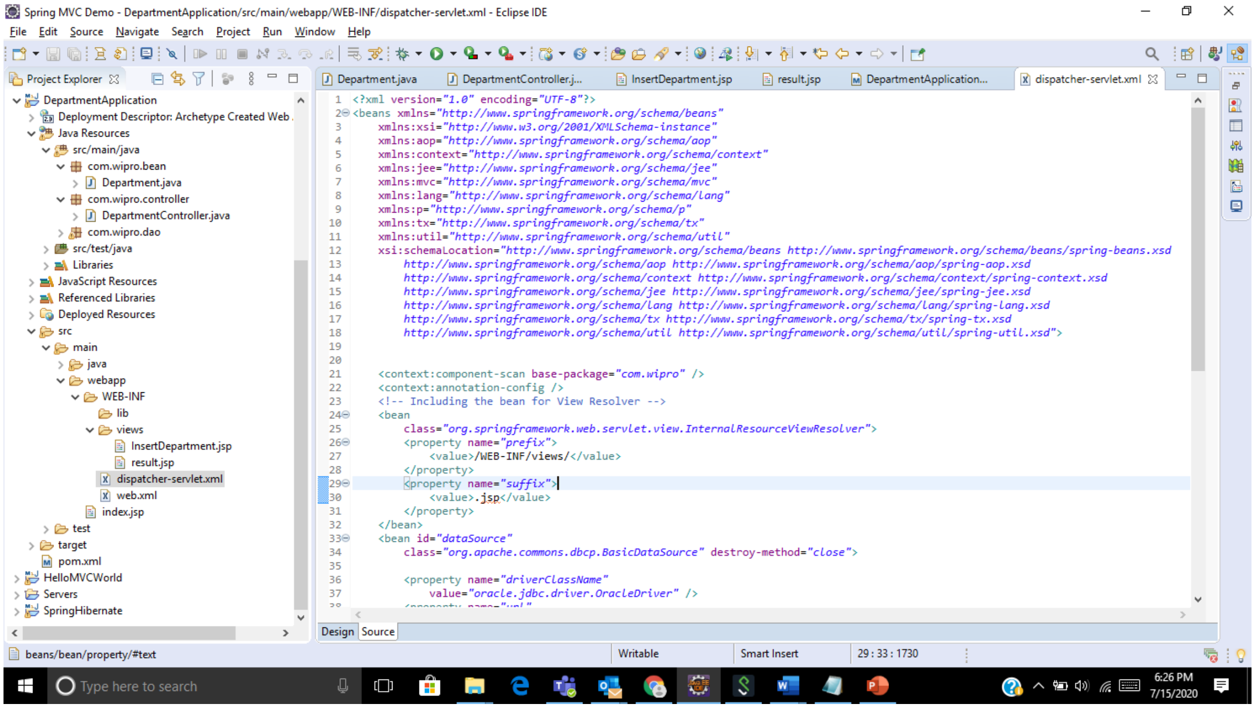
```
        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-core</artifactId>
                <version>5.2.1.RELEASE</version>
        </dependency>

        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-web</artifactId>
                <version>5.2.1.RELEASE</version>
        </dependency>

        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-webmvc</artifactId>
                <version>5.2.1.RELEASE</version>
        </dependency>
        <!-- dependencies required for Hibernate ORM Integration -->
        <dependency>
                <groupId>org.hibernate</groupId>
                <artifactId>hibernate-core</artifactId>
                <version>5.4.9.Final</version>
        </dependency>

        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-orm</artifactId>
                <version>5.2.1.RELEASE</version>
        </dependency>


        <dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-tx</artifactId>
                <version>5.2.6.RELEASE</version>
        </dependency>
        <dependency>
                <groupId>commons-dbcp</groupId>
                <artifactId>commons-dbcp</artifactId>
                <version>1.2.2</version>
        </dependency>

        <dependency>
                <groupId>com.oracle.database.jdbc</groupId>
                <artifactId>ojdbc6</artifactId>
                <version>11.2.0.4</version>
        </dependency>
```

