use users110;

DROP TABLE IF EXISTS `users`;

/\*!40101 SET @saved\_cs\_client = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `users` (

`ID` bigint(20) NOT NULL AUTO\_INCREMENT,

`firstname` varchar(100) DEFAULT NULL,

`lastname` varchar(100) DEFAULT NULL,

`email` varchar(100) DEFAULT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `eproduct`

--

LOCK TABLES `users` WRITE;

/\*!40000 ALTER TABLE `users` DISABLE KEYS \*/;

INSERT INTO `users` VALUES (1,'Alexander','Orlov','it9@inbox.ru'),(2,'Richard','Sapogov','sapog@ricci.com'),(3,'Ekaterina','Kotikovna','katya@google.com');

/\*!40000 ALTER TABLE `users` ENABLE KEYS \*/;

UNLOCK TABLES;

package alex.springmvc.controller;  
  
import java.util.List;  
  
import alex.springmvc.entity.User;  
import alex.springmvc.service.UserService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
  
  
@Controller  
public class UserController {  
  
 @Autowired  
 private UserService userService;  
  
 @GetMapping("/")  
 public String getIndexForm(){  
 return "index";  
  
 }  
  
 @GetMapping("/list")  
 public String listUsers(Model model) {  
 List <User> users = userService.getUsers();  
 model.addAttribute("users", users);  
 return "list-users";  
 }  
  
 @GetMapping("/showForm")  
 public String showFormForAdd(Model model) {  
 User user = new User();  
 model.addAttribute("user", user);  
 return "user-form";  
 }  
  
 @PostMapping("/saveUser")  
 public String saveUser(@ModelAttribute("user") User user) {  
 userService.saveUser(user);  
 return "redirect:/list";  
 }  
  
 @GetMapping("/updateForm")  
 public String showFormForUpdate(@RequestParam("userId") int id,  
 Model model) {  
 User user = userService.getUser(id);  
 model.addAttribute("user", user);  
 return "user-form";  
 }  
  
 @GetMapping("/delete")  
 public String deleteUser(@RequestParam("userId") int id) {  
 userService.deleteUser(id);  
 return "redirect:/list";  
 }  
}

package alex.springmvc.dao;  
  
import java.util.List;  
  
import alex.springmvc.entity.User;  
  
public interface UserDAO {  
  
 public List <User> getUsers();  
  
 public void saveUser(User user);  
  
 public User getUser(int id);  
  
 public void deleteUser(int id);  
}

package alex.springmvc.dao;  
  
  
import java.util.List;  
  
import javax.persistence.Query;  
import javax.persistence.criteria.CriteriaBuilder;  
import javax.persistence.criteria.CriteriaQuery;  
import javax.persistence.criteria.Root;  
  
import alex.springmvc.entity.User;  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Repository;  
  
  
@Repository  
public class UserDAOImpl implements UserDAO {  
  
 @Autowired  
 private SessionFactory sessionFactory;  
  
 @Override  
 public List<User> getUsers() {  
 Session session = sessionFactory.getCurrentSession();  
 CriteriaBuilder cb = session.getCriteriaBuilder();  
 CriteriaQuery<User> cq = cb.createQuery(User.class);  
 Root<User> root = cq.from(User.class);  
 cq.select(root);  
 Query query = session.createQuery(cq);  
 return query.getResultList();  
 }  
  
 @Override  
 public void deleteUser(int id) {  
 Session session = sessionFactory.getCurrentSession();  
 User user = session.byId(User.class).load(id);  
 session.delete(user);  
 }  
  
 @Override  
 public void saveUser(User user) {  
 Session currentSession = sessionFactory.getCurrentSession();  
 currentSession.saveOrUpdate(user);  
 }  
  
 @Override  
 public User getUser(int id) {  
 Session currentSession = sessionFactory.getCurrentSession();  
 User user = currentSession.get(User.class, id);  
 return user;  
 }  
}

package alex.springmvc.entity;  
import lombok.\*;  
  
import javax.persistence.Column;  
import javax.persistence.Entity;  
import javax.persistence.GeneratedValue;  
import javax.persistence.GenerationType;  
import javax.persistence.Id;  
import javax.persistence.Table;  
  
@Entity  
@Table(name = "users")  
@Getter  
@Setter  
@NoArgsConstructor  
public class User {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "ID")  
 private int id;  
  
 @Column(name = "firstname")  
 private String firstName;  
  
 @Column(name = "lastname")  
 private String lastName;  
  
 @Column(name = "email")  
 private String email;  
  
  
 @Override  
 public String toString() {  
 return "Customer [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email + "]";  
 }  
}

package alex.springmvc.service;  
import java.util.List;  
  
import alex.springmvc.entity.User;  
  
public interface UserService {  
  
 public List <User> getUsers();  
  
 public void saveUser(User user);  
  
 public User getUser(int id);  
  
 public void deleteUser(int id);  
  
}

package alex.springmvc.service;  
  
import java.util.List;  
  
import alex.springmvc.dao.UserDAO;  
import alex.springmvc.entity.User;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
  
