NSBT
SPRING DAY 5 & 6 ASSIGHMENT

Spring MVC

1. Create the First Spring MVC application to print “Hello World” on the JSP page. Create “index.jsp” with hyperlink to the Controller and return a JSP Page as response which prints “Hello World”
   Make the application first with XML configuration and then with Java configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:context="http://www.springframework.org/schema/context"
       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.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <mvc:annotation-driven/>
<!--    <context:component-scan base-package="com.hellocontroller" />-->

    <bean id="/" class="com.hellocontroller.Hellomy"/>
    <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>

</beans>
```

Controller:

```
package com.hellocontroller;

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

@Controller
public class Hellomy {

    @RequestMapping("/hello")
    public String hello()
    {
        return "HelloPage";
    }
    @RequestMapping("/sum")
    public String sum()
    {
        System.out.println("This is about page");
        return "about";
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://Java.sun.com/xml/ns/javaee"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://Java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
         id="WebApp_ID" version="3.0">
  <display-name>SprMVCEg1</display-name>
  <servlet>
    <servlet-name>spring-dispatcher</servlet-name>
    <servlet-class>
      org.springframework.web.servlet.DispatcherServlet
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>spring-dispatcher</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>


<!--        spring-dispatcher-servlet.xml-->
```

```
<html>
<body>
<h1>First Spring MVC Application Demo</h1>

<h2>Hi Rohit</h2>

</body>
</html>
```

```
<html>
<body>
<h1>First Spring MVC Application Demo</h1>

<h2>Gate is not life, Gate can change the life</h2>

</body>
</html>
```

2. Change the above application by sending the name of the User as a Query String parameter to the link created in “index.jsp” page. Retrieve this name with the help of @RequestParam annotation and then send it to the display.jsp page for printing. The response must look like as mentioned below:

   Hello <username>

   Example:

   Index.jsp contains the below hyperlink:

   <a href=”sayHello.do?username=Ajay”> Say Hello </a>

   The output which need to be printed on display.jsp is:

   Hello Ajay

```
package com.hellocontroller;

import org.springframework.stereotype.Component;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
```

```java
@Controller
public class Hellomy {

    @RequestMapping("/hello")
    public String hello(@RequestParam("userName") String name, Model model)
    {
        model.addAttribute("name",name);
        return "HelloPage";
    }
    @RequestMapping("/latha")
    public String sum(Model model)
    {
        System.out.println("This is about page");
        model.addAttribute("name","Rohit Kr Singh");
        return "about";
    }
}
```

```jsp
<html>
<body>
<h1>First Spring MVC Application Demo</h1>

<h2>Hi Rohit</h2>
<%
String str=(String)request.getAttribute("name");
%>
<h1> Hello <%= str%></h1>

</body>
</html>
```

```jsp
<html>
<body>
<h1>First Spring MVC Application Demo</h1>
<%
Object obj=request.getAttribute("name");
%>
<h1> Hello <%= obj%> Gate is not life, Gate can change the life</h1>
</body>
</html>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:context="http://www.springframework.org/schema/context"
       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.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">
    <mvc:annotation-driven/>
<!--    <context:component-scan base-package="com.hellocontroller" />-->

    <bean id="/" class="com.hellocontroller.Hellomy"/>
    <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>

</beans>
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="welcome.html">welcome page</a><br>
<a href="hello?userName=Ajay">About</a><br>
<a href="latha">Hello Page</a><br>
<a href="Jai">Jai Shree Ram</a>
</body>
</html>
```

3. Change the above application to return the response with the help of ModelAndView object.

```
package com.hellocontroller;

import org.springframework.stereotype.Component;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class Hellomy {

    @RequestMapping("/hello")
    public String hello(@RequestParam("userName") String name, Model model)
    {
        model.addAttribute("name",name);
        return "HelloPage";
    }
    @RequestMapping("/latha")
    public String sum(Model model)
    {
        System.out.println("This is about page");
        model.addAttribute("name","Rohit Kr Singh");
        return "about";
    }

    @RequestMapping("/Jai")
    public ModelAndView modelAndView(){
        ModelAndView mod=new ModelAndView();
        //setting the data
        mod.addObject("name", "Rohit kr Singh");
        //setting the view name
        mod.setViewName("ShreeRam");
        return mod;
    }
}
```

```
<html>
<body>
<h1>This is my Page using model and View</h1>

<h1>Jai Shree Ram.......!!!!!</h1>
<%
String str=(String)request.getAttribute("name");
%>
<h1> Hello <%= str%></h1>

</body>
</html>
```

4. In the above application, check for the difference between forward and redirect. How will you do the same and what will be the impact. Maintain the difference in a readMe.txt file.

About.jsp:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Form Page</title>
</head>

<style>
.Ram{
        width:50%;
        border: 2px solid Black;
        margin:auto;
        Padding:2px;
        font-size:20px;
        }
        #RegistrationForm table tr td input{
        font-size:20px
        }
</style>
<body>
<div class="Ram">
    <h1>RegistrationForm</h1>
    <form id="MyForm" action="Form", method="post", modelAttribute="user2">
        <table>
            <tr>
                <td>Enter Your Name</td>
                <td><input type="text", name="user_Name",
placeholder="Enter Here"></td>
            </tr>
            <tr>
                <td>Enter Your E-mail</td>
                <td> <input type="email", name="user_Email",
placeholder="Enter Here"></td>
            </tr>
            <tr>
                <td>Enter Your Password</td>
                <td><input type="password", name="password",
placeholder="Enter Here"></td>
            </tr>
            <tr>
            <td>Enter Date</td>
            <td><input type="date", name="date"></td>
            </tr>
            <tr>
```

```
                <td>Select Gender</td>
                <td> <input type="radio", name="user_Gender",
value="Male">Male &nbsp; &nbsp; <input type="radio", name="User_Gender",
value="Female">Female</td>
            </tr>
            <tr>
                <td>Enter About Yourself</td>
                <td><input type="text", name="about", placeholder="Enter
about Yourself Here"></td>
            </tr>
            <tr>
                <td>
                    Select Your Course
                </td>
                <td>
                    <select name="user_course">
                        <option value="Java">Java</option>
                        <option value="php">php</option>
                        <option value="IoT">IoT</option>
                        <option value="C++">C++</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td>

                </td>
                <td>
                    <button type="submit", >Register</button>
                    <button type="Reset", >Reset</button>
                </td>
            </tr>
            <tr>
                <td style="text-align: Right">
                    <input type="checkbox", value="Checked",
name="Condition">
                </td>
                <td>
                    Agree terms and Conditions
                </td>
            </tr>
        </table>
    </form>
</div>
</body>
</html>
```

Output.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<h1>Registration Details are : </h1>
<hr>
<h1>Name of the user is: ${user.user_Name}  ${user1.user_Name}
${user2.user_Name}</h1>


<hr>

<h1>Email-ID of the user is: ${user.user_Email}</h1>

<hr>

<h1>Password selected by the user is: ${user.password}</h1>
```

```
<hr>

<h1>Gender of the user is : ${user.user_Gender}</h1>

<hr>

<h1>About the user is: ${user.about}</h1>

<hr>

<h1>Course opted by the user is :${user.user_course}</h1>

<hr>
</body>
</html>
```

User.java

```
package com.hellocontroller;

import org.springframework.stereotype.Component;

@Component
public class User {
   private String user_Name;
   private String user_Email;
   private String password;
   private String user_Gender;
   private String about;
   private String user_course;

    public String getUser_Name() {
        return user_Name;
    }

    public void setUser_Name(String user_Name) {
        this.user_Name = user_Name;
    }

    public String getUser_Email() {
        return user_Email;
    }

    public void setUser_Email(String user_Email) {
        this.user_Email = user_Email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getUser_Gender() {
        return user_Gender;
    }

    public void setUser_Gender(String user_Gender) {
        this.user_Gender = user_Gender;
    }

    public String getAbout() {
        return about;
    }
    public void setAbout(String about) {
        this.about = about;
    }

    public String getUser_course() {
        return user_course;
    }

    public void setUser_course(String user_course) {
        this.user_course = user_course;
    }

    @Override
    public String toString() {
        return "User{" +
                "user_Name='" + user_Name + '\'' +
                ", user_Email='" + user_Email + '\'' +
                ", password='" + password + '\'' +
                ", user_Gender='" + user_Gender + '\'' +
                ", about='" + about + '\'' +
                ", user_course='" + user_course + '\'' +
                '}';
    }
}
```

RegForm.Java (controller)

```
package com.hellocontroller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
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.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;

@Controller
@RequestMapping("Registration")
public class RegForm {
    @ModelAttribute
    public void commonDataForModal(Model model) {
        model.addAttribute("header", "Learn Code");
        model.addAttribute("Desc", "java is an OOP");
        System.out.println(model.getAttribute("header"));
        System.out.println(model.getAttribute("Desc"));
    }

    @RequestMapping(path = "/regd")
    public String ShowForm() {
        return "RegForm";

    }

    @RequestMapping(path="/Form",method = RequestMethod.POST)
    public String Form(@ModelAttribute User user, @ModelAttribute User
user2,Model model){

     // model.addAttribute("user1",user2);//--><--Not Needed it's done
automatically by the Spring container---->
        if(user.getPassword().isEmpty())
        {
```

```java
            System.out.println("redirecting");
            return "redirect:regd";
        }
        if(user.getUser_Email().isEmpty())
        {
            System.out.println("forwarding.......!!!!");
            return "forward:Jai";
        }
        System.out.println(user2);
        System.out.println();
        System.out.println(user);
        return "output";
    }
  @RequestMapping("/Jai")
    public ModelAndView modelAndView(){
        ModelAndView mod=new ModelAndView();
        //setting the data
        mod.addObject("name", "Rohit kr Singh");

        LocalDateTime loc=LocalDateTime.now();
        mod.addObject("time",loc);
        //setting the view name
        mod.setViewName("ShreeRam");
        List<String> lords= new ArrayList<>();
        lords.add("Lord Shiva");
        lords.add("Lord Ganesh");
        lords.add("Lord Kartik");
        lords.add("Lord Hanuman");
        lords.add("Lord Krishna");
        mod.addObject("marks", lords);
        return mod;
    }
}
```

5. In the above application, create another controller to access Request Headers with the help of @RequestHeader annotation. Display different response based on the value of 'referrer' request header. If header value is null, it should take the user to error page else to home page.

Controller class:

```java
package com.hellocontroller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RequestMapping;
@Controller
public class ResponseHeader {
    @RequestMapping("/headerresponse")
    public String headerresponse(
            @RequestHeader("host") String hostName,
            @RequestHeader ("Accept") String acceptType,
            @RequestHeader ("Accept-Language") String acceptLang,
            @RequestHeader ("Accept-Encoding") String acceptEnc,
            @RequestHeader ("Cookie") String cookie,
            @RequestHeader ("User-Agent") String userAgent,
            @RequestHeader ("referer") String ref)
    {

        System.out.println("Host : " + hostName);
        System.out.println("Accept : " + acceptType);
        System.out.println("Accept Language : " + acceptLang);
        System.out.println("Accept Encoding : " + acceptEnc);
        System.out.println("Cookie : " + cookie);
        System.out.println("User-Agent : " + userAgent);
        System.out.println("referer: " + ref);
        if(ref.isEmpty())
            return "Error";
        else
            return "index";
    }
}
```

Webapp->WEB-INF->views->indesx.jsp:

HomePage:

```
    <!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="welcome.html">welcome page</a><br>
<a href="first/hello?userName=Ajay">About</a><br>
<a href="first/latha">Hello Page</a><br>
<a href="first/Jai">Jai Shree Ram</a><br>
<a href="first/headerresponse">HeaderResponse</a><br>
<a href="GetPost/myMethod">GetPost</a><br>
<a href="Registration/regd">Registration Form</a><br>
<a href="emiPage">Emi Calculation Page</a>
</body>
</html>
```

6. In the same application, create a form to take the details of Customer to register the same. The details which need to be accepted are in the as per the class given in Spring Day –[1] Question – 1. Use POST method to submit the details. Create a Controller to accept the values using @RequestParam annotation, **Use the same service and repository layer created in Question-4 of Spring Day -3 assignment and display the inputs taken in the Success.jsp page.**

Success.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="m" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<h1>Registration Details are : </h1>
<hr>
<h1>Customer Id of the Customer is: ${custID}


<hr>

<h1>Name of the Customer is: ${custName}</h1>

<hr>

<h1>Monthly Income  of the Customer is: ${inc}</h1>
```