6. Set the dispatcher Servlet.
   a. Edit the **web.xml** file from project path : src->main->webapp->WEB-INF->web.xml
   b. Include <servlet> and <servlet-mapping>

```
    <servlet>
       <servlet-name>dispatcher</servlet-name>
      <servlet-class>org.springframework.web.servlet.DispatcherServlet
       </servlet-class>
       </servlet>
       <servlet-mapping>
              <servlet-name>dispatcher</servlet-name>
              <url-pattern>/</url-pattern>
      </servlet-mapping>
```

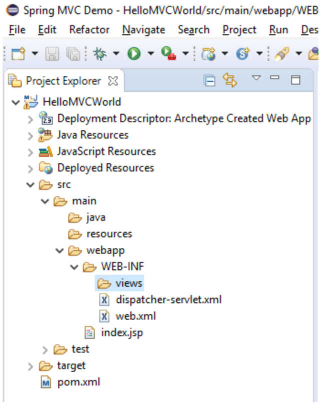
7. Create the Spring configuration file named dispatcher-servlet.xml

```
<?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:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:lang="http://www.springframework.org/schema/lang"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee.xsd
        http://www.springframework.org/schema/lang
http://www.springframework.org/schema/lang/spring-lang.xsd
        http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
        http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util.xsd">


    <context:component-scan base-package="com.wipro" />
    <context:annotation-config />
    <!-- Including the bean for View Resolver -->
    <bean

    class="org.springframework.web.servlet.view.InternalResourceViewResolver">
          <property name="prefix">
                 <value>/WEB-INF/views/</value>
          </property>
          <property name="suffix">
                 <value>.jsp</value>
        </property>
    </bean>
    <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

        <property name="driverClassName"
            value="oracle.jdbc.driver.OracleDriver" />
        <property name="url"
            value="jdbc:oracle:thin:@localhost:1521:orcl" />
        <property name="username" value="scott" />
        <property name="password" value="tiger" />
    </bean>

    <bean id="sessionFactory"
        class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">
        <property name="dataSource" ref="dataSource" />
        <property name="hibernateProperties">
            <props>
                <prop
key="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</prop>
                <prop key="hibernate.show_sql">true</prop>
                <prop key="hibernate.hbm2ddl.auto">create</prop>
            </props>
        </property>
        <property name="packagesToScan" value="com.wipro.bean" />


    </bean>
    <bean id="hibernateTemplate"
class="org.springframework.orm.hibernate5.HibernateTemplate">
     <property name="sessionFactory" ref="sessionFactory"></property>
     </bean>

    <tx:annotation-driven />

    <bean id="transactionManager"
        class="org.springframework.orm.hibernate5.HibernateTransactionManager">
        <property name="sessionFactory" ref="sessionFactory" />
    </bean>

</beans>
```



```xml
<?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:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jee="http://www.springframework.org/schema/jee"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xmlns:lang="http://www.springframework.org/schema/lang"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:util="http://www.springframework.org/schema/util"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee.xsd
        http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang.xsd
        http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd
        http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util.xsd">


    <context:component-scan base-package="com.wipro" />
    <context:annotation-config />
    <!-- Including the bean for View Resolver -->
    <bean
        class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix">
            <value>/WEB-INF/views/</value>
        </property>
        <property name="suffix">
            <value>.jsp</value>
        </property>
    </bean>
    <bean id="dataSource"
        class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">

        <property name="driverClassName"
            value="oracle.jdbc.driver.OracleDriver" />
```

8. Create the views folder under WEB-INF for defining the views
    a. Right click on WEB-INF folder NEW -> Folder (Enter Name as **views**) - > finish.



