**SPRING ASSIGNMENT Day\_8&9**

**Harshit Kushmakar| 16896**

1. Write an application to demonstrate two roles – Admin and Guest. In the Spring application created on Day 7, add two roles – Admin & User. The login is implemented using Spring Security. User can add Customers and view Customers. Admin should be able to delete, edit & View customers. Implement Spring Security using in-memory authentication and then JDBC Authentication //Using in-memory authentication //Customer.java

package com.bankApplication.model;  
import java.time.LocalDate;  
import java.util.Date;  
public class Customer {  
 private static Integer count = 100;  
 private String customerId ;  
 private String firstName;  
 private String lastName;  
 private String gender;  
 private String dateOfBirth;  
 private String contactNumber;  
 private String emailAddress;  
 private long monthlyIncome;  
 private String profession;  
 private double totalMonthlyExpense;  
 private String designation;  
 private String companyName;  
 public Customer(){  
 this.customerId = "C"+count++;  
 }  
 public Customer(String customerId, String firstName, String lastName, String   
gender, String dateOfBirth, String contactNumber, String emailAddress, long   
monthlyIncome, String profession, double totalMonthlyExpense, String designation,   
String companyName) {  
 this.customerId = customerId;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.gender = gender;  
 this.dateOfBirth = dateOfBirth;  
 this.contactNumber = contactNumber;  
 this.emailAddress = emailAddress;  
 this.monthlyIncome = monthlyIncome;  
 this.profession = profession;  
 this.totalMonthlyExpense = totalMonthlyExpense;  
 this.designation = designation;  
 this.companyName = companyName;  
 }  
 public String getCustomerId() {  
 return customerId;  
 }  
 public void setCustomerId(String customerId) {  
 this.customerId = customerId;  
 }  
 public String getFirstName() {  
 return firstName;  
 }  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
 public String getLastName() {  
 return lastName;  
 }  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
 public String getGender() {  
 return gender;  
 }  
 public void setGender(String gender) {  
 this.gender = gender;  
 }  
 public String getDateOfBirth() {  
 return dateOfBirth;  
 }  
 public void setDateOfBirth(String dateOfBirth) {  
 this.dateOfBirth = dateOfBirth;  
 }  
 public String getContactNumber() {  
 return contactNumber;  
 }  
 public void setContactNumber(String contactNumber) {  
 this.contactNumber = contactNumber;  
 }  
 public String getEmailAddress() {  
 return emailAddress;  
 }  
 public void setEmailAddress(String emailAddress) {  
 this.emailAddress = emailAddress;  
 }  
 public long getMonthlyIncome() {  
 return monthlyIncome;  
 }  
 public void setMonthlyIncome(long monthlyIncome) {  
 this.monthlyIncome = monthlyIncome;  
 }  
 public String getProfession() {  
 return profession;  
 }  
 public void setProfession(String profession) {  
 this.profession = profession;  
 }  
 public double getTotalMonthlyExpense() {  
 return totalMonthlyExpense;  
 }  
 public void setTotalMonthlyExpense(double totalMonthlyExpense) {  
 this.totalMonthlyExpense = totalMonthlyExpense;  
 }  
 public String getDesignation() {  
 return designation;  
 }  
 public void setDesignation(String designation) {  
 this.designation = designation;  
 }  
 public String getCompanyName() {  
 return companyName;  
 }  
 public void setCompanyName(String companyName) {  
 this.companyName = companyName;  
 }  
 @Override  
 public String toString() {  
 return "\n\nCustomer{" +  
 "customerId='" + customerId + '\'' +  
 ", firstName='" + firstName + '\'' +  
 ", lastName='" + lastName + '\'' +  
 ", gender='" + gender + '\'' +  
 ", dateOfBirth=" + dateOfBirth +  
 ", contactNumber='" + contactNumber + '\'' +  
 ", emailAddress='" + emailAddress + '\'' +  
 ", monthlyIncome=" + monthlyIncome +  
 ", profession='" + profession + '\'' +  
 ", totalMonthlyExpense=" + totalMonthlyExpense +  
 ", designation='" + designation + '\'' +  
 ", companyName='" + companyName + '\'' +  
 '}';  
 }  
}

