Phase – 2 Practice Project: Assisted Practice

28. Demonstrate integration of Hibernate with spring.

• Code

```
✓ pom.xml
```

```
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
http://maven.apache.org/maven-v4 0 0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>SpringHibernateWeb</groupId>
 <artifactId>SpringHibernateWeb</artifactId>
 <packaging>war</packaging>
 <version>0.0.1-SNAPSHOT
 <name>SpringHibernateWeb Maven Webapp</name>
 <url>http://maven.apache.org</url>
   <!-- JBoss repository for Hibernate -->
   <repositories>
       <repository>
           <id>JBoss repository</id>
<url>http://repository.jboss.org/nexus/content/groups/public/</url>
       </repository>
   </repositories>
  properties>
<org.springframework.version>3.0.5.RELEASE</org.springframework.version>
 </properties>
  <dependencies>
   <dependency>
     <groupId>junit
     <artifactId>junit</artifactId>
     <version>3.8.1
     <scope>test</scope>
   </dependency>
    <dependency>
   <groupId>org.hibernate.javax.persistence</groupId>
   <artifactId>hibernate-jpa-2.1-api</artifactId>
   <version>1.0.0.Final
</dependency>
   <dependency>
     <groupId>org.springframework
     <artifactId>spring-core</artifactId>
     <version>${org.springframework.version}</version>
   </dependency>
   <dependency>
     <groupId>org.springframework
     <artifactId>spring-expression</artifactId>
```

```
<version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework
 <artifactId>spring-beans</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-context</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework
 <artifactId>spring-context-support</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-jdbc</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-orm</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework
 <artifactId>spring-web</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
 <groupId>org.springframework
 <artifactId>spring-webmvc</artifactId>
 <version>${org.springframework.version}</version>
</dependency>
<dependency>
  <groupId>log4j
  <artifactId>log4j</artifactId>
  <version>1.2.16
  <scope>runtime</scope>
</dependency>
<dependency>
   <groupId>org.hibernate
   <artifactId>hibernate-core</artifactId>
   <version>3.6.3.Final
```

```
</dependency>
    <dependency>
       <groupId>javassist
       <artifactId>javassist</artifactId>
       <version>3.12.1.GA
    </dependency>
    <dependency>
       <groupId>taglibs
       <artifactId>standard</artifactId>
       <version>1.1.2
       <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>commons-dbcp
     <artifactId>commons-dbcp</artifactId>
      <version>1.4</version>
    </dependency>
   <dependency>
       <qroupId>mysql
       <artifactId>mysql-connector-java</artifactId>
       <version>5.1.9
    </dependency>
  </dependencies>
  <build>
    <finalName>SpringHibernateWeb</finalName>
  </build>
</project>
✓ Database Code
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD UNIQUE CHECKS=@@UNIQUE CHECKS, UNIQUE CHECKS=0 */;
/*!40014 SET @OLD FOREIGN KEY CHECKS=@@FOREIGN KEY CHECKS.
FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO'
/*!40111 SET @OLD SQL NOTES=@@SQL NOTES, SQL NOTES=0 */;
```

*/;

```
-- Table structure for table `eproduct`
DROP TABLE IF EXISTS 'eproduct';
/*!40101 SET @saved_cs_client = @ @character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE 'eproduct' (
`ID` bigint(20) NOT NULL AUTO_INCREMENT,
 `name` varchar(100) DEFAULT NULL,
 'price' decimal(10,2) DEFAULT NULL,
 `date_added` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
 `parts_hdd` varchar(10) DEFAULT NULL,
 `parts_cpu` varchar(10) DEFAULT NULL,
 `parts_ram` varchar(10) 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 `eproduct` WRITE;
/*!40000 ALTER TABLE `eproduct` DISABLE KEYS */;
INSERT INTO 'eproduct' VALUES (1,'HP Laptop ABC',21900.00,'2019-06-04 07:18:57','2 Gb
HDD', 'AMD Phenom', '4 Gb'), (2, 'Acer Laptop ABC', 23300.00, '2019-06-04 07:19:07', '500 Gb HDD', 'Core-
i7','4 Gb'),(3,'Lenovo Laptop ABC',33322.00,'2019-06-04 07:19:19','1 Tb HDD','Core-i7','8 Gb');
/*!40000 ALTER TABLE `eproduct` ENABLE KEYS */;
UNLOCK TABLES:
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
```

```
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

✓ EProductEntity.java

```
package com.ecommerce.entity;
import java.math.BigDecimal;
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name= "eproduct")
public class EProductEntity {
        @Id @GeneratedValue
        @Column(name = "ID")
        private long ID;
        @Column(name = "name")
        private String name;
        @Column(name = "price")
        private BigDecimal price;
        @Column(name = "date added")
        private Date dateAdded;
        public long getID() {return this.ID; }
        public String getName() { return this.name;}
        public BigDecimal getPrice() { return this.price;}
        public Date getDateAdded() { return this.dateAdded;}
        public void setID(long id) { this.ID = id;}
        public void setName(String name) { this.name = name;}
        public void setPrice(BigDecimal price) { this.price = price;}
        public void setDateAdded(Date date) { this.dateAdded = date;}
}
```

✓ EproductDAO.java