9. Create the entity bean class called Department under the package com.wipro.bean

```java
package com.wipro.bean;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
```

```
@Table(name="MYDept")
public class Department {
        @Id
        private int deptno;
        @Column(length = 10)
        private String dname;
        @Column(length = 10)
        private String loc;
        public Department() {

        }
        public Department(int deptno, String dname, String loc) {
                super();
                this.deptno = deptno;
                this.dname = dname;
                this.loc = loc;
        }
        public int getDeptno() {
                return deptno;
        }
        public void setDeptno(int deptno) {
                this.deptno = deptno;
        }
        public String getDname() {
                return dname;
        }
        public void setDname(String dname) {
                this.dname = dname;
        }
        public String getLoc() {
                return loc;
        }
        public void setLoc(String loc) {
                this.loc = loc;
        }
        @Override
        public String toString() {
                return "Department [deptno=" + deptno + ", dname=" + dname + ", loc=" + loc + "]";
        }
}
```

10. Edit the Index.jsp page to request for inserting operation for Department

```
<html>
<body>
<h2>Department Actions</h2>
<pre>
<a href="PreInsertDepartment">Insert Department</a>
</pre>
</body>
</html>
```

11. Create the DataAccess Layer called DepartmentDao under com.wipro.dao

```
package com.wipro.dao;

import java.util.List;

import javax.transaction.Transactional;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.query.Query;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.orm.hibernate5.HibernateTemplate;
import org.springframework.stereotype.Repository;

import com.wipro.bean.Department;

@Repository
@Transactional
public class DepartmentDao {
        @Autowired
        SessionFactory sessionFactory;

        @Autowired
        HibernateTemplate hibernateTemplate;

        public int getDepartmentId() {
                int id=0;
                Session session = sessionFactory.openSession();
                Query<Department> qry = session.createQuery("Select max(d.deptno) from
Department d");
                List l=qry.list();
                System.out.println("L value"+l + " : " );
                if(l!=null && l.size()>0) {
                        Object b=l.get(0);
                if(b!=null)
```

```
            id=(Integer) b;
        }
        session.close();
        return id+10;
    }
    public boolean insertDepatment(Department department) {
        hibernateTemplate.persist(department);
        return true;
    }
}
```

12. Create the controller class named **DepartmentController** under the package called com.wipro.controller

```
package com.wipro.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;

import com.wipro.bean.Department;
import com.wipro.dao.DepartmentDao;

@Controller
public class DepartmentController {

    @Autowired
    DepartmentDao dao;

    @RequestMapping("PreInsertDepartment")
    public ModelAndView preInsert() {
            Department department = new Department();
            department.setDeptno(dao.getDepartmentId());
            ModelAndView mv = new
ModelAndView("InsertDepartment","department",department);
            return mv;
    }

    @RequestMapping("InsertDepartment")
    public ModelAndView insertDepartment(@ModelAttribute("department") Department dept) {

            ModelAndView mv = new ModelAndView("result","department",dept);
            if(dao.insertDepatment(dept))
```