//LoanAgreement.java

package com.bankApplication.model;  
import java.time.LocalDate;  
public class LoanAgreement {  
 private String agreementId ;  
 private String lesseeId;  
 private int tenure;  
 private double roi;  
 private double loanAmount;  
 private String repaymentFrequency ;  
 private String loanDisbursalDate;  
 private String status;  
 private String productCode;  
 public LoanAgreement(){}  
 public LoanAgreement(String agreementId, String lesseeId, int tenure, double roi,  
double loanAmount, String repaymentFrequency, String loanDisbursalDate, String status,  
String productCode) {  
 this.agreementId = agreementId;  
 this.lesseeId = lesseeId;  
 this.tenure = tenure;  
 this.roi = roi;  
 this.loanAmount = loanAmount;  
 this.repaymentFrequency = repaymentFrequency;  
 this.loanDisbursalDate = loanDisbursalDate;  
 this.status = status;  
 this.productCode = productCode;  
 }  
 public String getAgreementId() {  
 return agreementId;  
 }  
 public void setAgreementId(String agreementId) {  
 this.agreementId = agreementId;  
 }  
 public String getLesseeId() {  
 return lesseeId;  
 }  
 public void setLesseeId(String lesseeId) {  
 this.lesseeId = lesseeId;  
 }  
 public int getTenure() {  
 return tenure;  
 }  
 public void setTenure(int tenure) {  
 this.tenure = tenure;  
 }  
 public double getRoi() {  
 return roi;  
 }  
 public void setRoi(double roi) {  
 this.roi = roi;  
 }  
 public double getLoanAmount() {  
 return loanAmount;  
 }  
 public void setLoanAmount(double loanAmount) {  
 this.loanAmount = loanAmount;  
 }  
 public String getRepaymentFrequency() {  
 return repaymentFrequency;  
 }  
 public void setRepaymentFrequency(String repaymentFrequency) {  
 this.repaymentFrequency = repaymentFrequency;  
 }  
 public String getLoanDisbursalDate() {  
 return loanDisbursalDate;  
 }  
 public void setLoanDisbursalDate(String loanDisbursalDate) {  
 this.loanDisbursalDate = loanDisbursalDate;  
 }  
 public String getStatus() {  
 return status;  
 }  
 public void setStatus(String status) {  
 this.status = status;  
 }  
 public String getProductCode() {  
 return productCode;  
 }  
 public void setProductCode(String productCode) {  
 this.productCode = productCode;  
 }  
 @Override  
 public String toString() {  
 return "LoanAgreement{" +  
 "agreementId='" + agreementId + '\'' +  
 ", lesseeId='" + lesseeId + '\'' +  
 ", tenure=" + tenure +  
 ", roi=" + roi +  
 ", loanAmount=" + loanAmount +  
 ", repaymentFrequency='" + repaymentFrequency + '\'' +  
 ", loanDisbursalDate=" + loanDisbursalDate +  
 ", status='" + status + '\'' +  
 ", productCode='" + productCode + '\'' +  
 '}';  
 }  
}

//CustomerDao.java

package com.bankApplication.dao;  
import com.bankApplication.model.Customer;  
import org.springframework.stereotype.Repository;  
import java.sql.\*;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.List;  
@Repository  
public class CustomerDao {  
 Connection con = null;  
 Statement stmt = null;  
 public CustomerDao() {  
 try {  
 Class.forName("oracle.jdbc.driver.OracleDriver");  
 con =  
DriverManager.getConnection("jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c", "sh", "sh");  
 stmt = con.createStatement();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public void save(Customer c) {  
 try {  
 java.util.Date d = new SimpleDateFormat("yyyy-MMdd").parse(c.getDateOfBirth());  
 String sd = new SimpleDateFormat("dd/MM/yyyy").format(d);  
 System.out.println(sd);  
 String s = "insert into Customer\_16866 values('"  
 + c.getCustomerId() + "','" + c.getFirstName() + "','"  
 + c.getLastName() + "','" + c.getGender() + "','" + sd + "','"  
 + c.getContactNumber() + "','" + c.getEmailAddress() + "'," +  
c.getMonthlyIncome() + ",'"  
 + c.getProfession() + "'," + c.getTotalMonthlyExpense() + ",'" +  
c.getDesignation() + "','" + c.getCompanyName() + "')";  
 stmt.execute(s);  
 } catch (SQLException e1) {  
 e1.printStackTrace();  
 } catch (ParseException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 public String searchByCustId(String id) {  
 try {  
 ArrayList<Object> list = new ArrayList<>();  
 PreparedStatement pstmt = con.prepareStatement("select \* from  
LoanAgreement\_16886 where lessee\_id = ? ");  
 pstmt.setString(1, id);  
 ResultSet rs = pstmt.executeQuery();  
 while (rs.next()) {  
 list.add(rs.getString(1));  
 list.add(rs.getString(2));  
 list.add(rs.getInt(3));  
 list.add(rs.getDouble(4));  
 list.add(rs.getDouble(5));  
 list.add(rs.getString(6));  
 list.add(rs.getString(7));  
 list.add(rs.getString(8));  
 list.add(rs.getString(9));  
 list.add(" \n~~~\n ");  
 }  
 if (list.isEmpty()) {  
 return "Data do not exist";  
 }  
 return list.toString();  
 } catch (SQLException e1) {  
 return "Database Connection Error";  
 }  
 }  
 public String delete(String id) {  
 try {  
 PreparedStatement pstmt = con.prepareStatement("Delete from Customer\_16866  
where customer\_id = ? ");  
 pstmt.setString(1, id);  
 ResultSet rs = pstmt.executeQuery();  
 return selectAll();  
 } catch (SQLException e1) {  
 return "Database connection error";  
 }  
 }  
 public String selectAll() {  
 try {  
 PreparedStatement pstmt = con.prepareStatement("select \* from  
Customer\_16866 ");  
 ResultSet rs = pstmt.executeQuery();  
 while (rs.next()) {  
 customers.add((new Customer(rs.getString(1), rs.getString(2),  
rs.getString(3), rs.getString(4), rs.getDate(5).toString(), rs.getString(6),  
rs.getString(7), rs.getLong(8), rs.getString(9), rs.getDouble(10), rs.getString(11),  
rs.getString(12))));  
 }  
 if (customers.isEmpty()) {  
 return "Data do not exist";  
 }  
 return customers.toString();  
 } catch (SQLException e1) {  
 return "Database Connection Error";  
 }  
 }  
 public boolean updateCustomer(Customer c) {  
 try {  
 String s = "update Customer\_16866 set customerName= '" +  
 c.getFirstName() + c.getLastName() + "', dateOfBirth='" +  
 c.getDateOfBirth() +  
 "', contactNumber='" + c.getContactNumber() +  
"', emailAddress='" + c.getEmailAddress() +  
"', profession ='" + c.getProfession() +  
"',designation='" + c.getDesignation() +  
"', companyName='" + c.getCompanyName() + " where customerId=" +  
c.getCustomerId();  
 if (stmt.executeUpdate(s) > 0) {  
 return true;  
 }  
 } catch (SQLException e1) {  
 return false;  
 }  
 return false;  
 }  
 public boolean deleteCustomer(String number) {  
 try {  
 PreparedStatement pstmt = con.prepareStatement("delete from  
LMS\_CUSTOMER\_M\_ABHAYMANGAL\_16886\_ABHAY where CUSTOMER\_ID = ?");  
 pstmt.setString(1, number);  
 if (pstmt.executeUpdate() > 0) {  
 return true;  
 }  
 } catch (SQLException e1) {  
 return false;  
 }  
 return false;  
 }  
}

