

Program 1 : Create a bean class called product (id, name, rate, qty, launch_date).Create object through dependency injection.

// Product.java

```
package com.hcl.spring;

public class Product
{
    // attributes declaration
    private int id;
    private String name;
    private float rate;
    private int qty;
    private String launch_date;

    public Product() {
        super();
        // TODO Auto-generated constructor stub
    }

    // parameterized constructor
    public Product(int id, String name, float rate, int
qty, String launch_date) {
        super();
    }
}
```

```
        this.id = id;

        this.name = name;

        this.rate = rate;

        this.qty = qty;

        this.launch_date = launch_date;
    }

    // getters and setters for attributes
    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    public float getRate() {
        return rate;
    }
}
```

```

}

public void setPrice(float rate) {
    this.rate = rate;
}

public int getQty() {
    return qty;
}

public void setQty(int qty) {
    this.qty = qty;
}

public String getLaunch_date() {
    return launch_date;
}

public void setLaunch_date(String launch_date) {
    this.launch_date = launch_date;
}

// user-defined init method
public void userInit()
{
    System.out.println("object created...");
}

```

```

// user-defined destroy method

public void userDestroy()
{
    System.out.println("garbaged
successfully...");
}

// To string method for displaying output
@Override
public String toString() {
    return "Product [id=" + id + ", name=" + name
+ ", rate=" + rate + ", qty=" + qty + ", launch_date="
+ launch_date + "];"
}

}

```

// product.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"

xsi:schemaLocation =
"http://www.springframework.org/schema/beans"
http://www.springframework.org/schema/beans/spring-

```

beans-3.0.xsd">

```
<bean id="prod1" class ="com.hcl.spring.Product" >
<property name="id" value="101" />
<property name="name" value="Airtel" />
<property name="rate" value="678.89" />
<property name="qty" value="28" />
<property name="launch_date" value="26-08-1998" />
</bean>

<bean id="prod2" class ="com.hcl.spring.Product">
<property name="id" value="102" />
<property name="name" value="Airtel" />
<property name="rate" value="800" />
<property name="qty" value="8" />
<property name="launch_date" value="26-11-1997" />
</bean>
</beans>
```

// UseProduct.java

```
package com.hcl.spring;
```

```
import
```

```
org.springframework.context.support.ClassPathXmlApplicat  
ionContext;
```

```
public class UseProduct {
```

```

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
        // creating context object for xml file
        ClassPathXmlApplicationContext context=new
ClassPathXmlApplicationContext("product.xml");

        // creating object for bean class

        Product pr1 =
context.getBean("prod1",Product.class); // No Casting

        Product pr2 =
context.getBean("prod2",Product.class); // No casting

        System.out.println("Using Setters Injection");
        // displaying objects by using setter injection
        System.out.println(pr1);
        System.out.println(pr2);

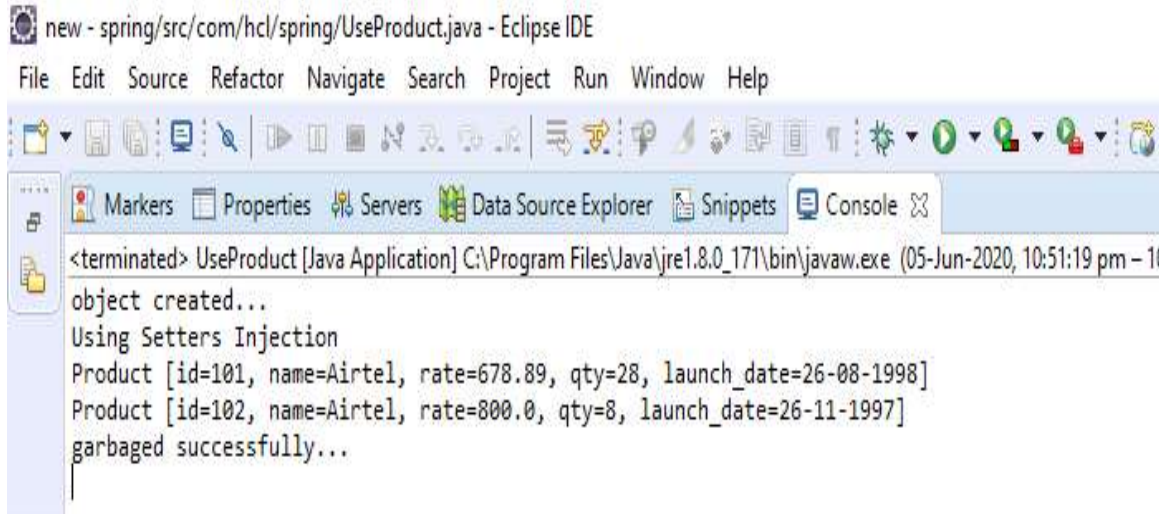
        // closing context object
        context.close();