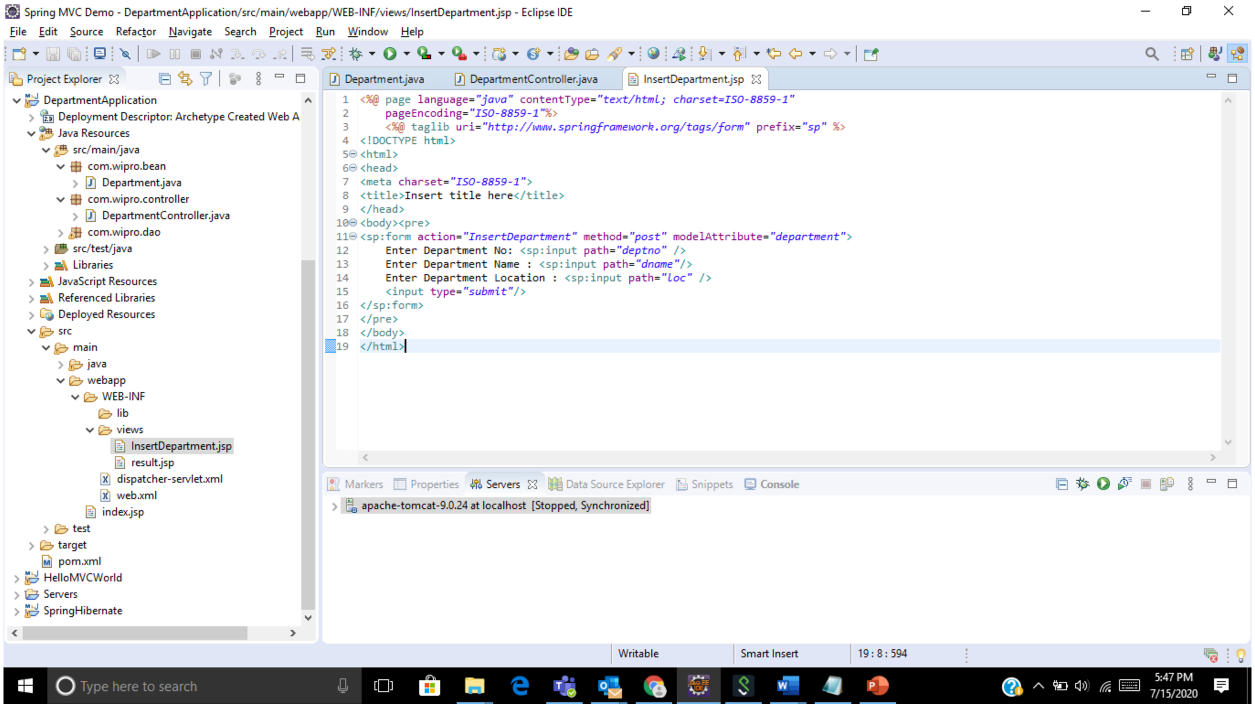
```
                    mv.addObject("msg", "Inserted Successfully");
            else
                    mv.addObject("msg", "Insert Failed");
            return mv;
    }
}
```

13. Create a new jsp page under WEB-INF -> views called InsertDepartment.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ taglib uri="http://www.springframework.org/tags/form" prefix="sp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body><pre>
<sp:form action="InsertDepartment" method="post" modelAttribute="department">
   Enter Department No: <sp:input path="deptno" />
   Enter Department Name : <sp:input path="dname"/>
   Enter Department Location : <sp:input path="loc" />
   <input type="submit"/>
</sp:form>
</pre>
</body>
</html>
```

14. Create a new jsp page under WEB-INF -> views called result.jsp

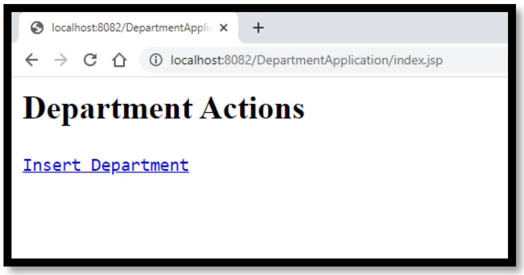
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<pre>
${msg}
Department [ ${department.deptno } , ${department.dname } , ${department.loc } ]

</pre>
</body>
</html>
```



```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
    <%@ taglib uri="http://www.springframework.org/tags/form" prefix="sp" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body><pre>
<sp:form action="InsertDepartment" method="post" modelAttribute="department">
    Enter Department No: <sp:input path="deptno" />
    Enter Department Name : <sp:input path="dname"/>
    Enter Department Location : <sp:input path="loc" />
    <input type="submit"/>
</sp:form>
</pre>
</body>
</html>
```

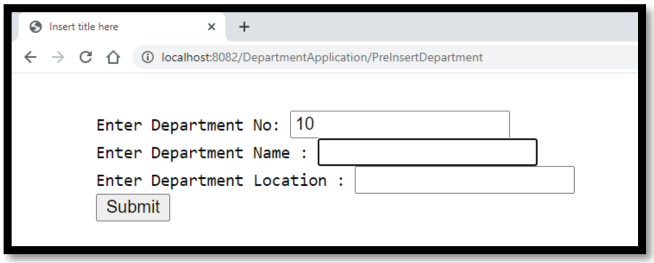
apache-tomcat-9.0.24 at localhost [Stopped, Synchronized]