//LoanDao.java

package com.bankApplication.dao;  
import com.bankApplication.model.LoanAgreement;  
import java.sql.\*;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
public class LoanDao {  
 Connection con = null;  
 Statement stmt = null;  
 public LoanDao() {  
 try {  
 Class.forName("oracle.jdbc.driver.OracleDriver");  
 con =  
DriverManager.getConnection("jdbc:oracle:thin:@10.1.50.198:1535/nsbt19c", "sh", "sh");  
 stmt = con.createStatement();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
 public int save(LoanAgreement e) {  
 try {  
 Date d = new SimpleDateFormat("yyyy-MMdd").parse(e.getLoanDisbursalDate());  
 String sd = new SimpleDateFormat("dd/MM/yyyy").format(d);  
 String s = "insert into LoanAgreement\_16886 values(" + "'"  
 + e.getAgreementId() + "','" + e.getLesseeId() + "',"  
 + e.getTenure() + "," + e.getRoi() + "," + e.getLoanAmount() + ",'"  
 + e.getRepaymentFrequency() + "','" +sd + "','" + e.getStatus() +  
"','"  
 + e.getProductCode() + "')";  
 return stmt.executeUpdate(s);  
 }  
 catch (SQLException | ParseException e1) {  
 e1.printStackTrace();  
 }  
 return 0;  
 }  
 public String searchByLoanId(String id) {  
 try {  
 ArrayList<Object> list = new ArrayList<>();  
 PreparedStatement pstmt = con.prepareStatement("select \* from  
LoanAgreement\_16886 where Agreement\_ID = ? ");  
 pstmt.setString(1,id);  
 ResultSet rs = pstmt.executeQuery();  
 while (rs.next()) {  
 list.add(rs.getString(1));  
 list.add(rs.getString(2));  
 list.add(rs.getInt(3));  
 list.add(rs.getDouble(4));  
 list.add(rs.getDouble(5));  
 list.add(rs.getString(6));  
 list.add(rs.getString(7));  
 list.add(rs.getString(8));  
 list.add(rs.getString(9));  
 list.add(" \n~~~\n ");  
 }  
 if(list.isEmpty()){  
 return "Data do not exist";  
 }  
 return list.toString();  
 }  
 catch (SQLException e1) {  
 return "Database Connection Error";  
 }  
 }  
}

//AppInitializer.java

package com.bankApplication.config;  
import  
org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitia  
lizer;  
public class AppInitializer extends  
AbstractAnnotationConfigDispatcherServletInitializer {  
 @Override  
 protected String[] getServletMappings() {  
 return new String[] { "/" };  
 }  
 @Override  
 protected Class<?>[] getRootConfigClasses() {  
 return new Class[] {SecurityFilterConfig.class};  
 }  
 @Override  
 protected Class<?>[] getServletConfigClasses() {  
 return new Class[] { MVCConfig.class };  
 }  
}