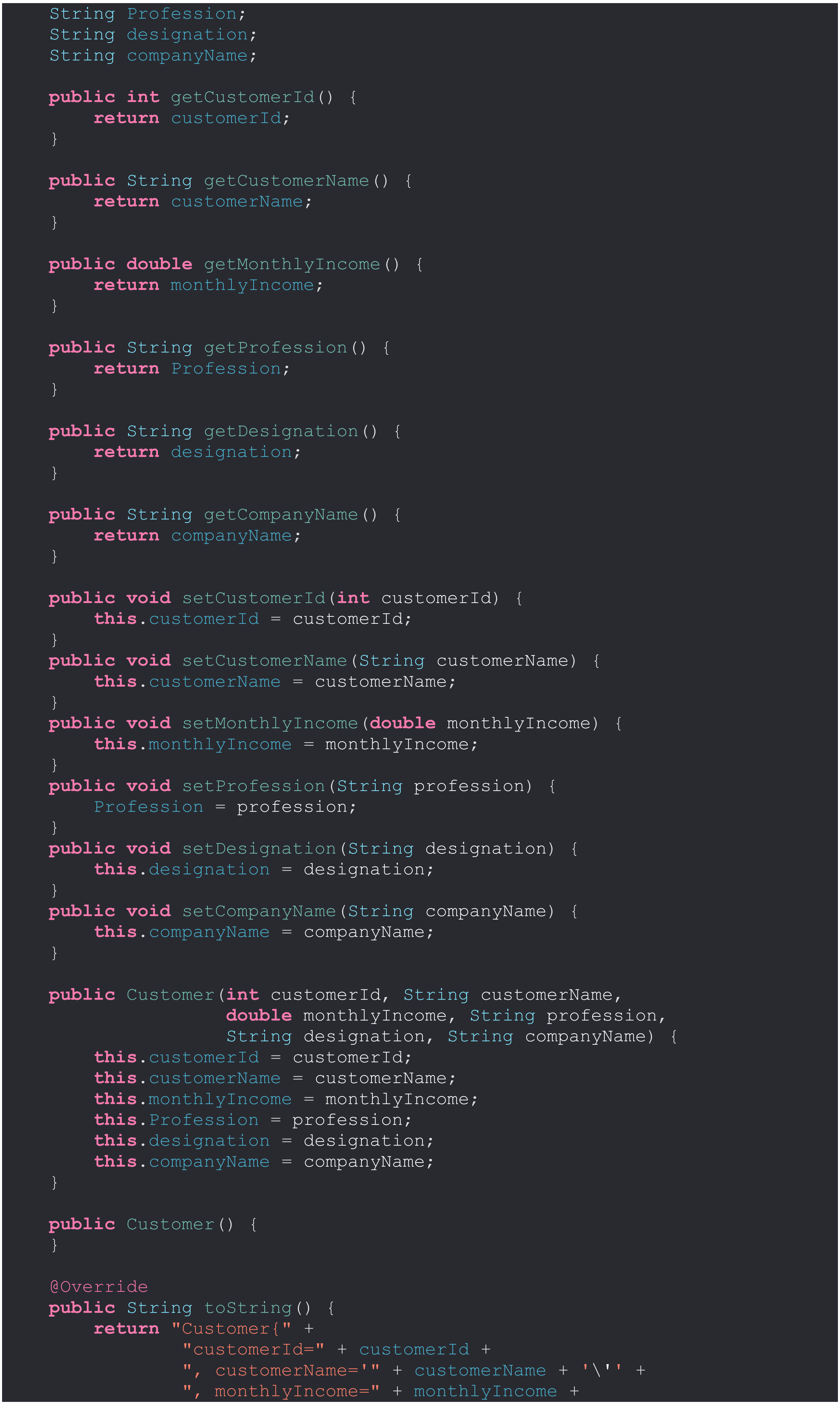
```
<hr>
<h1>Profession of the of the Customer is: ${pro}</h1>

<hr>
<h1>Designation of the Customer is: ${design}</h1>

 <hr>
 <h1>Company Name of the Customer is: ${compName}</h1>


</body>
</html>
```

Customer Registration From:

```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
 <html>
    <head>
        <title>User Registration Form</title>
        <style>
            .error {
                color: red;
            }
        </style>
    </head>
    <body>
        <h1>User Registration Form</h1>
        <form:form action="CustomerController" method=”post” >
            CUSTOMER ID (*) : <form:input path="customerId"/>

            <br>
            USER NAME (*) : <form:input path="customerName"/>

            <br>
            Monthly Income (*) : <form:input path="monthlyIncome"/>

            <br>
             PROFESSION (*) : <form:input path="Profession"/>

            <br>
            <br>
            DESIGNATION (*) : <form:input path="designation"/>

            <br>
            <br>
            COMPANY NAME (*) : <form:input path="companyName"/>

            <br>
            <input type="submit" value="SUBMIT"/>
            <input type="reset" value="RESET"/>
        </form:form>
    </body>
 </html>-
```

Customer.java

```
package com.hellocontroller.service;

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

@Component
public class Customer {
    int customerId;
    String customerName;
    double monthlyIncome;
```



```
    String Profession;
    String designation;
    String companyName;

    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getMonthlyIncome() {
        return monthlyIncome;
    }

    public String getProfession() {
        return Profession;
    }

    public String getDesignation() {
        return designation;
    }

    public String getCompanyName() {
        return companyName;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public void setMonthlyIncome(double monthlyIncome) {
        this.monthlyIncome = monthlyIncome;
    }
    public void setProfession(String profession) {
        Profession = profession;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    public Customer(int customerId, String customerName,
                    double monthlyIncome, String profession,
                    String designation, String companyName) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.monthlyIncome = monthlyIncome;
        this.Profession = profession;
        this.designation = designation;
        this.companyName = companyName;
    }

    public Customer() {
    }

    @Override
    public String toString() {
        return "Customer{" +
                "customerId=" + customerId +
                ", customerName='" + customerName + '\'' +
                ", monthlyIncome=" + monthlyIncome +
                ", Profession='" + Profession + '\'' +
                ", designation='" + designation + '\'' +
                ", companyName='" + companyName + '\'' +
                '}';
    }
}
```

MyAppConfig.java

```
package com.hellocontroller.service;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.EnableAspectJAutoProxy;

@Configuration
@ComponentScan
public class MyAppConfig {
}
```

CustomerController.java

```
package com.hellocontroller.service;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

@Controller
public class CustomerController {

    @RequestMapping("/FormQ8")
    public String customerForm(Model model){
        model.addAttribute("customer", new Customer());
        return "custForm";
    }

    @RequestMapping(path="/CustomerController" method=Request)
    public String myCustomer(@RequestParam("customerId") int custID,
                             @RequestParam("customerName") String custName,
                             @RequestParam("monthlyIncome") double inc,
                             @RequestParam("Profession") String pro,
                             @RequestParam("designation") String design,
                             @RequestParam("companyName") String compName,
Model model) throws SQLException {
        //Customer cust=new
Customer(custID,custName,inc,pro,design,compName);
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        Customer cust = (Customer) context.getBean(Customer.class);
        cust.setCustomerId(custID);
        cust.setCustomerName(custName);
        cust.setMonthlyIncome(inc);
        cust.setProfession(pro);
        cust.setDesignation(design);
        cust.setCompanyName(compName);
        customerdao.insertCustomer(cust);
        System.out.println(cust);
        model.addAttribute("custID",cust.getCustomerId());
        model.addAttribute("custName",cust.getCustomerName());
        model.addAttribute("inc",cust.getMonthlyIncome());
        model.addAttribute("pro",cust.getProfession());
        model.addAttribute("design",cust.getDesignation());
        model.addAttribute("compName",cust.getCompanyName());
        return "Success";
    }
}
```

CustomerDAO.java

```
package com.hellocontroller.service;
import org.springframework.stereotype.Repository;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import static com.hellocontroller.service.GetConnection.getConnection;
@Repository
public class CustomerDAO {
    public boolean insertCustomer(Customer customer) throws SQLException {
        System.out.println("inside insert customer");
        getConnection();
        System.out.println("inside DAO" + customer);

        String sql = "insert into Customer_16908 values(?,?,?,?,?,?)";
        try {
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setInt(1, customer.getCustomerId());
            pst.setString(2, customer.getCustomerName());
            pst.setDouble(3, customer.getMonthlyIncome());
            pst.setString(4, customer.getProfession());
            pst.setString(5, customer.getDesignation());
            pst.setString(6, customer.getCompanyName());

            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean deleteCustomer(String customerId) throws SQLException {
        getConnection();
        try {
            String sql = "delete from Customer_16908 where CUSTOMER_ID=?";
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setString(1, customerId);
            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean updateCustomer(int customerId, String custName,
                                  double minc, String profession,
                                  String designation, String companyName) {
        getConnection();
        try {
            String sql = "update from Customer_16908 where customerID=?";
            PreparedStatement pst = getConnection().prepareStatement(sql);
            pst.setInt(1, customerId);
            pst.setString(2, custName);
            pst.setDouble(3, minc);
            pst.setString(4, profession);
            pst.setString(5, designation);
            pst.setString(6, companyName);
            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
            return false;
        }
    }
    public List<Customer> getCustomersByName(String customerName) throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
getConnection().createStatement().executeQuery("select * from
CUSTOMER_M_16908 where FIRST_NAME='" + customerName + "'");
                            cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
    public List<Customer> getAllCustomers(Customer Customer) throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
                    getConnection().createStatement().executeQuery("select
* from Customer_16908");
                            cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
}
```