    }

}

```

Output :



Program 2 : Create a Spring/Hibernate based MVC web application to persist data in a table. Table should have at-least 5 fields with varchar, numeric and date fields.

// Product.java (Model Class)

```
package com.hcl.model;  
  
import java.sql.Date;  
  
import javax.persistence.Column;  
  
import javax.persistence.Entity;  
  
import javax.persistence.Id;  
  
import javax.persistence.Table;  
  
@Entity  
@Table(name="product")
```

```

public class Product
{
    @Id
    @Column(name="pid")
    private int pid;
    @Column(name="pname")
    private String pname;
    @Column(name="price")
    private int price;
    @Column(name="qty")
    private String qty;
    @Column(name="launch_date")
    private Date launch_date;
    public Product() {
        super();
        // TODO Auto-generated constructor stub
    }
    // parameterized constructor
    public Product(int pid, String pname, int price,
String qty, Date launch_date) {
        super();
        this.pid = pid;

```



```

        this.pname = pname;

        this.price = price;

        this.qty = qty;

        this.launch_date = launch_date;
    }

    // getters and setters for attributes
    public int getPid() {
        return pid;
    }

    public void setPid(int pid) {
        this.pid = pid;
    }

    public String getPname() {
        return pname;
    }

    public void setPname(String pname) {
        this.pname = pname;
    }

    public int getPrice() {
        return price;
    }
}

```

```
    public void setPrice(int price) {  
        this.price = price;  
    }  
  
    public String getQty() {  
        return qty;  
    }  
  
    public void setQty(String qty) {  
        this.qty = qty;  
    }  
  
    public Date getLaunch_date() {  
        return launch_date;  
    }  
  
    public void setLaunch_date(Date launch_date) {  
        this.launch_date = launch_date;  
    }  
}
```

// IProductDAO.java (DAO Interface)

```
package com.hcl.dao;  
  
import org.hibernate.SessionFactory;  
  
import com.hcl.model.Product;
```

```

public interface IProductDAO
{
    // abstract methods of interface
    public void setSessionFactory(SessionFactory s);
    public void addProduct(Product p);
}

// ProductDAO.java (DAO Class)

package com.hcl.dao;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
import com.hcl.model.Product;

@Repository //Identifies a DAO class

public class ProductDAO implements IProductDAO
{
    @Autowired // Automatic initialization through
    dependency injection

    private SessionFactory sessionFactory;

    // implementation of two abstract methods of

```

IProductDAO interface

```
@Override
    public void setSessionFactory(SessionFactory s) {
        // TODO Auto-generated method stub
        this.sessionFactory=s;
    }

    @Override
    public void addProduct(Product p) {
        // TODO Auto-generated method stub
        Session
session=sessionFactory.getCurrentSession();
        session.save(p);
    }
}
```

// IProductService.java (Service Interface)

```
package com.hcl.service;
import com.hcl.model.Product;
public interface IProductService
{
```

```

        // abstract method of interface
        void addProduct(Product p);
    }

```

// ProductService.java (Service Class)

```

package com.hcl.service;

import javax.transaction.Transactional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.hcl.dao.ProductDAO;
import com.hcl.model.Product;

@Service //Identifies a service class
public class ProductService implements IProductService
{
    @Autowired
    private ProductDAO productDAO;