//MVCConfig.java

package com.bankApplication.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.springframework.web.servlet.view.InternalResourceViewResolver;  
import org.springframework.web.servlet.view.JstlView;  
@Configuration  
@EnableWebMvc  
@ComponentScan(basePackages = {"com.bankApplication"})  
public class MVCConfig implements WebMvcConfigurer {  
 @Bean  
 public InternalResourceViewResolver resolver() {  
 InternalResourceViewResolver resolver = new InternalResourceViewResolver();  
 resolver.setViewClass(JstlView.class);  
 resolver.setPrefix("/WEB-INF/views/");  
 resolver.setSuffix(".jsp");  
 return resolver;  
 }  
}

//SecurityFilterConfig.java

package com.bankApplication.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.springframework.web.servlet.view.InternalResourceViewResolver;  
import org.springframework.web.servlet.view.JstlView;  
@Configuration  
@EnableWebMvc  
@ComponentScan(basePackages = {"com.bankApplication"})  
public class MVCConfig implements WebMvcConfigurer {  
 @Bean  
 public InternalResourceViewResolver resolver() {  
 InternalResourceViewResolver resolver = new InternalResourceViewResolver();  
 resolver.setViewClass(JstlView.class);  
 resolver.setPrefix("/WEB-INF/views/");  
 resolver.setSuffix(".jsp");  
 return resolver;  
 }  
}

//CustomerController

package com.bankApplication.controller;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import javax.validation.Valid;  
import com.bankApplication.dao.CustomerDao;  
import com.bankApplication.dao.LoanDao;  
import com.bankApplication.model.Customer;  
import com.bankApplication.model.LoanAgreement;  
import org.springframework.beans.PropertyEditorRegistry;  
import org.springframework.beans.propertyeditors.CustomDateEditor;  
import org.springframework.beans.propertyeditors.StringTrimmerEditor;  
import org.springframework.security.core.Authentication;  
import org.springframework.security.core.context.SecurityContextHolder;  
import  
org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.validation.BindingResult;  
import org.springframework.validation.DataBinder;  
import org.springframework.web.bind.WebDataBinder;  
import org.springframework.web.bind.annotation.\*;  
import org.springframework.web.servlet.ModelAndView;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
@Controller  
@RequestMapping("/customer")  
public class CustomerController {  
 @InitBinder  
 public void initBinder(WebDataBinder dataBinder) {  
 SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");  
 dataBinder.registerCustomEditor(Date.class, new CustomDateEditor(dateFormat,  
true));  
 }  
 @RequestMapping(value="/logout", method=RequestMethod.GET)  
 public String logoutPage(HttpServletRequest request, HttpServletResponse response)  
{  
 Authentication auth = SecurityContextHolder.getContext().getAuthentication();  
 if (auth != null){  
 new SecurityContextLogoutHandler().logout(request, response, auth); }  
 return "redirect:/";  
 }  
 @RequestMapping("/register")  
 public String showForm(Model theModel) {  
 theModel.addAttribute("customer", new Customer());  
 return "register";  
 }  
 @RequestMapping("/processForm")  
 public String processForm(@Valid @ModelAttribute("customer") Customer customerObj,  
BindingResult theBindingResult) {  
 if (theBindingResult.hasErrors()) {  
 return "register";  
 } else {  
 System.out.println(customerObj);  
 CustomerDao dao = new CustomerDao();  
 dao.save(customerObj);  
 return "display";  
 }  
 }  
 @RequestMapping("/search")  
 public String search(Model theModel) {  
 theModel.addAttribute("page","display");  
 return "search";  
 }  
 @RequestMapping("/searchByCustomerCode")  
 public ModelAndView searchByCustomerCode(@RequestParam("customeridser")String id )  
{  
 CustomerDao dao = new CustomerDao();  
 ModelAndView modelAndView = new ModelAndView("searchByCusId");  
 String loanData = dao.searchByCustId(id);  
 modelAndView.addObject("loan",loanData);  
 return modelAndView;  
 }  
 @RequestMapping("/searchByLoanId")  
 public ModelAndView searchByLoanId(@RequestParam("loanidser")String id ) {  
 LoanDao dao = new LoanDao();  
 ModelAndView modelAndView = new ModelAndView("searchByLoanID");  
 String loanData = dao.searchByLoanId(id);  
 modelAndView.addObject("loan",loanData);  
 return modelAndView;  
 }  
 @RequestMapping("/applyLoan")  
 public String showApplyForm(Model theModel) {  
 theModel.addAttribute("loan", new LoanAgreement());  
 return "applyloan";  
 }  
 @RequestMapping("/processLoanForm")  
 public String processLoanForm(@Valid @ModelAttribute("loan") LoanAgreement  
loanAgreement, BindingResult theBindingResult) {  
 if (theBindingResult.hasErrors()) {  
 return "applyloan";  
 } else {  
 LoanDao dao = new LoanDao();  
 dao.save(loanAgreement);  
 System.out.println(loanAgreement);  
 return "display";  
 }  
 }  
}