GetConnection.java

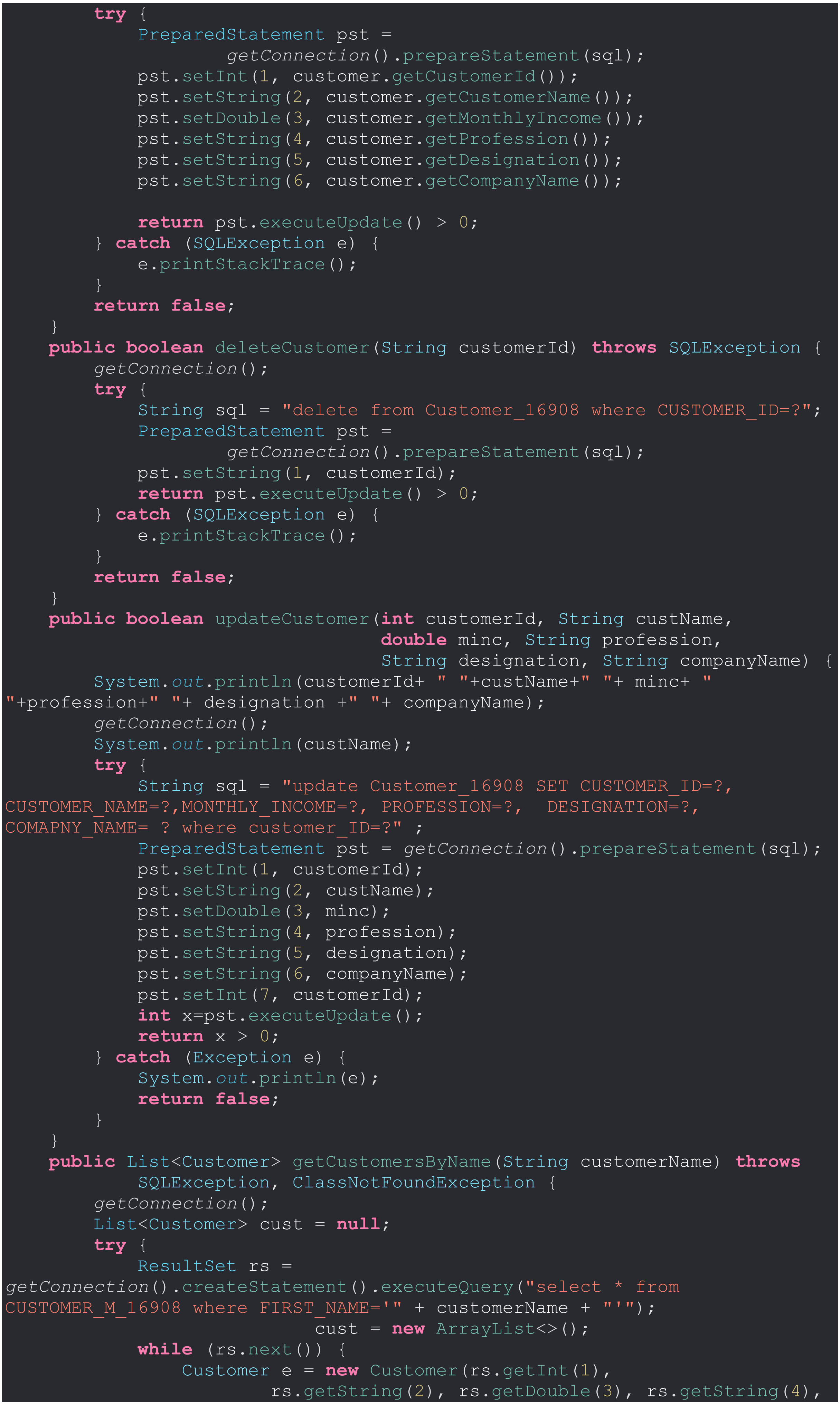
```
package com.hellocontroller.service;
;
import org.springframework.stereotype.Component;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

@Component("Connection")
public class GetConnection {
    public static Connection connection;
    public static Connection getConnection() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            connection =
DriverManager.getConnection("jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c",
"training", "training");
            return connection;
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

7. In the same form create two Submit buttons with the two values – Add and Update. When user clicks on Add Button – A new customer is registered. When user clicks on Update button – the customer details with the given customer id are updated.

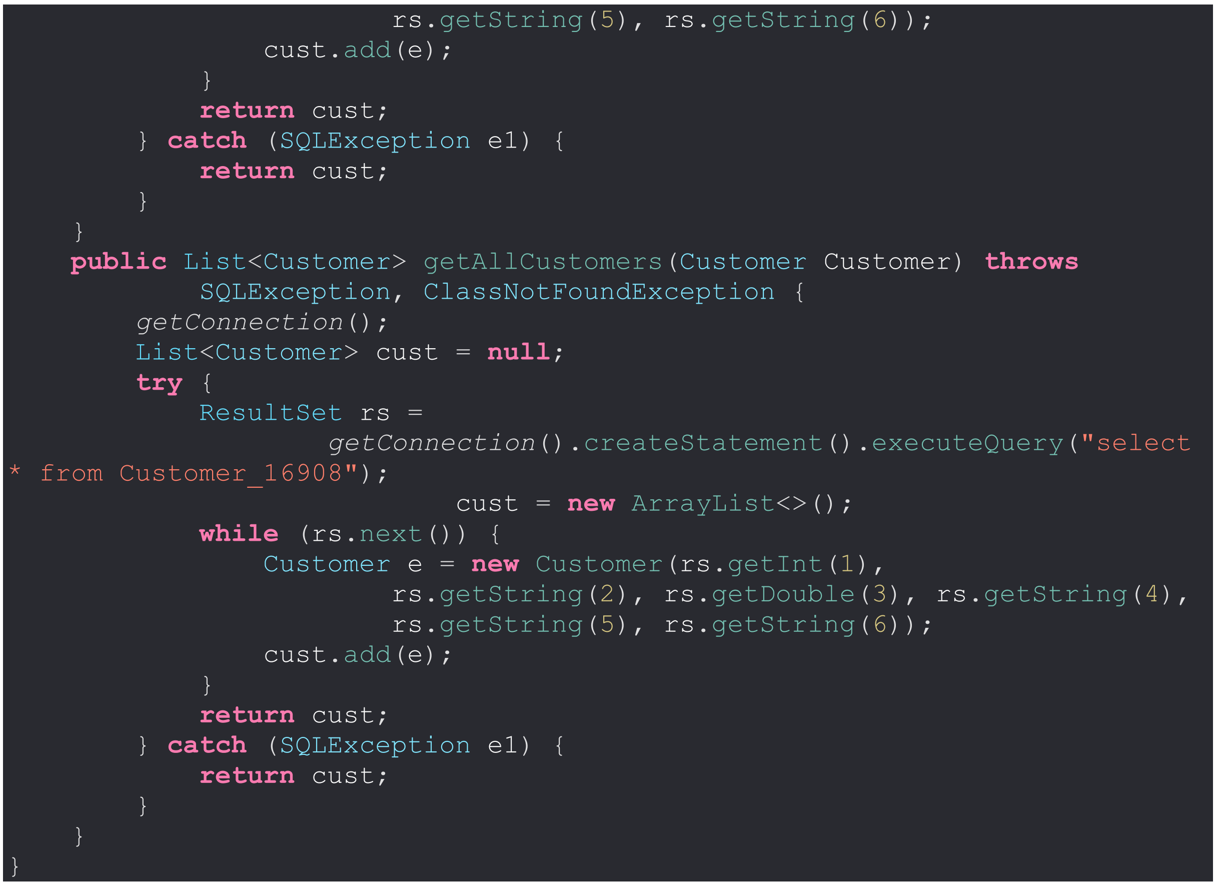
```
package com.hellocontroller.service;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.annotation.AnnotationConfigApplicationCon
text;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
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.bind.annotation.RequestParam;
import java.sql.SQLException;

@Controller
public class CustomerController {

    @RequestMapping("/FormQ8")
    public String customerForm(Model model){
        model.addAttribute("customer", new Customer());
        return "custForm";
    }

    @RequestMapping("/AddCustomerController")
    public String addCustomer(@RequestParam("customerId") int custID,
                              @RequestParam("customerName") String
custName,
                              @RequestParam("monthlyIncome") double
inc,
                              @RequestParam("Profession") String pro,
                              @RequestParam("designation") String
design,
                              @RequestParam("companyName") String
compName, Model model) throws SQLException {
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        Customer cust = (Customer) context.getBean(Customer.class);
        cust.setCustomerId(custID);
        cust.setCustomerName(custName);
        cust.setMonthlyIncome(inc);
        cust.setProfession(pro);
        cust.setDesignation(design);
        cust.setCompanyName(compName);
        System.out.println(cust);
        customerdao.insertCustomer(cust);
        model.addAttribute("custID",cust.getCustomerId());
        model.addAttribute("custName",cust.getCustomerName());
        model.addAttribute("inc",cust.getMonthlyIncome());
        model.addAttribute("pro",cust.getProfession());
        model.addAttribute("design",cust.getDesignation());
        model.addAttribute("compName",cust.getCompanyName());
        return "Success";
    }
    @RequestMapping(path="/UpdateCustomerController",method=
RequestMethod.POST)
    public String updateCustomer(@RequestParam("customerId") int
custID,
                             @RequestParam("customerName") String
custName,
                             @RequestParam("monthlyIncome") double
inc,
                             @RequestParam("Profession") String pro,
                             @RequestParam("designation") String
design,
                             @RequestParam("companyName") String
compName, Model model) throws SQLException {
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        Customer cust = (Customer) context.getBean(Customer.class);

customerdao.updateCustomer(custID,custName,inc,pro,design,compName);
        model.addAttribute("custID",custID);
        model.addAttribute("custName",custName);
        model.addAttribute("inc",inc);
        model.addAttribute("pro",pro);
        model.addAttribute("design",design);
        model.addAttribute("compName",compName);
        return "Success";
    }
```

```
package com.hellocontroller.service;

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

@Component
public class Customer {
    int customerId;
    String customerName;
    double monthlyIncome;
    String Profession;
    String designation;
    String companyName;

    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getMonthlyIncome() {
        return monthlyIncome;
    }

    public String getProfession() {
        return Profession;
    }

    public String getDesignation() {
        return designation;
    }

    public String getCompanyName() {
        return companyName;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public void setMonthlyIncome(double monthlyIncome) {
        this.monthlyIncome = monthlyIncome;
    }
    public void setProfession(String profession) {
        Profession = profession;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    public Customer(int customerId, String customerName,
                    double monthlyIncome, String profession,
                    String designation, String companyName) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.monthlyIncome = monthlyIncome;
        this.Profession = profession;
        this.designation = designation;
        this.companyName = companyName;
    }

    public Customer() {
    }

    @Override
    public String toString() {
        return "Customer{" +
                "customerId=" + customerId +
                ", customerName='" + customerName + '\'' +
                ", monthlyIncome=" + monthlyIncome +
                ", Profession='" + Profession + '\'' +
                ", designation='" + designation + '\'' +
                ", companyName='" + companyName + '\'' +
                '}';
    }
}
```

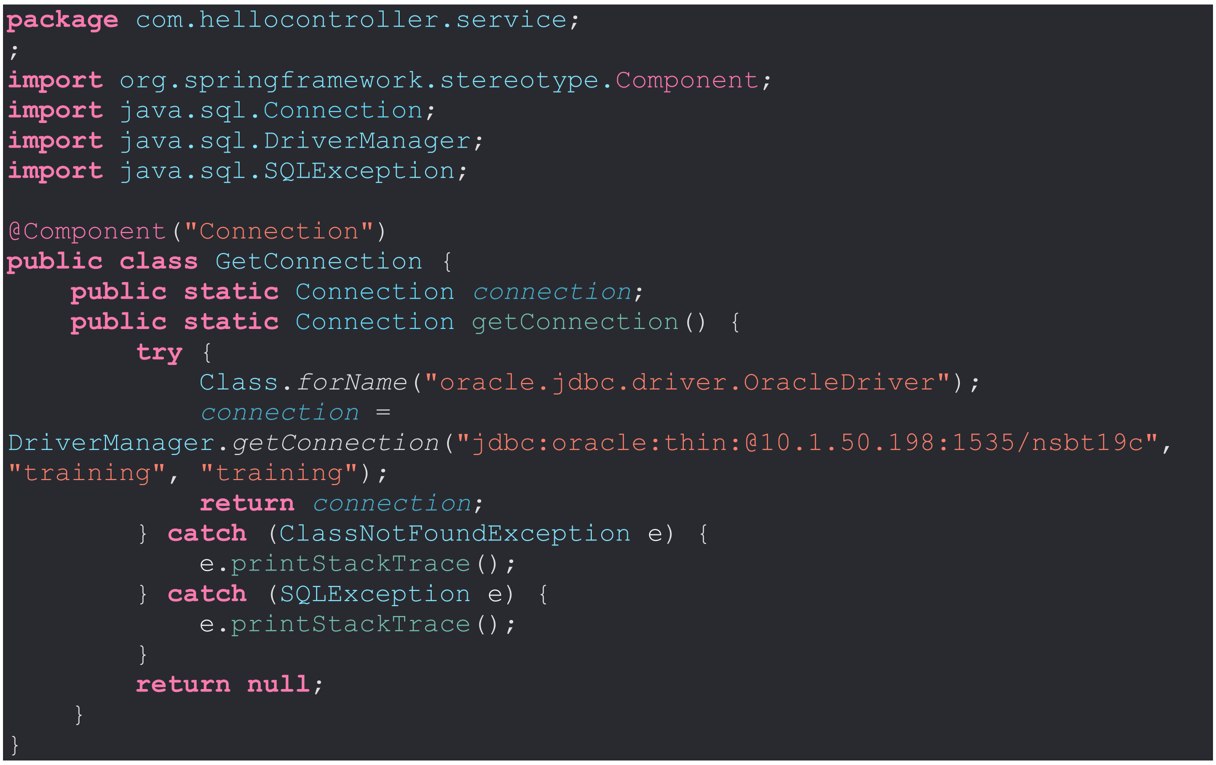
```
package com.hellocontroller.service;
import org.springframework.stereotype.Repository;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import static com.hellocontroller.service.GetConnection.getConnection;
@Repository
public class CustomerDAO {
    public boolean insertCustomer(Customer customer) throws SQLException {
        getConnection();
        String sql = "insert into Customer_16908 values(?,?,?,?,?,?)";
```



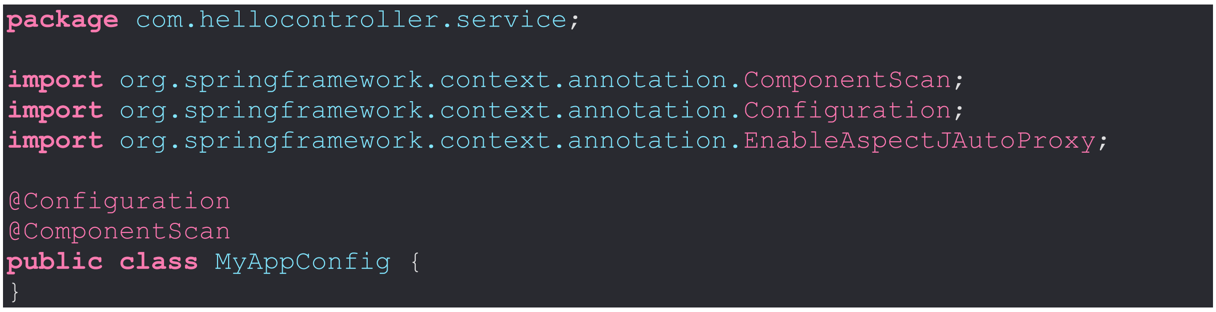
```
        try {
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setInt(1, customer.getCustomerId());
            pst.setString(2, customer.getCustomerName());
            pst.setDouble(3, customer.getMonthlyIncome());
            pst.setString(4, customer.getProfession());
            pst.setString(5, customer.getDesignation());
            pst.setString(6, customer.getCompanyName());