  
@Service  
public class UserServiceImpl implements UserService {  
  
 @Autowired  
 private UserDAO userDAO;  
  
 @Override  
 @Transactional  
 public List <User> getUsers() {  
 return userDAO.getUsers();  
 }  
  
 @Override  
 @Transactional  
 public void saveUser(User user) {  
 userDAO.saveUser(user);  
 }  
  
 @Override  
 @Transactional  
 public User getUser(int id) {  
 return userDAO.getUser(id);  
 }  
  
 @Override  
 @Transactional  
 public void deleteUser(int id) {  
 userDAO.deleteUser(id);  
 }  
}

*<%--  
 Created by IntelliJ IDEA.  
 User: Александр  
 Date: 13.5.22  
 Time: 16:12  
 To change this template use File | Settings | File Templates.  
--%>*<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Searching for a Specific User </title>  
</head>  
  
  
<body>  
  
<div align="left">  
 <h2> Searching for Specific User</h2>  
 <form name=Searchung action="updateForm">  
 <p>User ID: <input name="userId" /></p>  
 <button>Submit</button>  
 </form>  
</div>  
  
</body>  
</html>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"  
 pageEncoding="ISO-8859-1"%>  
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  
 <title>javaguides.net</title>  
 <link href="<c:url value="/resources/css/bootstrap.min.css" />"  
 rel="stylesheet">  
 <script src="<c:url value="/resources/js/jquery-1.11.1.min.js" />"></script>  
 <script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>  
</head>  
<body>  
<div class="container">  
 <div class="col-md-offset-1 col-md-10">  
 <h2>Users manager</h2>  
 <hr />  
  
 <input type="button" value="Add User"  
 onclick="window.location.href='showForm'; return false;"  
 class="btn btn-primary" />  
 <br/><br/>  
 <div class="panel panel-info">  
 <div class="panel-heading">  
 <div class="panel-title">Users List</div>  
 </div>  
 <div class="panel-body">  
 <table class="table table-striped table-bordered">  
 <tr>  
 <th>First Name</th>  
 <th>Last Name</th>  
 <th>Email</th>  
 <th>Action</th>  
 </tr>  
  
 *<!-- loop over and print our customers -->* <c:forEach var="tempUser" items="${users}">  
  
 *<!-- construct an "update" link with customer id -->* <c:url var="updateLink" value="/updateForm">  
 <c:param name="userId" value="${tempUser.id}" />  
 </c:url>  
  
 *<!-- construct an "delete" link with customer id -->* <c:url var="deleteLink" value="/delete">  
 <c:param name="userId" value="${tempUser.id}" />  
 </c:url>  
 <tr>  
 <td>${tempUser.firstName}</td>  
 <td>${tempUser.lastName}</td>  
 <td>${tempUser.email}</td>  
 <td>  
 *<!-- display the update link -->* <a href="${updateLink}">Update</a>  
 | <a href="${deleteLink}"  
 onclick="if (!(*confirm*('Are you sure you want to delete this customer?'))) return false">Delete</a>  
 </td>  
 </tr>  
 </c:forEach>  
 </table>  
 </div>  
 </div>  
 </div>  
</div>  
</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="form"%>  
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="ISO-8859-1">  
 <title>Add/Update user </title>  
 <link href="<c:url value="/resources/css/bootstrap.min.css" />" rel="stylesheet">  
 <script src="<c:url value="/resources/js/jquery-1.11.1.min.js" />"></script>  
 <script src="<c:url value="/resources/js/bootstrap.min.js" />"></script>  
  
</head>  
<body>  
<div class="container">  
 <div class="col-md-offset-2 col-md-7">  
 <h2 class="text-center">Using Spring MVC 5 + Hibernate 5 + JSP + MySQL </h2>  
 <div class="panel panel-info">  
 <div class="panel-heading">  
 <div class="panel-title">Add/Update User</div>  
 </div>  
 <div class="panel-body">  
 <form:form action="saveUser" cssClass="form-horizontal"  
 method="post" modelAttribute="user">  
  
 *<!-- need to associate this data with user id -->* <form:hidden path="id" />  
  
 <div class="form-group">  
 <label for="firstName" class="col-md-3 control-label">First  
 Name</label>  
 <div class="col-md-9">  
 <form:input path="firstName" cssClass="form-control" />  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="lastName" class="col-md-3 control-label">Last  
 Name</label>  
 <div class="col-md-9">  
 <form:input path="lastName" cssClass="form-control" />  
 </div>  
 </div>  
  
 <div class="form-group">  
 <label for="email" class="col-md-3 control-label">Email</label>  
 <div class="col-md-9">  
 <form:input path="email" cssClass="form-control" />  
 </div>  
 </div>  
  
 <div class="form-group">  
 *<!-- Button -->* <div class="col-md-offset-3 col-md-9">  
 <form:button cssClass="btn btn-primary">Submit</form:button>  
 </div>  
 </div>  
  