//index.jsp

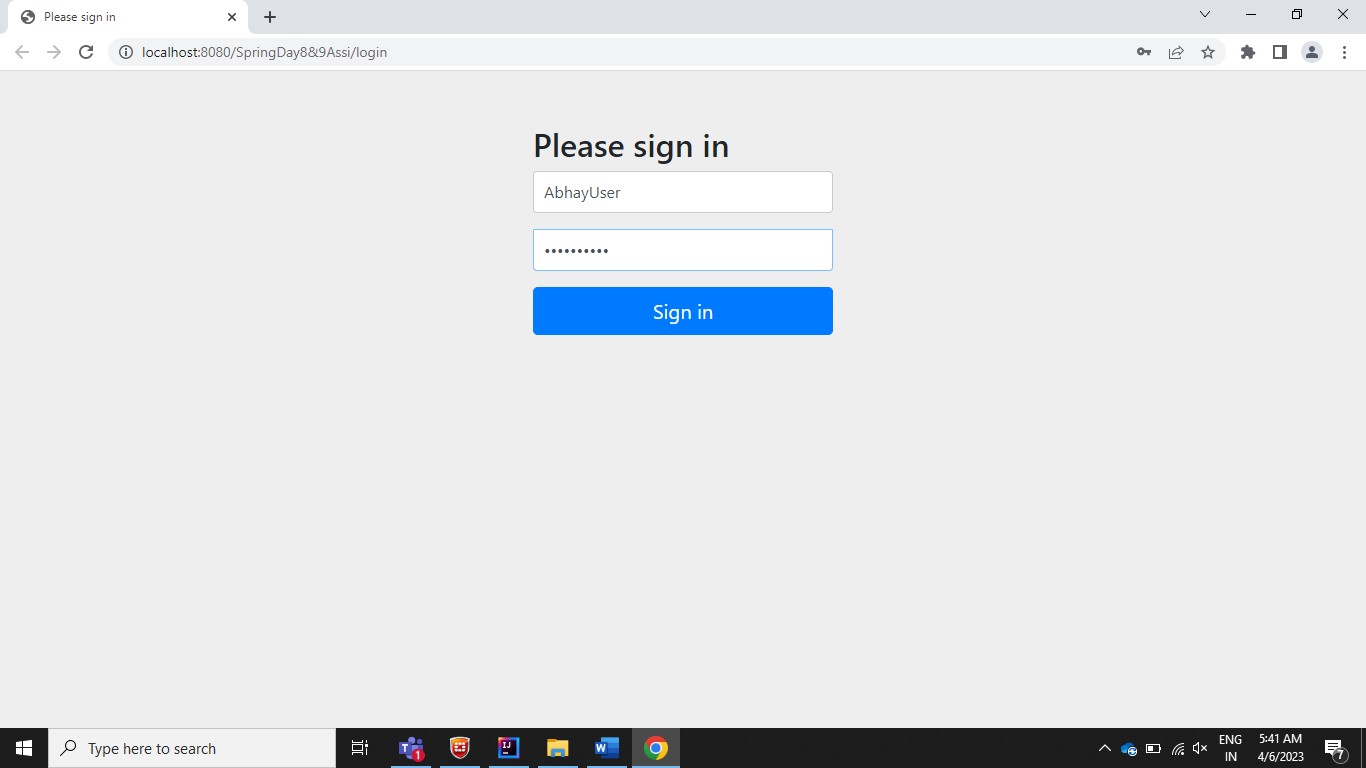
<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<body>  
<h2><a href="customer/register">Click Here to Register </a></h2>  
<h2><a href="customer/applyLoan">Click Here to Apply for Loan </a></h2>  
<h2><a href="customer/search">Click Here to Search</a></h2>  
<a href="<c:url value='customer/logout' />">Click here to logout</a>  
</body>  
</html>  
//register.jsp  
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>  
 <html>  
 <head>  
 <title>Customer Registration Form</title>  
 <style>  
 .error {  
 color: red;  
 }  
 </style>  
 </head>  
 <body>  
 <h1> Customer Registration Form</h1>  
 <i>Fill out the form. Asterisk (\*) means required.</i> <br><br>  
 <form:form action="processForm" modelAttribute="customer">  
 First Name(\*): <form:input path="firstName" />  
 <form:errors path="firstName" cssClass="error" /> <br><br>  
 Last Name : <form:input path="lastName" />  
 <form:errors path="lastName" cssClass="error" /> <br><br>  
 Gender(\*) : Male: <form:radiobutton path="gender" value="M" />  
 Female: <form:radiobutton path="gender" value="F" />  
 <form:errors path="gender" cssClass="error" /> <br><br>  
 DOB(\*) : <form:input path="dateOfBirth" type="date"/><br><br>  
 Contact Number(\*): <form:input path="contactNumber" />  
 <form:errors path="contactNumber" cssClass="error" /> <br><br>  
 Email Address(\*): <form:input path="emailAddress" />  
 <form:errors path="emailAddress" cssClass="error"  
/> <br><br>  
 Monthly Income(\*): <form:input path="monthlyIncome" />  
 <form:errors path="monthlyIncome" cssClass="error" /> <br><br>  
 Profession: <form:input path="profession" />  
 <form:errors path="profession" cssClass="error" /> <br><br>  
 Total Monthly Expense(\*): <form:input path="totalMonthlyExpense" />  
 <form:errors path="totalMonthlyExpense" cssClass="error" />  
<br><br>  
 Designation: <form:input path="designation" />  
 <form:errors path="designation" cssClass="error" /> <br><br>  
 Company Name: <form:input path="companyName" />  
 <form:errors path="companyName" cssClass="error" /> <br><br>  
 <input type="submit" value="Submit" />  
 </form:form>  
 </body>  
 </html>