            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean deleteCustomer(String customerId) throws SQLException {
        getConnection();
        try {
            String sql = "delete from Customer_16908 where CUSTOMER_ID=?";
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setString(1, customerId);
            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean updateCustomer(int customerId, String custName,
                                  double minc, String profession,
                                  String designation, String companyName) {
        System.out.println(customerId+ " "+custName+" "+ minc+ "
"+profession+" "+ designation +" "+ companyName);
        getConnection();
        System.out.println(custName);
        try {
            String sql = "update Customer_16908 SET CUSTOMER_ID=?,
CUSTOMER_NAME=?,MONTHLY_INCOME=?, PROFESSION=?,  DESIGNATION=?,
COMAPNY_NAME= ? where customer_ID=?" ;
            PreparedStatement pst = getConnection().prepareStatement(sql);
            pst.setInt(1, customerId);
            pst.setString(2, custName);
            pst.setDouble(3, minc);
            pst.setString(4, profession);
            pst.setString(5, designation);
            pst.setString(6, companyName);
            pst.setInt(7, customerId);
            int x=pst.executeUpdate();
            return x > 0;
        } catch (Exception e) {
            System.out.println(e);
            return false;
        }
    }
    public List<Customer> getCustomersByName(String customerName) throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
getConnection().createStatement().executeQuery("select * from
CUSTOMER_M_16908 where FIRST_NAME='" + customerName + "'");
                            cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
```



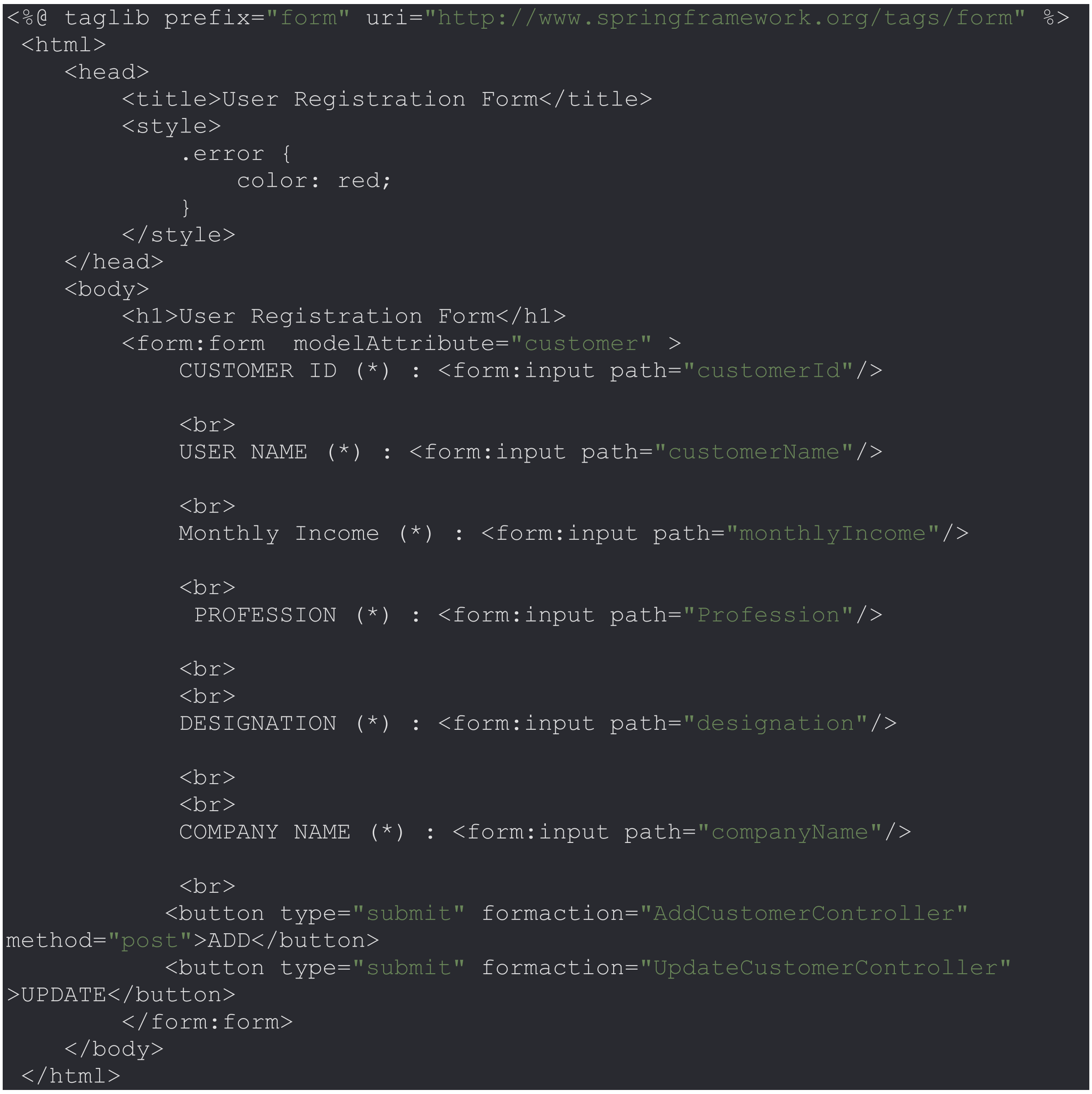
```
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
    public List<Customer> getAllCustomers(Customer Customer)  throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
                    getConnection().createStatement().executeQuery("select
* from Customer_16908");
                            cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
}
```



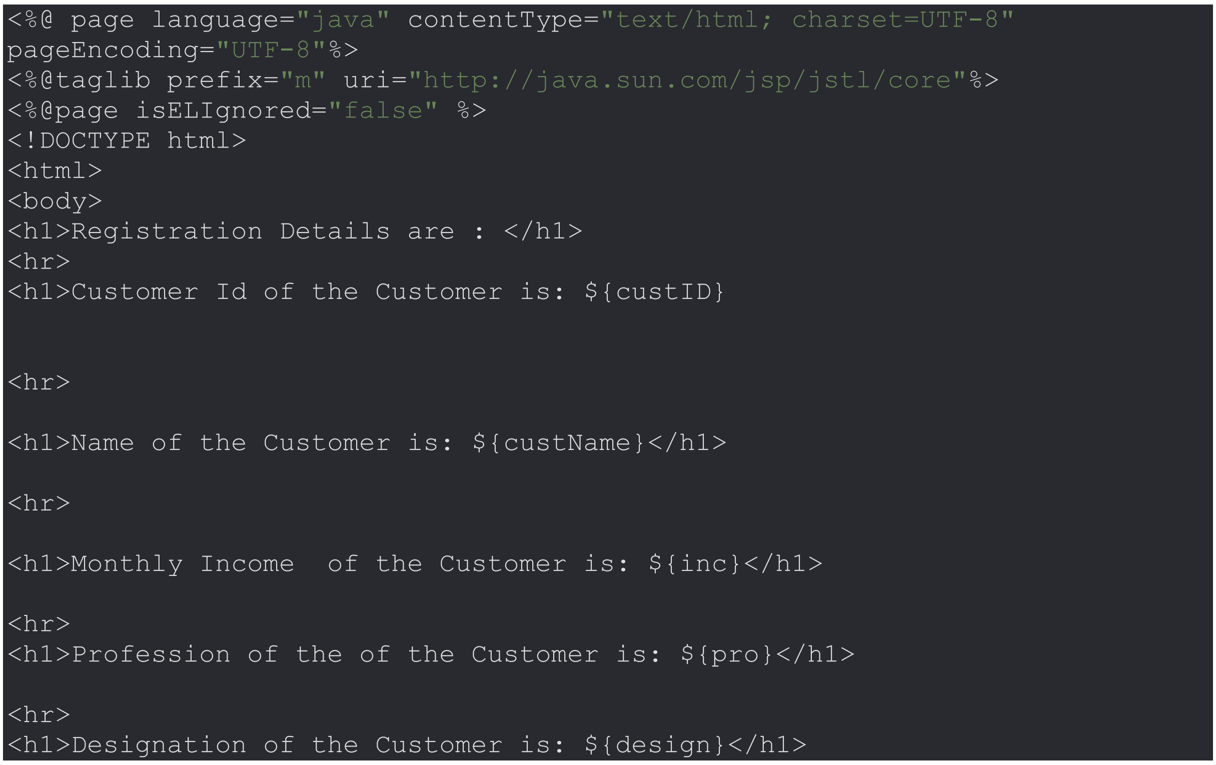
```
package com.hellocontroller.service;
;
import org.springframework.stereotype.Component;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

@Component("Connection")
public class GetConnection {
    public static Connection connection;
    public static Connection getConnection() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            connection =
DriverManager.getConnection("jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c",
"training", "training");
            return connection;
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```



```
package com.hellocontroller.service;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.EnableAspectJAutoProxy;

@Configuration
@ComponentScan
public class MyAppConfig {
}
```



```
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
 <html>
    <head>
        <title>User Registration Form</title>
        <style>
            .error {
                color: red;
            }
        </style>
    </head>
    <body>
        <h1>User Registration Form</h1>
        <form:form  modelAttribute="customer" >
            CUSTOMER ID (*) : <form:input path="customerId"/>

            <br>
            USER NAME (*) : <form:input path="customerName"/>

            <br>
            Monthly Income (*) : <form:input path="monthlyIncome"/>

            <br>
             PROFESSION (*) : <form:input path="Profession"/>

            <br>
            <br>
            DESIGNATION (*) : <form:input path="designation"/>

            <br>
            <br>
            COMPANY NAME (*) : <form:input path="companyName"/>

            <br>
           <button type="submit" formaction="AddCustomerController"
method="post">ADD</button>
           <button type="submit" formaction="UpdateCustomerController"
>UPDATE</button>
        </form:form>
    </body>
 </html>
```



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="m" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<h1>Registration Details are : </h1>
<hr>
<h1>Customer Id of the Customer is: ${custID}