 </form:form>  
 </div>  
 </div>  
 </div>  
</div>  
</body>  
</html>

*<?*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"  
 xmlns:tx="http://www.springframework.org/schema/tx"  
 xmlns:mvc="http://www.springframework.org/schema/mvc"  
 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  
 http://www.springframework.org/schema/mvc  
 http://www.springframework.org/schema/mvc/spring-mvc.xsd  
 http://www.springframework.org/schema/tx  
 http://www.springframework.org/schema/tx/spring-tx.xsd">  
 *<!-- Add support for component scanning -->* <context:component-scan base-package="alex.springmvc" />  
 *<!-- Add support for conversion, formatting and validation support -->* <mvc:annotation-driven/>  
 *<!-- Define Spring MVC view resolver -->* <bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name="prefix" value="/WEB-INF/views/" />  
 <property name="suffix" value=".jsp" />  
 </bean>  
 *<!-- Step 1: Define Database DataSource / connection pool -->* <bean id="myDataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"  
 destroy-method="close">  
 <property name="driverClass" value="com.mysql.jdbc.Driver" />  
 <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/users110?useSSL=false" />  
 <property name="user" value="root" />  
 <property name="password" value="vestaeagle" />  
 *<!-- these are connection pool properties for C3P0 -->* <property name="minPoolSize" value="5" />  
 <property name="maxPoolSize" value="20" />  
 <property name="maxIdleTime" value="30000" />  
 </bean>  
 *<!-- Step 2: Setup Hibernate session factory -->* <bean id="sessionFactory"  
 class="org.springframework.orm.hibernate5.LocalSessionFactoryBean">  
 <property name="dataSource" ref="myDataSource" />  
 <property name="packagesToScan" value="alex.springmvc.entity" />  
 <property name="hibernateProperties">  
 <props>  
 <prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>  
 <prop key="hibernate.show\_sql">true</prop>  
 </props>  
 </property>  
 </bean>  
 *<!-- Step 3: Setup Hibernate transaction manager -->* <bean id="myTransactionManager"  
 class="org.springframework.orm.hibernate5.HibernateTransactionManager">  
 <property name="sessionFactory" ref="sessionFactory"/>  
 </bean>  
 *<!-- Step 4: Enable configuration of transactional behavior based on annotations -->* <tx:annotation-driven transaction-manager="myTransactionManager" />  
 *<!-- Add support for reading web resources: css, images, js, etc ... -->* <mvc:resources location="/resources/" mapping="/resources/\*\*"></mvc:resources>  
</beans>

*<?*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://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd" id="WebApp\_ID" version="3.1">  
 <display-name>spring-mvc-crud-demo</display-name>  
 <welcome-file-list>  
 <welcome-file>index.jsp</welcome-file>  
 <welcome-file>index.html</welcome-file>  
 </welcome-file-list>  
 <servlet>  
 <servlet-name>dispatcher</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>/WEB-INF/spring-config.xml</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>dispatcher</servlet-name>  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
</web-app>

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>alex</groupId>  
 <artifactId>task110</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <name>task110</name>  
 <packaging>war</packaging>  
  
 <properties>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <junit.version>5.7.1</junit.version>  
 <failOnMissingWebXml>false</failOnMissingWebXml>  
 <spring.version>5.2.21.RELEASE</spring.version>  
 <hibernate.version>5.3.5.Final</hibernate.version>  
 <hibernate.validator>6.0.17.Final</hibernate.validator>  
 <c3p0.version>0.9.1.2</c3p0.version>  
 <tld.version>1.1.2</tld.version>  
 <servlets.version>4.0.1</servlets.version>  
 <jsp.version>2.1</jsp.version>  
 <hsqldb.version>1.8.0.10</hsqldb.version>  
 </properties>  
 <dependencies>  
 *<!-- Spring MVC Dependency -->* <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 *<!-- Spring ORM -->* <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-orm</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 *<!-- Hibernate Core -->* <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-core</artifactId>  
 <version>5.6.3.Final</version>  
 </dependency>  
 *<!-- Hibernate Validator -->* <dependency>  
 <groupId>org.hibernate</groupId>  
 <artifactId>hibernate-validator</artifactId>  
 <version>${hibernate.validator}</version>  
 </dependency>  
 *<!-- JSTL Dependency -->* <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
 <dependency>  
 <groupId>taglibs</groupId>  
 <artifactId>standard</artifactId>  
 <version>${tld.version}</version>  
 </dependency>  
 *<!-- Servlet Dependency -->* <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>${servlets.version}</version>  
 <scope>provided</scope>  
 </dependency>  
 *<!-- JSP Dependency -->* <dependency>  
 <groupId>javax.servlet.jsp</groupId>  
 <artifactId>jsp-api</artifactId>  
 <version>${jsp.version}</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.27</version>  
 </dependency>  
 <dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
 <version>RELEASE</version>  
 <scope>compile</scope>  
 </dependency>  
  
 *<!-- https://mvnrepository.com/artifact/c3p0/c3p0 -->* <dependency>  
 <groupId>c3p0</groupId>  
 <artifactId>c3p0</artifactId>  
 <version>${c3p0.version}</version>  
 </dependency>  
  
 </dependencies>  
 <build>  
 <finalName>SpringSetup</finalName>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <configuration>  
 <source>15</source>  
 <target>15</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>