//applyloan.jsp

<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>  
<html>  
<head>  
<title>Customer Registration Form</title>  
<style>  
.error {  
color: red;  
}  
</style>  
</head>  
<body>  
<h1> Apply For Loan</h1>  
<i>Fill out the form. Asterisk (\*) means required.</i> <br><br>  
<form:form action="processLoanForm" modelAttribute="loan">  
Agreement ID(\*): <form:input path="agreementId" />  
<form:errors path="agreementId" cssClass="error" /> <br><br>  
Lessee Id(\*) : <form:input path="lesseeId" />  
<form:errors path="lesseeId" cssClass="error" /> <br><br>  
Tenure : <form:input path="tenure" />  
<form:errors path="tenure" cssClass="error" /> <br><br>  
Rate : <form:input path="roi" />  
<form:errors path="roi" cssClass="error" /> <br><br>  
Loan Amount : <form:input path="loanAmount" />  
<form:errors path="loanAmount" cssClass="error" /> <br><br>  
Repayment Frequency(\*) : <form:input path="repaymentFrequency" />  
<form:errors path="repaymentFrequency" cssClass="error" /> <br><br>  
loanDisbursalDate : <form:input path="loanDisbursalDate" type="date"/><br><br>  
Status(\*) : <form:input path="status" />  
<form:errors path="status" cssClass="error" /> <br><br>  
Product Code(\*) : <form:input path="productCode" />  
<form:errors path="productCode" cssClass="error" /> <br><br>  
<input type="submit" value="Apply" />  
</form:form>  
</body>  
</html>  
//search.jsp  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
<form action="searchByCustomerCode" >  
 Search by Customer Id : <input type="search" name="customeridser"><br><br>  
 <input type="submit" value="submit">  
</form>  
<form action="searchByLoanId" >  
 Search by Loan Id : <input type="search" name="loanidser"><br><br>  
 <input type="submit" value="submit">  
</form>  
</body>  
</html>