<hr>

<h1>Name of the Customer is: ${custName}</h1>

<hr>

<h1>Monthly Income  of the Customer is: ${inc}</h1>

<hr>
<h1>Profession of the of the Customer is: ${pro}</h1>

<hr>
<h1>Designation of the Customer is: ${design}</h1>
```

```
 <hr>
 <h1>Company Name of the Customer is: ${compName}</h1>


</body>
</html>
```

8. In the same application as above, create a controller to fetch the names of all Customers. Display the list of Customer Names with each name acting as a hyperlink. When one hyperlink is clicked, the complete details of that customer are displayed. Demonstrate the use of @PathVariable annotation to create the application.

GetCustomerByNamePage.jsp

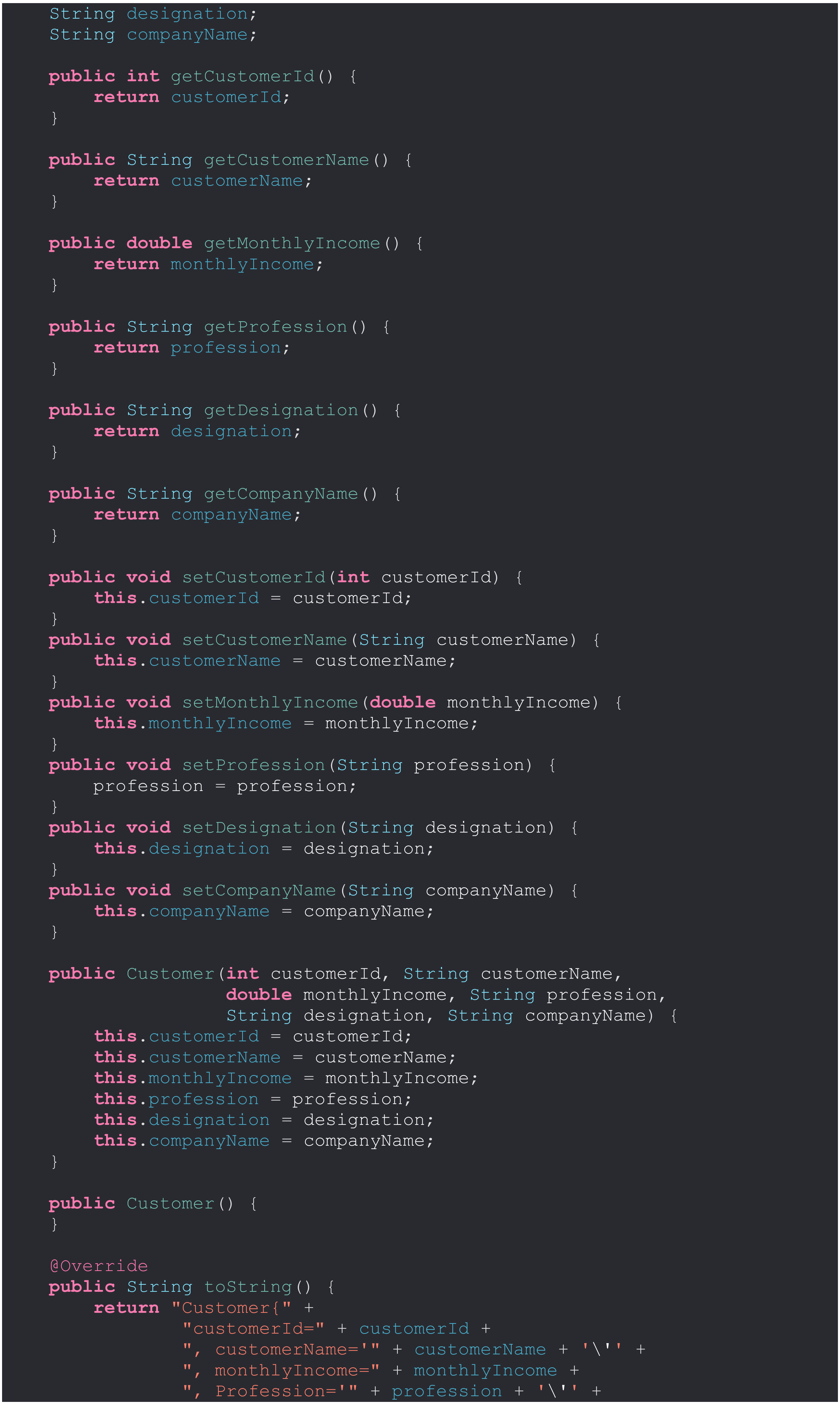
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="m" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<hr>
<h1> Detail of the Customer are: </h1><hr>
<h1>Name of the Customer is: ${cust.get(0).customerName}</h1><hr>
<h1>Customer Id of the Customer is: ${cust.get(0).customerId}</h1><hr>
<h1>Monthly Income of the Customer is:
${cust.get(0).monthlyIncome}</h1><hr>
<h1>Profession of the Customer is: ${cust.get(0).profession}</h1><hr>
<h1>Designation of the Customer is: ${cust.get(0).designation}</h1><hr>
<h1>Company Name of the Customer is: ${cust.get(0).companyName}</h1><hr>
</body>
</html>
```

GetCustomerPage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="m" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<hr>
<h1>Click on the name of the Customer to fetch all its details: </h1>
<m:forEach var="item" items="${cust}">
<h3><a
href="getCustomerByName/${item.customerName}">${item.customerName}</h3>
</m:forEach>

</body>
</html>
```

Customer.java

```
package com.hellocontroller.service;

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

@Component
public class Customer {
    int customerId;
    String customerName;
    double monthlyIncome;
    String profession;
```



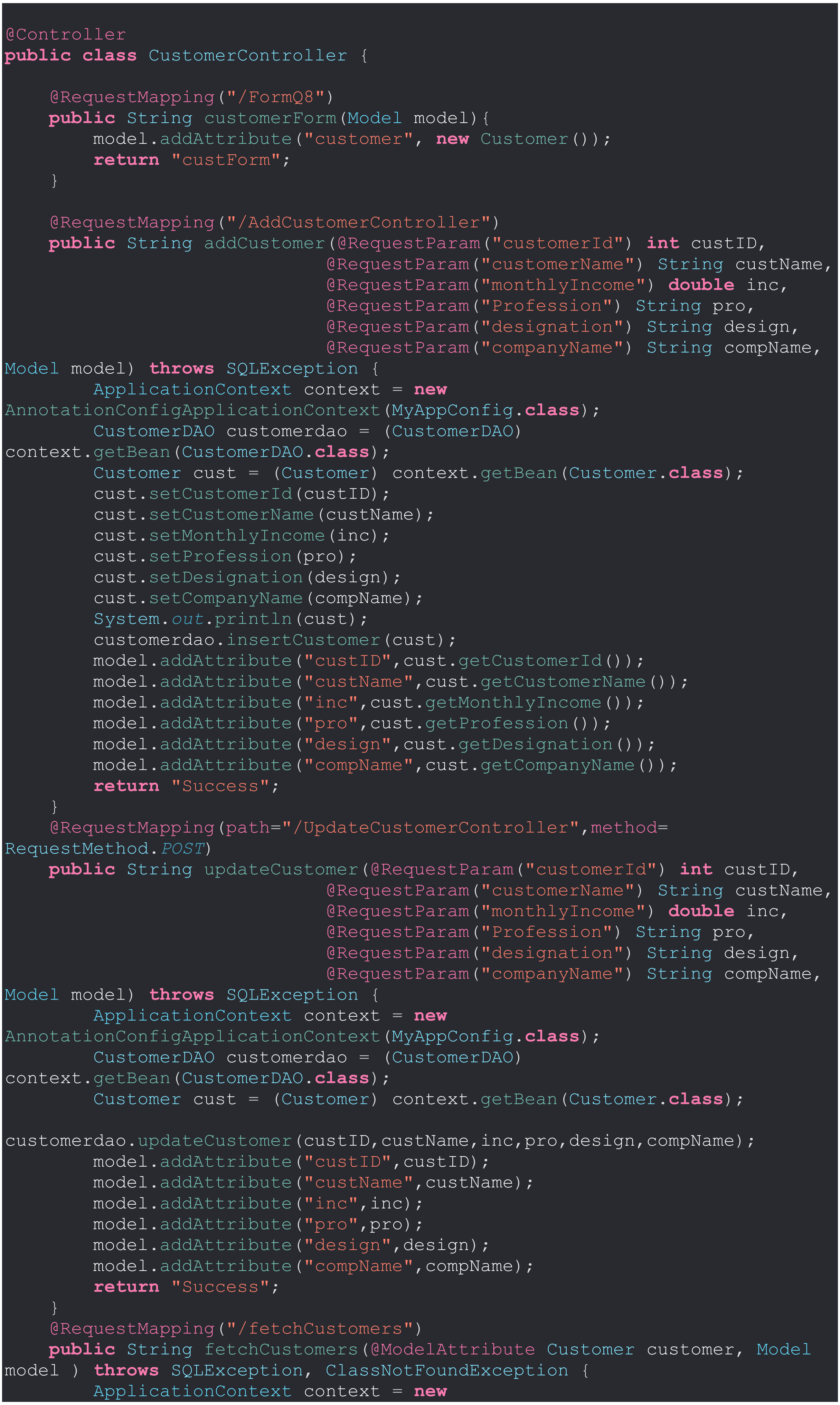
```java
    String designation;
    String companyName;

    public int getCustomerId() {
        return customerId;
    }

    public String getCustomerName() {
        return customerName;
    }

    public double getMonthlyIncome() {
        return monthlyIncome;
    }

    public String getProfession() {
        return profession;
    }

    public String getDesignation() {
        return designation;
    }

    public String getCompanyName() {
        return companyName;
    }

    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }
    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }
    public void setMonthlyIncome(double monthlyIncome) {
        this.monthlyIncome = monthlyIncome;
    }
    public void setProfession(String profession) {
        profession = profession;
    }
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    public void setCompanyName(String companyName) {
        this.companyName = companyName;
    }

    public Customer(int customerId, String customerName,
                    double monthlyIncome, String profession,
                    String designation, String companyName) {
        this.customerId = customerId;
        this.customerName = customerName;
        this.monthlyIncome = monthlyIncome;
        this.profession = profession;
        this.designation = designation;
        this.companyName = companyName;
    }