15. Run the Application from index.jsp file

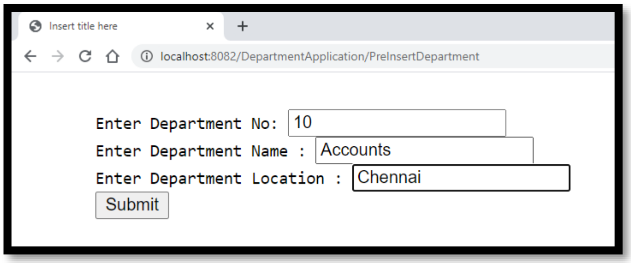

localhost:8082/DepartmentAppli
localhost:8082/DepartmentApplication/index.jsp

Department Actions

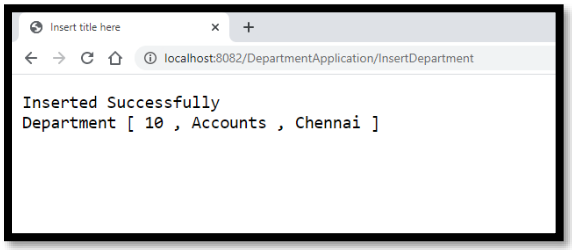
Insert Department



Enter Department No: 10
Enter Department Name :
Enter Department Location :
Submit

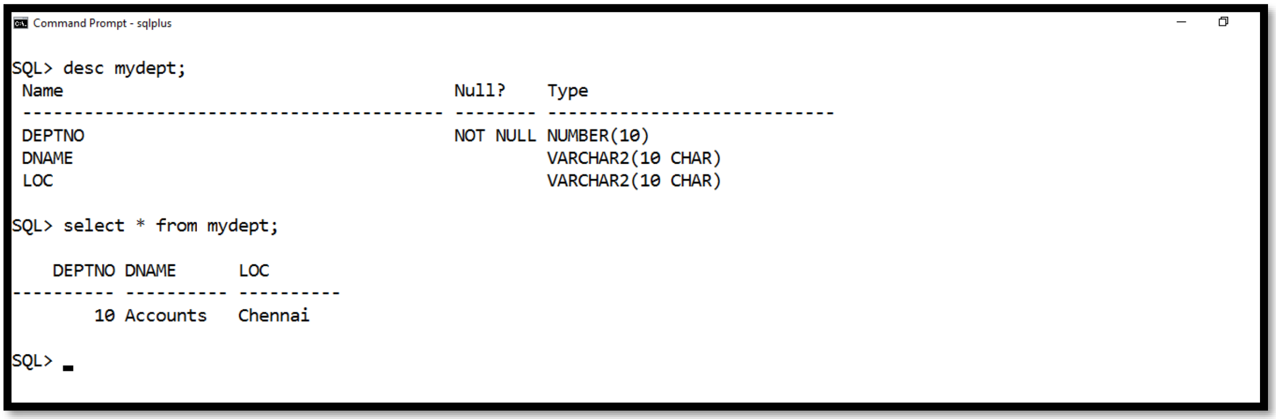


```
Enter Department No: 10
Enter Department Name : Accounts
Enter Department Location : Chennai
```



```
Inserted Successfully
Department [ 10 , Accounts , Chennai ]
```

16. Now connect to the database and check for the table creation and inserti



```
SQL> desc mydept;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPTNO                                    NOT NULL NUMBER(10)
 DNAME                                              VARCHAR2(10 CHAR)
 LOC                                                VARCHAR2(10 CHAR)

SQL> select * from mydept;

    DEPTNO DNAME      LOC
---------- ---------- ----------
        10 Accounts   Chennai

SQL> _
```