//display.jsp

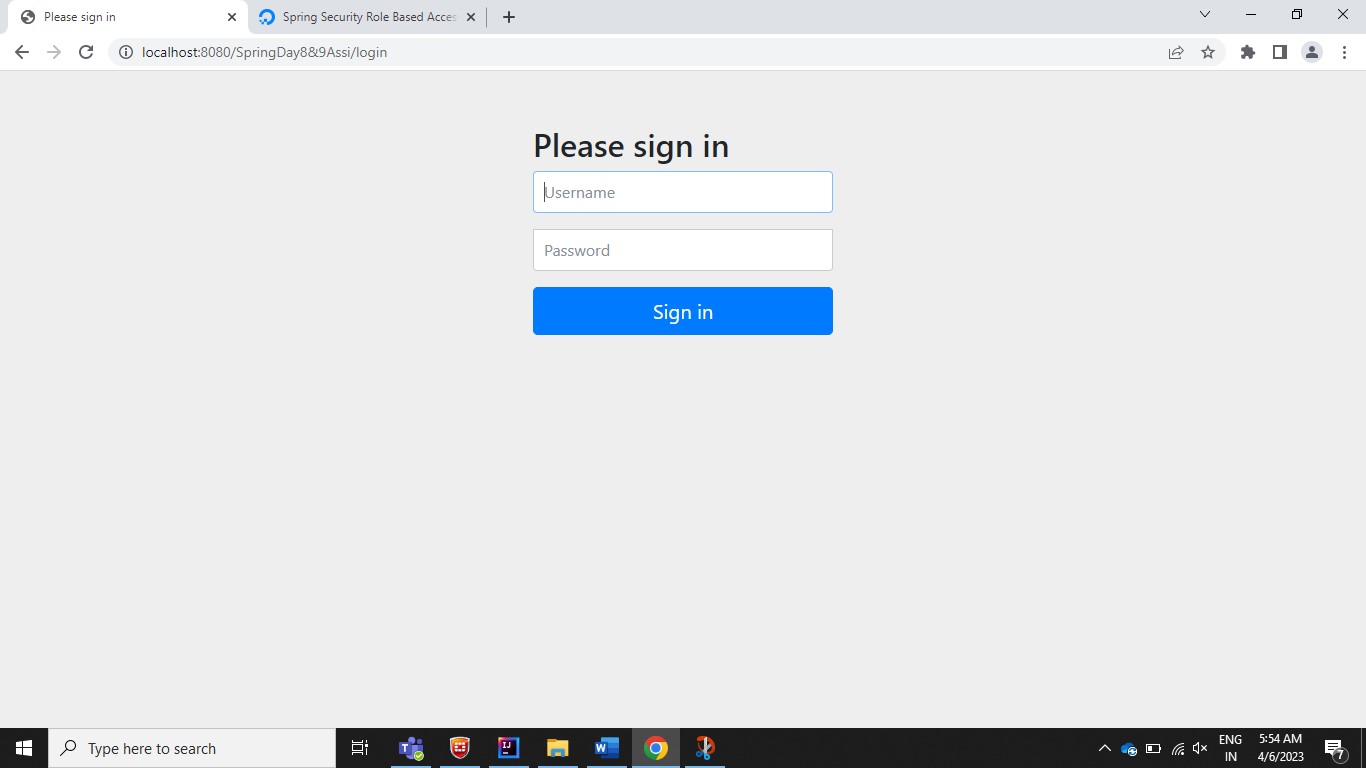
<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head></head>  
<body>  
<a href="<c:url value='customer/logout' />">Click here to logout</a>  
<h1> Register Successfully </h1>  
</body>  
</html>  
//searchByCustId.jsp  
<%@ page language="java" contentType="text/html; charset=UTF-8"  
 pageEncoding="UTF-8"%>  
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head></head>  
<body>  
<a href="<c:url value='customer/logout' />">Click here to logout</a>  
<h1> Register Successfully </h1>  
</body>  
</html>  
//searchByLoanId.jsp  
<%@ page isELIgnored = "false" %>  
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="ISO-8859-1">  
<title>User Data</title>  
</head>  
<body>  
<h3>Loan Details as per Customer Code :</h3>  
${loan}  
</body>  
</html>