    public Customer() {
    }

    @Override
    public String toString() {
        return "Customer{" +
                "customerId=" + customerId +
                ", customerName='" + customerName + '\'' +
                ", monthlyIncome=" + monthlyIncome +
                ", Profession='" + profession + '\'' +
                ", designation='" + designation + '\'' +
                ", companyName='" + companyName + '\'' +
                '}';
    }
}
```

CustomerDAO.java

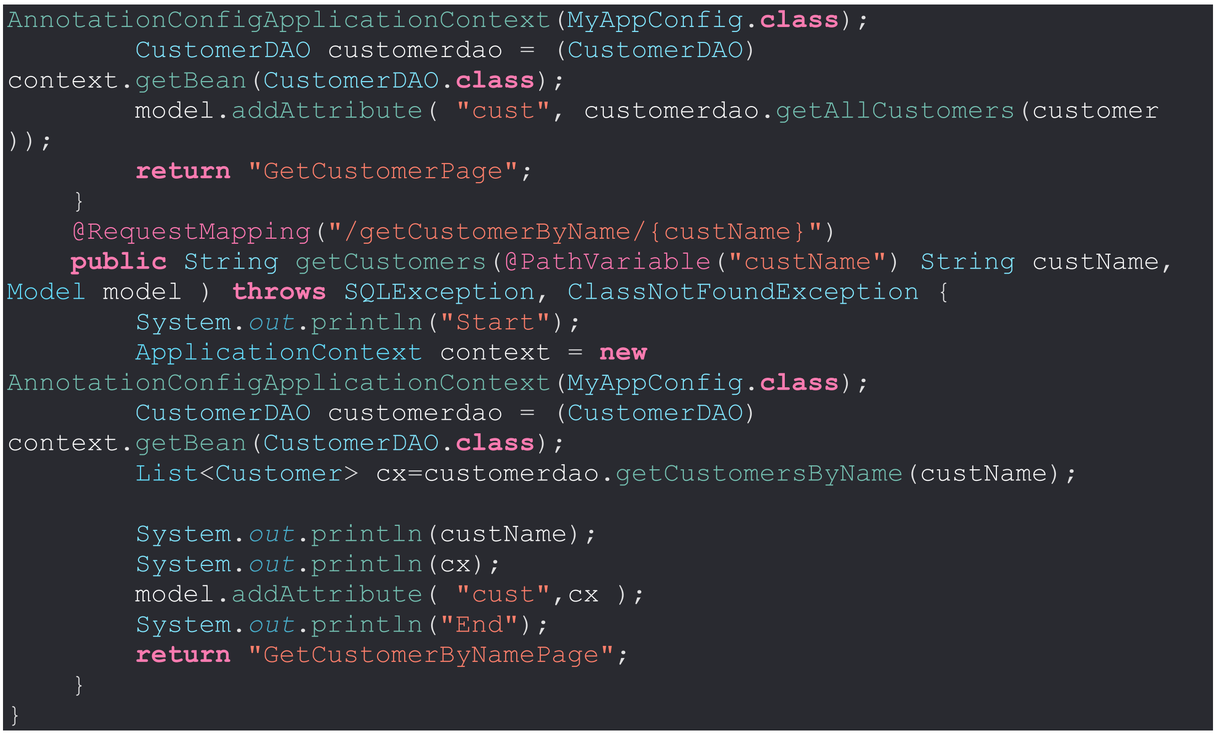
```
package com.hellocontroller.service;
import org.springframework.stereotype.Repository;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import static com.hellocontroller.service.GetConnection.getConnection;
@Repository
public class CustomerDAO {
    public boolean insertCustomer(Customer customer) throws SQLException {
        getConnection();
        String sql = "insert into Customer_16908 values(?,?,?,?,?,?)";
        try {
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setInt(1, customer.getCustomerId());
            pst.setString(2, customer.getCustomerName());
            pst.setDouble(3, customer.getMonthlyIncome());
            pst.setString(4, customer.getProfession());
            pst.setString(5, customer.getDesignation());
            pst.setString(6, customer.getCompanyName());

            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean deleteCustomer(String customerId) throws SQLException {
        getConnection();
        try {
            String sql = "delete from Customer_16908 where CUSTOMER_ID=?";
            PreparedStatement pst =
                    getConnection().prepareStatement(sql);
            pst.setString(1, customerId);
            return pst.executeUpdate() > 0;
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return false;
    }
    public boolean updateCustomer(int customerId, String custName,
                                  double minc, String profession,
                                  String designation, String companyName) {
        System.out.println(customerId+ " "+custName+" "+ minc+ "
"+profession+" "+ designation +" "+ companyName);
        getConnection();
        System.out.println(custName);
        try {
            String sql = "update Customer_16908 SET CUSTOMER_ID=?,
CUSTOMER_NAME=?,MONTHLY_INCOME=?, PROFESSION=?,  DESIGNATION=?,
COMAPNY_NAME= ? where customer_ID=?" ;
            PreparedStatement pst = getConnection().prepareStatement(sql);
            pst.setInt(1, customerId);
            pst.setString(2, custName);
            pst.setDouble(3, minc);
            pst.setString(4, profession);
            pst.setString(5, designation);
            pst.setString(6, companyName);
            pst.setInt(7, customerId);
            int x=pst.executeUpdate();
            return x > 0;
        } catch (Exception e) {
            System.out.println(e);
            return false;
        }
    }
    public List<Customer> getCustomersByName(String customerName) throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
getConnection().createStatement().executeQuery("select * from
Customer_16908 where CUSTOMER_NAME='" + customerName + "'");
                    cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
    public List<Customer> getAllCustomers(Customer Customer) throws
            SQLException, ClassNotFoundException {
        getConnection();
        List<Customer> cust = null;
        try {
            ResultSet rs =
                    getConnection().createStatement().executeQuery("select
* from Customer_16908");
                    cust = new ArrayList<>();
            while (rs.next()) {
                Customer e = new Customer(rs.getInt(1),
                        rs.getString(2), rs.getDouble(3), rs.getString(4),
                        rs.getString(5), rs.getString(6));
                cust.add(e);
            }
            return cust;
        } catch (SQLException e1) {
            return cust;
        }
    }
}
```

CustomerController.java

```
package com.hellocontroller.service;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import javax.xml.stream.events.StartDocument;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```



```
@Controller
public class CustomerController {

    @RequestMapping("/FormQ8")
    public String customerForm(Model model){
        model.addAttribute("customer", new Customer());
        return "custForm";
    }

    @RequestMapping("/AddCustomerController")
    public String addCustomer(@RequestParam("customerId") int custID,
                             @RequestParam("customerName") String custName,
                             @RequestParam("monthlyIncome") double inc,
                             @RequestParam("Profession") String pro,
                             @RequestParam("designation") String design,
                             @RequestParam("companyName") String compName,
Model model) throws SQLException {
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        Customer cust = (Customer) context.getBean(Customer.class);
        cust.setCustomerId(custID);
        cust.setCustomerName(custName);
        cust.setMonthlyIncome(inc);
        cust.setProfession(pro);
        cust.setDesignation(design);
        cust.setCompanyName(compName);
        System.out.println(cust);
        customerdao.insertCustomer(cust);
        model.addAttribute("custID",cust.getCustomerId());
        model.addAttribute("custName",cust.getCustomerName());
        model.addAttribute("inc",cust.getMonthlyIncome());
        model.addAttribute("pro",cust.getProfession());
        model.addAttribute("design",cust.getDesignation());
        model.addAttribute("compName",cust.getCompanyName());
        return "Success";
    }
    @RequestMapping(path="/UpdateCustomerController",method=
RequestMethod.POST)
    public String updateCustomer(@RequestParam("customerId") int custID,
                             @RequestParam("customerName") String custName,
                             @RequestParam("monthlyIncome") double inc,
                             @RequestParam("Profession") String pro,
                             @RequestParam("designation") String design,
                             @RequestParam("companyName") String compName,
Model model) throws SQLException {
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        Customer cust = (Customer) context.getBean(Customer.class);

customerdao.updateCustomer(custID,custName,inc,pro,design,compName);
        model.addAttribute("custID",custID);
        model.addAttribute("custName",custName);
        model.addAttribute("inc",inc);
        model.addAttribute("pro",pro);
        model.addAttribute("design",design);
        model.addAttribute("compName",compName);
        return "Success";
    }
    @RequestMapping("/fetchCustomers")
    public String fetchCustomers(@ModelAttribute Customer customer, Model
model ) throws SQLException, ClassNotFoundException {
        ApplicationContext context = new
```

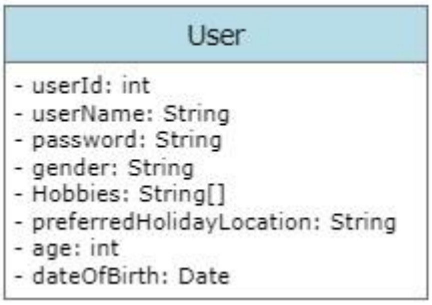


```
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        model.addAttribute( "cust", customerdao.getAllCustomers(customer
));
        return "GetCustomerPage";
    }
    @RequestMapping("/getCustomerByName/{custName}")
    public String getCustomers(@PathVariable("custName") String custName,
Model model ) throws SQLException, ClassNotFoundException {
        System.out.println("Start");
        ApplicationContext context = new
AnnotationConfigApplicationContext(MyAppConfig.class);
        CustomerDAO customerdao = (CustomerDAO)
context.getBean(CustomerDAO.class);
        List<Customer> cx=customerdao.getCustomersByName(custName);

        System.out.println(custName);
        System.out.println(cx);
        model.addAttribute( "cust",cx );
        System.out.println("End");
        return "GetCustomerByNamePage";
    }
}
```

NSBT

SPRING DAY 5 & 6 ASSIGHMENT

9. Create an application to create a form for User Registration. The User Bean is as per the class diagram:



User

- userId: int
- userName: String
- password: String
- gender: String
- Hobbies: String[]
- preferredHolidayLocation: String
- age: int
- dateOfBirth: Date

a. Bind the User Bean with the User Registration Form using Spring Tags. The userId and as follows:

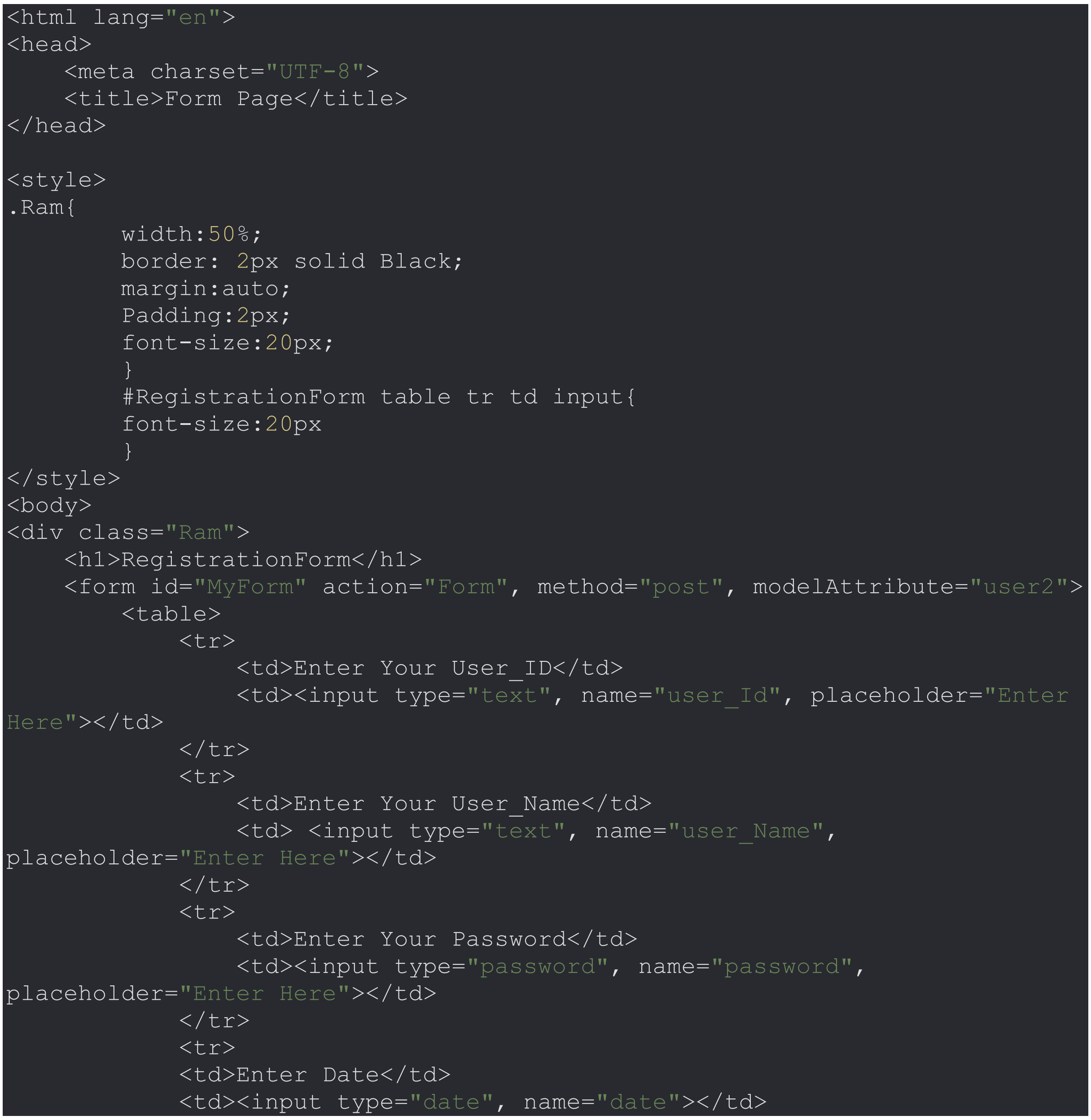
Field	Screen Object
Username	Text box
Password	Password
Gender	Radio button
Hobbies	Check box
preferredHolidayLocation	Drop Down
Age	number
DateOfBirth	Date

b. Use ModelAttribute to retrieve the User details filled in the form, generate userId and calculate age based on the dateOfBirth and display the complete User details on a display.jsp page. If the age is less than 21, the request is forwarded to an error page.