    @Override
    @Transactional // defines transaction
    public void addProduct(Product p) {
        // TODO Auto-generated method stub
        productDAO.addProduct(p);
    }
}

```

```
}  
  
}
```

// ProductController.java (Controller Class)

```
package com.hcl.controllers;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.dao.DataIntegrityViolationException;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.servlet.ModelAndView;  
import com.hcl.model.Product;  
import com.hcl.service.ProductService;  
  
@Controller  
  
public class ProductController  
{  
  
    @Autowired  
  
    private ProductService productservice;;  
  
    @RequestMapping(value="/add", method = RequestMethod.POST)
```

```

    public ModelAndView addProduct(@ModelAttribute("product")
Product product)
    {
        // creating object for ModelAndView
        ModelAndView mv = new ModelAndView();
        // try block
        try
        {
            productservice.addProduct(product);
            mv.setViewName("Success"); //Return success.jsp
page
        }
        // catch block
        catch(DataIntegrityViolationException e) {
            mv.addObject(product);
            mv.addObject("errmsg","ID already exists");
//creates error message if id already exists
            mv.setViewName("addProduct"); //Return data with
addProduct.jsp
        }
        return mv;
    }
}
// web.xml

```

```

<?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"
version="3.1">

    <display-name>SpringHibernate</display-name>

    <servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>
org.springframework.web.servlet.DispatcherServlet</servl
et-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

```

// dispatcher-servlet.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<beans

```



```
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schem
a/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

<context:component-scan base-package="com.hcl" />

<bean id="dataSource"
class="org.apache.commons.dbcp2.BasicDataSource">

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

<property name="url" value="jdbc:oracle:thin:
@localhost:1521/orcl2" />
```

```

<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="annotatedClasses">
<list>
<value>com.hcl.model.Product</value>
</list>
</property>
</bean>

<tx:annotation-driven transaction-
manager="hibernateTransactionManager"
proxy-target-class="true" />
<mvc:annotation-driven />

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

```

```

<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/WEB-INF/pages/">
</property>

<property name="suffix" value=".jsp"></property>

</bean>

</beans>

```

// addProduct.jsp

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

<body>

    <!-- form opening -->

    <form action="add" method="post" >

```

```

PID : <input type='text' name='pid'
value='${product.pid}' />

<span style="color:red"> ${errmsg} </span><br/>

PName : <input type='text' name='pname'
value='${product.pname}' /> <br/>

Price : <input type='text' name='price' value=
'${product.price}' /> <br/>

Quantity : <input type='text' name='qty'
value='${product.qty}' /> <br/>

Launch_date : <input type='date' name='launch_date'
value='${product.launch_date}' /> <br/>

<input type='submit' value='submit' />

</body>

</html>

```

// Success.jsp

```

<%@ page language="java" contentType="text/html;
charset=ISO-8859-1"

    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

```

```
<title>Insert title here</title>
</head>
<body bgcolor="orange">
    Product added sucessfully....
</body>
</html>
```

// Home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<jsp:include page="/WEB-INF/pages/addProduct.jsp" />
```

Output 1 :

ProductCont... web.xml Success.jsp Home.jsp Apache Tomc... Apache Tomc... Insert titl...

http://localhost:8081/ProductSpring/Home.jsp

PID :

PName :