Graphical user interface, text, application, email

Description automatically generated

//when click on logout

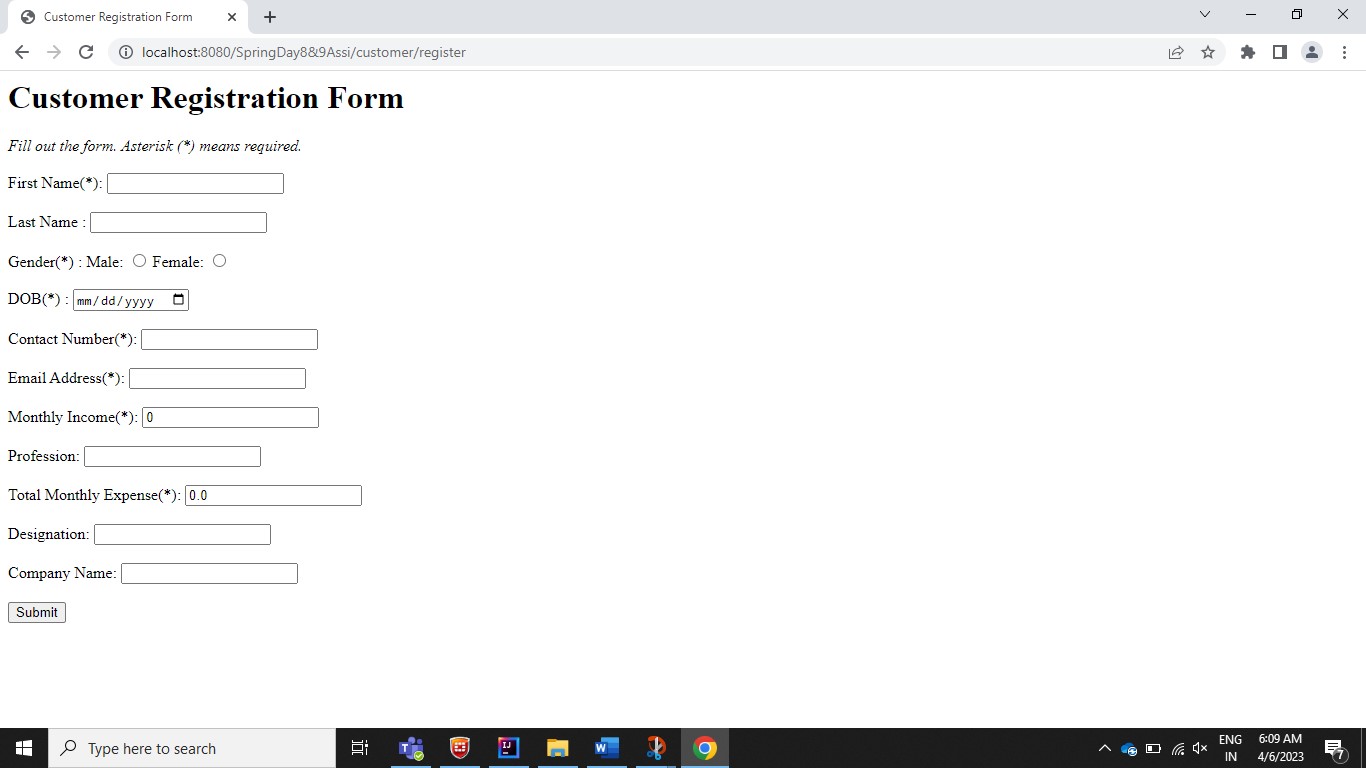


//ADMIN can only access Search and USER can access ALL URLs

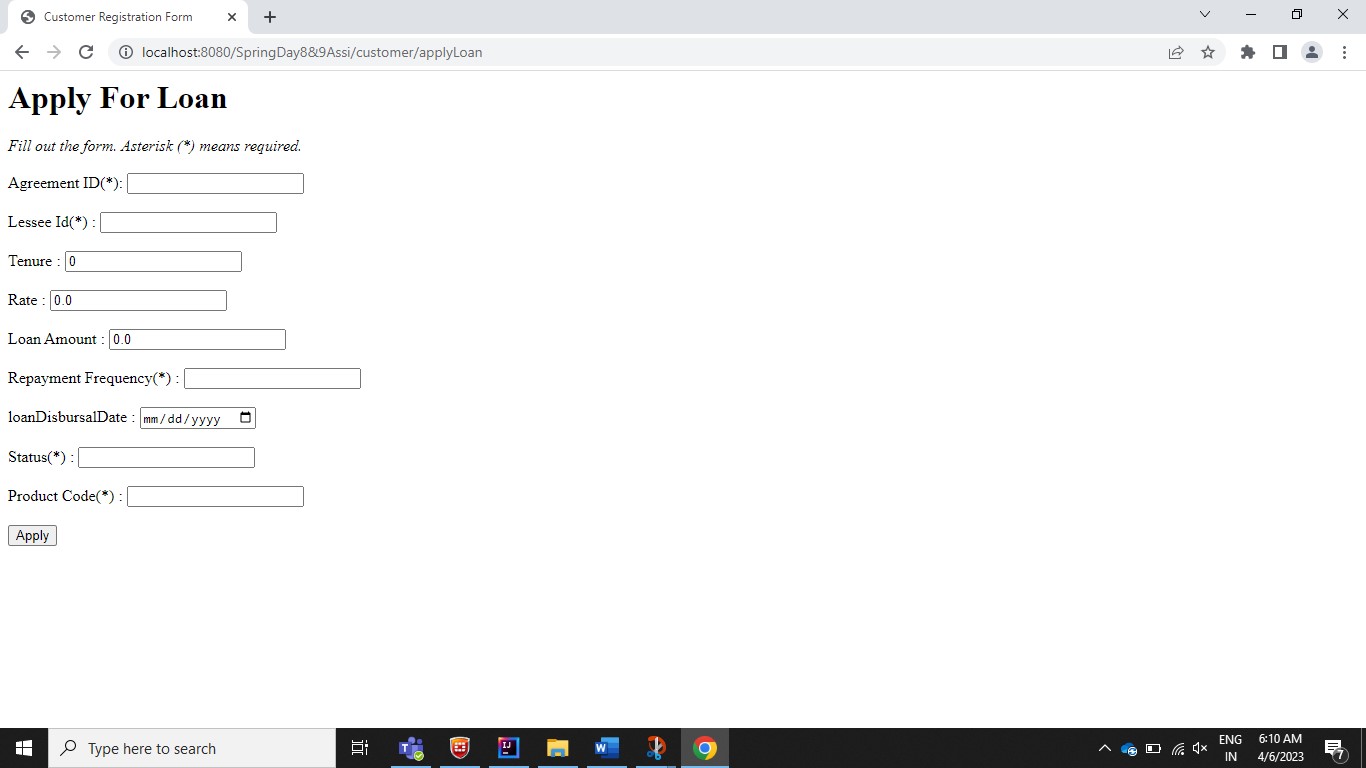
Graphical user interface, text, application, email

Description automatically generated

//only USER role can access



//only USER role can access



//Both USER & ADMIN role can access

Graphical user interface, application

Description automatically generated