c. Use propertyEditor for dateOfBirth field.

age will not be displayed on the screen. The Screen objects for each attribute is

d. While displaying the form, the values for Hobbies and preferredHolidayLocation must be prefilled with the dynamic data coming from the server.



```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Form Page</title>
</head>

<style>
.Ram{
        width:50%;
        border: 2px solid Black;
        margin:auto;
        Padding:2px;
        font-size:20px;
        }
        #RegistrationForm table tr td input{
        font-size:20px
        }
</style>
<body>
<div class="Ram">
    <h1>RegistrationForm</h1>
    <form id="MyForm" action="Form", method="post", modelAttribute="user2">
        <table>
            <tr>
                <td>Enter Your User_ID</td>
                <td><input type="text", name="user_Id", placeholder="Enter
Here"></td>
            </tr>
            <tr>
                <td>Enter Your User_Name</td>
                <td> <input type="text", name="user_Name",
placeholder="Enter Here"></td>
            </tr>
            <tr>
                <td>Enter Your Password</td>
                <td><input type="password", name="password",
placeholder="Enter Here"></td>
            </tr>
            <tr>
            <td>Enter Date</td>
            <td><input type="date", name="date"></td>
            </tr>
            <tr>
                <td>Select Gender</td>
                <td> <input type="radio", name="gender", value="Male">Male
&nbsp; &nbsp; <input type="radio", name="gender",
value="Female">Female</td>
            </tr>
            <tr>
                <td>Enter Your age</td>
                <td><input type="number", name="age", placeholder="Enter
Your age Here"></td>
            </tr>
            <tr>
                <td>
                    Select Your Course
                </td>
                <td>
                    <select name="preferredHolidayLocation">
                        <option value="Goa">Goa</option>
                        <option value="Shimla">Shimla</option>
                        <option value="Manali">Manali</option>
                        <option value="Vietnam">C+Vietnam</option>
                        <option value="maldives">maldives</option>
                        <option value="New York">New York</option>
                    </select>
                </td>
            </tr>
            <tr>
             <td>select Your Hobbies</td>

              <td><div class="form-check form-check-inline">
                  <input class="form-check-input" type="checkbox"
name="hobby" value="Computer Games">
                  <label class="form-check-label"
for="inlineCheckbox1">Computer Games</label>
              </div>
                  <div class="form-check form-check-inline">
                      <input class="form-check-input" type="checkbox"
name="hobby" value="Outdoor games">
                      <label class="form-check-label"
for="inlineCheckbox2">Outdoor games</label>
                  </div>
                  <div class="form-check form-check-inline">
                      <input class="form-check-input" type="checkbox"
name="hobby" value="Indoor Games" >
                      <label class="form-check-label"
for="inlineCheckbox3">Indoor Games</label>
                  </div>
              </td>
            </tr>
            <tr>
                <td>

                </td>
                <td>
                    <button type="submit", >Register</button>
                    <button type="Reset", >Reset</button>
                </td>
            </tr>
        </table>
    </form>
</div>
</body>
</html>
```

```
<%@page isELIgnored="false" %>
<html>
<head>
<title>
<h1>This is an Error Page</h1>
</title>
</head>
<body>
<h1>Sorry to you Inform You that You do not fulfill the age criterion as
your age is ${user.age}</h1>
<hr>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@taglib prefix="m" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false" %>
<!DOCTYPE html>
<html>
<body>
<h1>Registration Details are : </h1>
<hr>
<h1>Name of the user is: ${user.user_Id}


<hr>

<h1>Email-ID of the user is: ${user.user_Name}</h1>

<hr>

<h1>Password selected by the user is: ${user.password}</h1>

<hr>

<h1>Date is : ${user.date}</h1>

<hr>

<h1>Gender of the user is: ${user.gender}</h1>

<hr>

<h1>Age of the user is :${user.age}</h1>

<hr>

<h1>Holiday location opted by the user is
:${user.preferredHolidayLocation}</h1>
<hr>
<h1>Hobby of the user are: </h1>
<m:forEach var="item" items="${user.hobby}">
<h3> ${item}</h3>
</m:forEach>

</body>
</html>
```

```
package com.hellocontroller;

import org.springframework.stereotype.Component;

import java.util.Arrays;
```

```
@Component
public class User {
    private String user_Id;
    private String user_Name;
    private String password;
    private String date;
    private String gender;
    private int age;
    private String preferredHolidayLocation;
    private String hobby[];


    public String getUser_Id() {
        return user_Id;
    }

    public void setUser_Id(String user_Id) {
        this.user_Id = user_Id;
    }

    public String getUser_Name() {
        return user_Name;
    }

    public void setUser_Name(String user_Name) {
        this.user_Name = user_Name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getPreferredHolidayLocation() {
        return preferredHolidayLocation;
    }

    public void setPreferredHolidayLocation(String
preferredHolidayLocation) {
        this.preferredHolidayLocation = preferredHolidayLocation;
    }

    public String[] getHobby() {
        return hobby;
    }

    public void setHobby(String[] hobby) {
        this.hobby = hobby;
    }

    @Override
    public String toString() {
        return "User{" +
                "user_Id='" + user_Id + '\'' +
                ", user_Name='" + user_Name + '\'' +
                ", password='" + password + '\'' +
                ", date='" + date + '\'' +
                ", gender='" + gender + '\'' +
                ", age=" + age +
                ", preferredHolidayLocation='" + preferredHolidayLocation +
'\'' +
                ", hobby=" + Arrays.toString(hobby) +
                '}';
    }
}
```

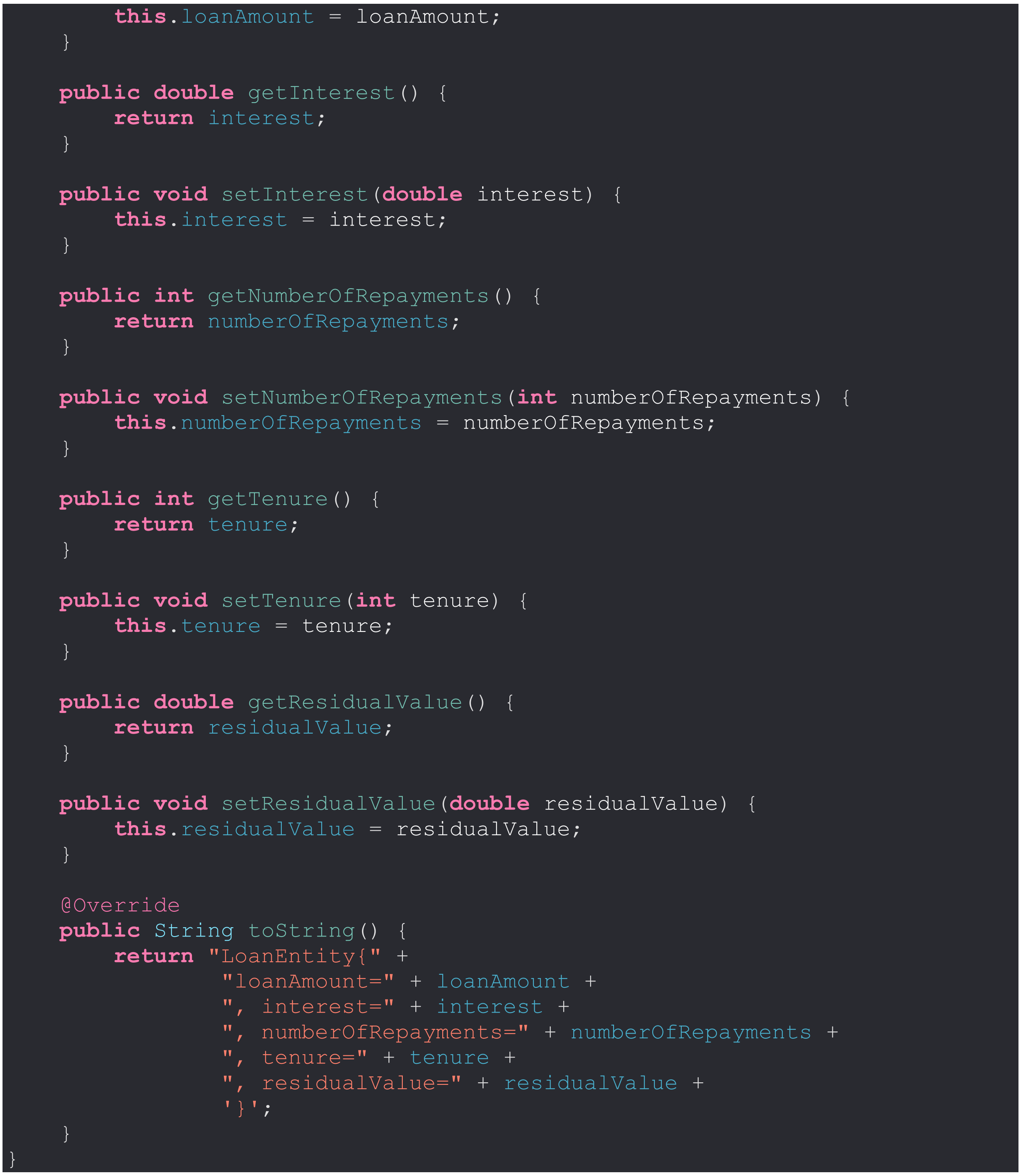
```
package com.hellocontroller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
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.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.time.LocalDateTime;
import java.util.ArrayList;
import java.util.List;

@Controller
@RequestMapping("Registration")
public class RegForm {
    @ModelAttribute
    public void commonDataForModal(Model model) {
        model.addAttribute("header", "Learn Code");
        model.addAttribute("Desc", "java is an OOP");
        System.out.println(model.getAttribute("header"));
        System.out.println(model.getAttribute("Desc"));
    }

    @RequestMapping(path = "/regd")
    public String ShowForm() {
        return "RegForm";

    }

    @RequestMapping(path="/Form",method = RequestMethod.POST)
    public String Form(@ModelAttribute User user,Model model){

//          model.addAttribute("user1",user2);//--><--Not Needed it's done
automatically by the Spring container---->
        if(user.getAge()>21)
        {
            System.out.println("redirecting");
            System.out.println(user);
            return "output";
        }
       else return "Error";
    }
```

10. Create an application to calculate the EMI by providing the required information. The EMI formula can be used from Java – Day 1 assignment. If the details are valid, EMI is calculated otherwise it throws an Exception. Write code to demonstrate the various ways available for Exception Handling.