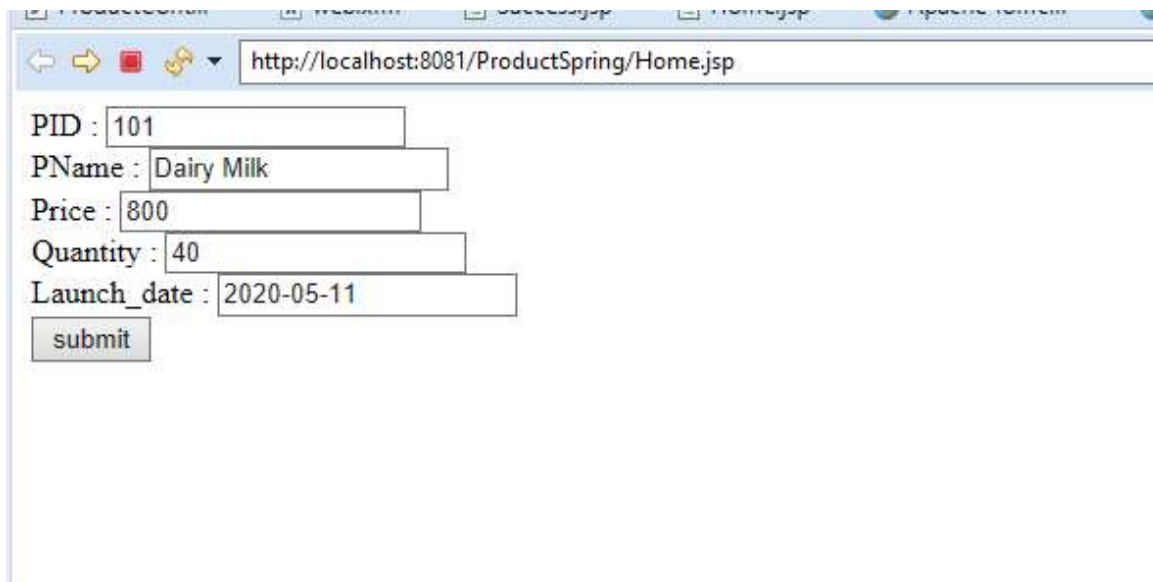
Price :

Quantity :

Launch_date : x

http://localhost:8081/ProductSpring/add

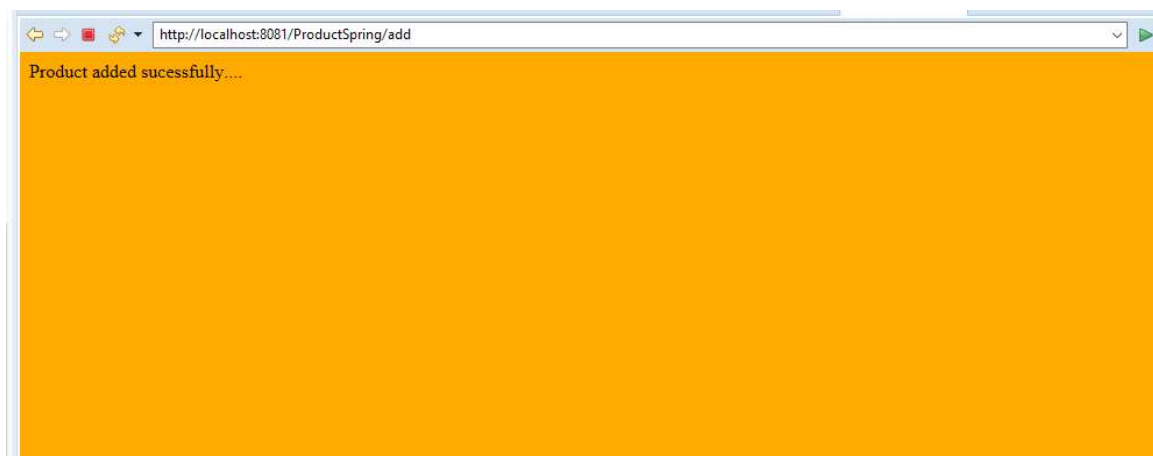
Product added sucessfully....



A screenshot of a web browser window displaying a form for adding a product. The address bar shows the URL `http://localhost:8081/ProductSpring/Home.jsp`. The form contains the following fields and values:

Field	Value
PID :	101
PName :	Dairy Milk
Price :	800
Quantity :	40
Launch_date :	2020-05-11

Below the fields is a button labeled "submit".



Output 2 : If We try to insert same id product error will exist .

http://localhost:8081/ProductSpring/add

PID : ID already exists

PName :

Price :

Quantity :

Launch_date :

SQL> select * from product;

PID	PNAME	PRICE	QTY	LAUNCH_DA
101	Dairy Milk	800	40	11-MAY-20
103	Apple	50000	1	26-AUG-98

SQL> |