```
package com.ecommerce.dao;
import java.util.List;
import org.hibernate.SessionFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.ecommerce.entity.EProductEntity;
@Repository
public class EProductDAO {
        @Autowired
   private SessionFactory sessionFactory;
        @SuppressWarnings("unchecked")
        public List<EProductEntity> getAllProducts() {
this.sessionFactory.getCurrentSession().createQuery("from
EProducts").list();
        }
}
✓ EProductController.java
package com.ecommerce.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import com.ecommerce.entity.EProductEntity;
import com.ecommerce.service.EProductService;
@Controller
public class EProductController {
        @Autowired
            private EProductService eproductService;
            @RequestMapping(value = "/productList", method =
RequestMethod.GET)
            public String listProducts(ModelMap map)
                map.addAttribute("eproduct", new EProductEntity());
                map.addAttribute("productList",
eproductService.getAllProducts());
```

```
return "productList";
}
```

✓ eproduct-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"</pre>
    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: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:annotation-config />
    <context:component-scan base-package="com.ecommerce.controller" />
    <bean id="jspViewResolver"</pre>
class="org.springframework.web.servlet.view.InternalResourceViewResolver")
        cproperty name="viewClass"
value="org.springframework.web.servlet.view.JstlView"></property>
        cproperty name="prefix" value="/WEB-INF/view/"></property>
        cproperty name="suffix" value=".jsp"></property>
    </bean>
    <bean id="messageSource"</pre>
class="org.springframework.context.support.ReloadableResourceBundleMessa"
geSource">
        cproperty name="basename" value="classpath:messages"></property>
        cproperty name="defaultEncoding" value="UTF-8">
    </bean>
    <bean id="propertyConfigurer"</pre>
class="org.springframework.beans.factory.config.PropertyPlaceholderConfi">class="org.springframework.beans.factory.config.PropertyPlaceholderConfi">class="org.springframework.beans.factory.config.PropertyPlaceholderConfi"
gurer"
        p:location="/WEB-INF/jdbc.properties"></bean>
    <bean id="dataSource"</pre>
```

```
class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close"
       p:driverClassName="${jdbc.driverClassName}"
       p:url="${jdbc.databaseurl}" p:username="${jdbc.username}"
       p:password="${jdbc.password}"></bean>
    <bean id="sessionFactory"</pre>
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean">
       cproperty name="dataSource" ref="dataSource"></property>
       property name="configLocation">
           <value>classpath:hibernate.cfg.xml</value>
       </property>
       cproperty name="configurationClass">
           <value>org.hibernate.cfg.AnnotationConfiguration</value>
       property name="hibernateProperties">
           ops>
               key="hibernate.dialect">${jdbc.dialect}
               </props>
       </property>
    </bean>
    <bean id="eproductDAO" class="com.ecommerce.dao.EProductDAO"></bean>
    <bean id="eproductService"</pre>
class="com.ecommerce.service.EProductService"></bean>
    <tx:annotation-driven />
    <bean id="transactionManager"</pre>
class="org.springframework.orm.hibernate3.HibernateTransactionManager">
       cproperty name="sessionFactory" ref="sessionFactory">
    </bean>
</beans>
✓ jdbc.properties
jdbc.driverClassName=com.mysql.jdbc.Driver
jdbc.dialect=org.hibernate.dialect.MySQLDialect
jdbc.databaseurl=jdbc:mysql://127.0.0.1:3306/ecommerce
jdbc.username=root
jdbc.password=root
✓ productList.jsp
<%@taglib uri="http://www.springframework.org/tags" prefix="spring"%>
<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<html>
    <head>
       <title>Spring With Hibernate</title>
   </head>
   <body>
    <h3>Product List </h3>
    <c:if test="${!empty productList}">
   Name
```

```
Price
       Date Added
    <c:forEach items="${productList}" var="product">
       ${product.name} 
           ${product.price}
           ${product.date added}
       </c:forEach>
    </c:if>
   </body>
</html>
✓ index.jsp
<html>
<body>
<h2>Spring With Hibernate</h2>
<a href="/productList">Product List</a>
</body>
</html>
✓ hibernate.cfg.xml
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC</pre>
"-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <mapping class="com.ecommerce.entity.EProductEntity"></mapping>
    </session-factory>
</hibernate-configuration>
✓ web.xml
<!DOCTYPE web-app PUBLIC</pre>
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app 2 3.dtd" >
<web-app>
  <display-name>Archetype Created Web Application</display-name>
   <welcome-file-list>
       <welcome-file>/WEB-INF/view/index.jsp</welcome-file>
   </welcome-file-list>
    <servlet>
       <servlet-name>eproduct</servlet-name>
       <servlet-class>
           org.springframework.web.servlet.DispatcherServlet
       </servlet-class>
```

```
<load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>eproduct</servlet-name>
        <url-pattern>/</url-pattern>
   </servlet-mapping>
    <context-param>
       <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/eproduct-servlet.xml</param-value>
    </context-param>
    stener>
        stener-
class>org.springframework.web.context.ContextLoaderListener/listener-
class>
   </listener>
</web-app>
```