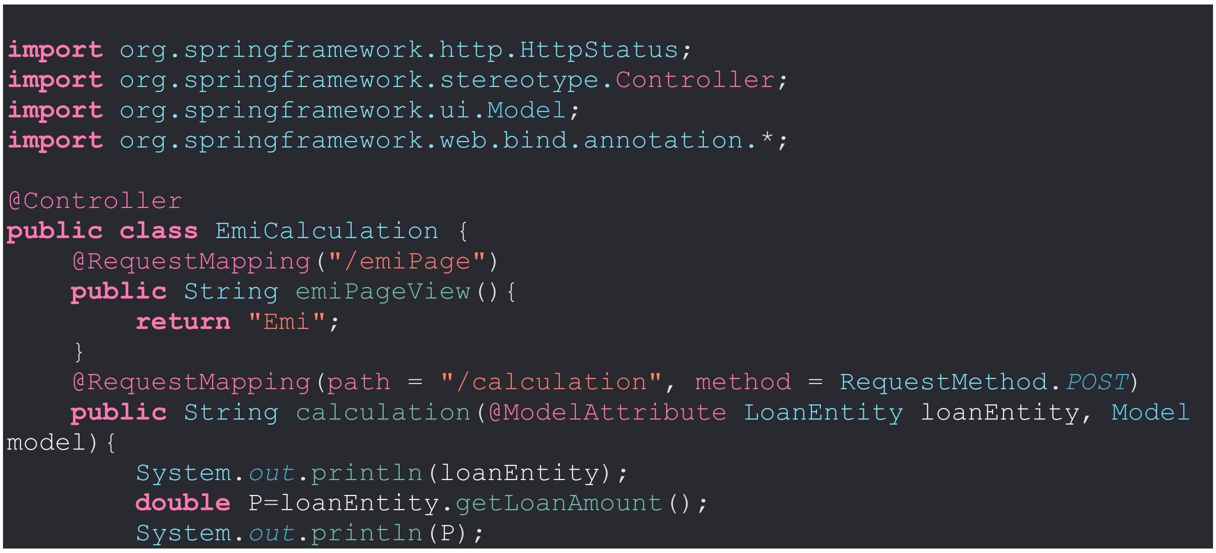
With Exception Handler Class:

```
package com.hellocontroller;

import org.springframework.http.HttpStatus;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseStatus;
@ControllerAdvice
public class myExceptionHandler {
    @ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
    @ExceptionHandler(value= NullPointerException.class)
    public String nullExceptionHandler(Model m){
        m.addAttribute("msg", "NullPointerException has Occured");
        return "Exception";
    }
    @ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
    @ExceptionHandler(value= NumberFormatException.class)
    public String numFormatExceptionHandler(Model m){
        m.addAttribute("msg", "NumberFormatException has Occured");
        return "Exception";
    }
    @ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
    @ExceptionHandler(value= Exception.class)
    public String ExceptionHandler(Model m){
        m.addAttribute("msg", "Exception");
        System.out.println("inside My Handler class");
        return "Exception";
    }

}
```

```
package com.hellocontroller;

public class LoanEntity {
    private double loanAmount;
    private double interest;
    private int numberOfRepayments;
    private int tenure;
    private double residualValue;

    public double getLoanAmount() {
        return loanAmount;
    }

    public void setLoanAmount(double loanAmount) {
```



```
        this.loanAmount = loanAmount;
    }

    public double getInterest() {
        return interest;
    }

    public void setInterest(double interest) {
        this.interest = interest;
    }

    public int getNumberOfRepayments() {
        return numberOfRepayments;
    }

    public void setNumberOfRepayments(int numberOfRepayments) {
        this.numberOfRepayments = numberOfRepayments;
    }

    public int getTenure() {
        return tenure;
    }

    public void setTenure(int tenure) {
        this.tenure = tenure;
    }

    public double getResidualValue() {
        return residualValue;
    }

    public void setResidualValue(double residualValue) {
        this.residualValue = residualValue;
    }

    @Override
    public String toString() {
        return "LoanEntity{" +
                "loanAmount=" + loanAmount +
                ", interest=" + interest +
                ", numberOfRepayments=" + numberOfRepayments +
                ", tenure=" + tenure +
                ", residualValue=" + residualValue +
                '}';
    }
}
```



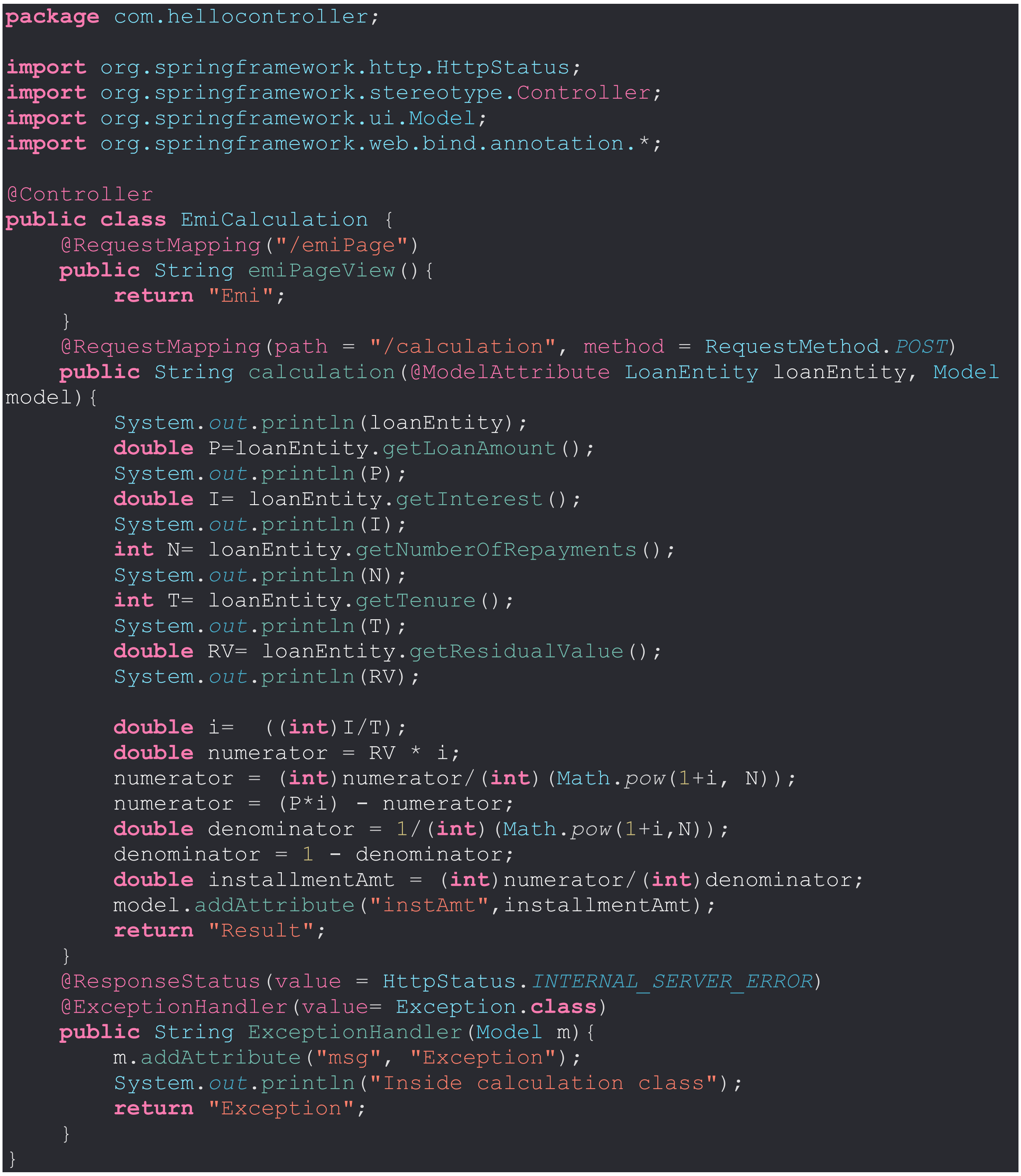
```
import org.springframework.http.HttpStatus;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
public class EmiCalculation {
    @RequestMapping("/emiPage")
    public String emiPageView(){
        return "Emi";
    }
    @RequestMapping(path = "/calculation", method = RequestMethod.POST)
    public String calculation(@ModelAttribute LoanEntity loanEntity, Model
model){
        System.out.println(loanEntity);
        double P=loanEntity.getLoanAmount();
        System.out.println(P);
        double I= loanEntity.getInterest();
        System.out.println(I);
        int N= loanEntity.getNumberOfRepayments();
        System.out.println(N);
        int T= loanEntity.getTenure();
        System.out.println(T);
        double RV= loanEntity.getResidualValue();
        System.out.println(RV);

        double i=  ((int)I/T);
        double numerator = RV * i;
        numerator = (int)numerator/(int)(Math.pow(1+i, N));
        numerator = (P*i) - numerator;
        double denominator = 1/(int)(Math.pow(1+i,N));
        denominator = 1 - denominator;
        double installmentAmt = (int)numerator/(int)denominator;
        model.addAttribute("instAmt",installmentAmt);
        return "Result";
    }
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
     .card-img-top{
     height:30%;
     width:100%;
     margin:3px;
     padding:20px;
     }
      .col-11 {
     height:100%;
     width:100%;
     margin:50px;
     padding:25px;
     }
    </style>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.c
ss" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
</head>
<body>
<div class="row row-cols-1 row-cols-md-3 g-4">
    <div class="col">
        <div class="card h-100">
            <img src="image/img_1.png" class="card-img-top" alt="image
cannot be displayed">
            <div class="card-body">
                <h5 class="card-title">Emi Calculator</h5>
                <p class="card-text">
                <form class="row g-3" action="calculation" method="post">
                    <div class="col-md-6">
                        <label for="inputpassword4" class="form-
label">Enter Original Loan Principal Amounte</label>
                        <input type="number" step="any" class="form-
control" name="loanAmount" placeholder="Enter Original Loan Principal
Amount here">
                    </div>
                      <div class="col-md-6">
                                            <label for="inputpassword4"
class="form-label">Enter interest rate per annum</label>
```

```
                                            <input type="number" step="any"
class="form-control" name="interest" placeholder="Enter interest rate per
annum here">
                    </div>
                    <div class="col-md-6">
                           <label for="inputpassword4" class="form-
label">Enter Number of payments in a year</label>
                           <input type="number" step="any" class="form-
control" name="tenure" placeholder="Enter  Number of payments in a year
here">
                    </div>
                      <div class="col-md-6">
                                            <label for="inputpassword4"
class="form-label">Enter tenure or number of installments</label>
                                            <input type="number" step="any"
class="form-control" name="numberOfRepayments" placeholder="Enter tenure or
number of installments here">
                                        </div>
                    <div class="col-md-6">
                        <label for="inputpassword4" class="form-
label">Enter Residual Value of a loan at the end of tenure</label>
                        <input type="number" step="any" class="form-
control" name="residualValue" placeholder="Enter Residual Value of a loan
at the end of tenure">
                    </div>
                <div class="col-11 text-center">
                    <button type="submit" class="btn btn-primary
">SUBMIT</button>
                </div>
            </form>
                </p>
            </div>
            <div class="card-footer">
                <small class="text-muted">This is the loan EMI calculator
page</small>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```

```
<%@page isErrorPage="true" %>
<html>
<head>
 <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.c
ss" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
<title>Sorry Page not Found !!!!! </title>
</head>
<body>
<div class="container p-3 text-center">
<img  src="image/error-404.png" class="img-gluid"/>
<h1 class="display-3">Sorry Page not Found !!!!! </h1>
<p><%=exception%></p>
<a class="btn btn-outline-Primary mt-3" href="/SprMVCEg1/index.jsp"  >Home
Page</a>

</div>
</body>
</html>
```

Without Exception Handler Class:



```
package com.hellocontroller;

import org.springframework.http.HttpStatus;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

@Controller
public class EmiCalculation {
    @RequestMapping("/emiPage")
    public String emiPageView(){
        return "Emi";
    }
    @RequestMapping(path = "/calculation", method = RequestMethod.POST)
    public String calculation(@ModelAttribute LoanEntity loanEntity, Model
model){
        System.out.println(loanEntity);
        double P=loanEntity.getLoanAmount();
        System.out.println(P);
        double I= loanEntity.getInterest();
        System.out.println(I);
        int N= loanEntity.getNumberOfRepayments();
        System.out.println(N);
        int T= loanEntity.getTenure();
        System.out.println(T);
        double RV= loanEntity.getResidualValue();
        System.out.println(RV);

        double i=  ((int)I/T);
        double numerator = RV * i;
        numerator = (int)numerator/(int)(Math.pow(1+i, N));
        numerator = (P*i) - numerator;
        double denominator = 1/(int)(Math.pow(1+i,N));
        denominator = 1 - denominator;
        double installmentAmt = (int)numerator/(int)denominator;
        model.addAttribute("instAmt",installmentAmt);
        return "Result";
    }
    @ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
    @ExceptionHandler(value= Exception.class)
    public String ExceptionHandler(Model m){
        m.addAttribute("msg", "Exception");
        System.out.println("Inside calculation class");
        return "Exception";
    }
}
```