|  |
| --- |
| ICIN BANK-Online Banking  **(Source Code)** |

Contents

Contents

[1. Project Modules: 1](#_Toc96438426)

[2. Features of the application: 2](#_Toc96438427)

[3. GitHub Link. 2](#_Toc96438428)

[4. Folder Structure: 2](#_Toc96438429)

[5. Project Code: 2](#_Toc96438430)

# Project Modules:

* Node js.
* Visual Studios.
* Angular 13.3.1
* Angular .json
* HTML
* Css
* Java Scriot
* karma configuration model.
* Sql Data Base
* PowerShell

# Features of the application:

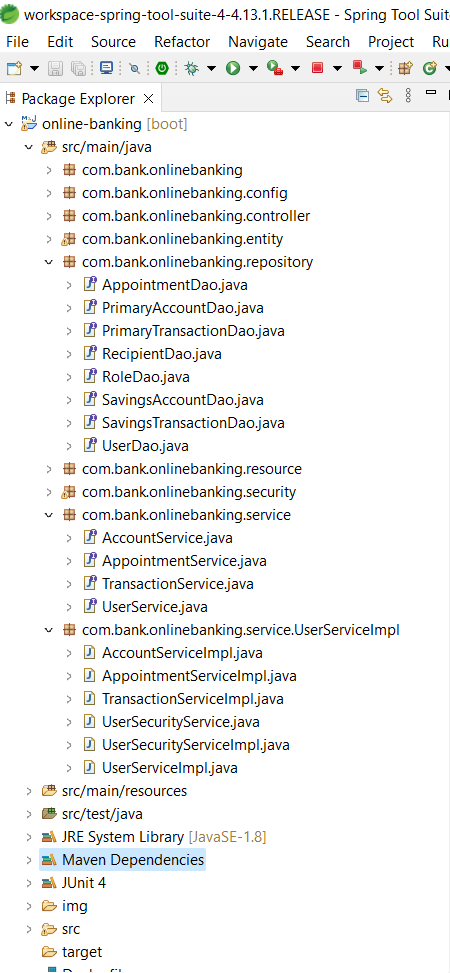
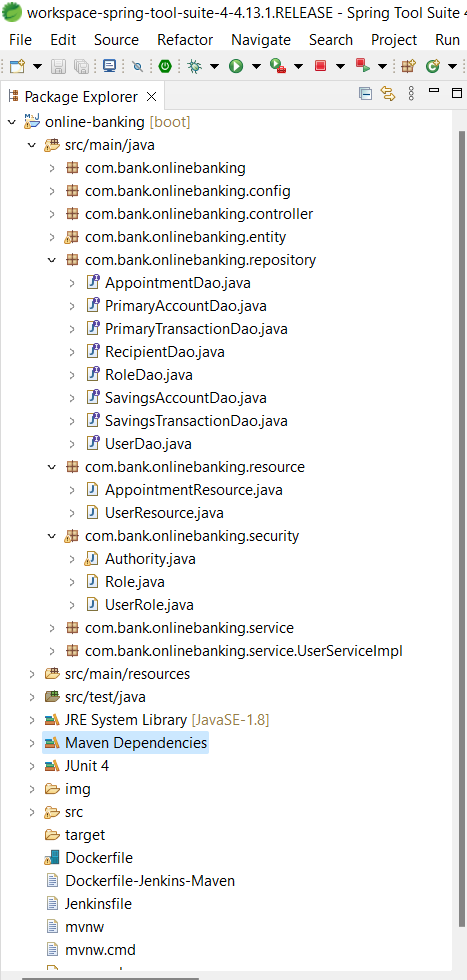
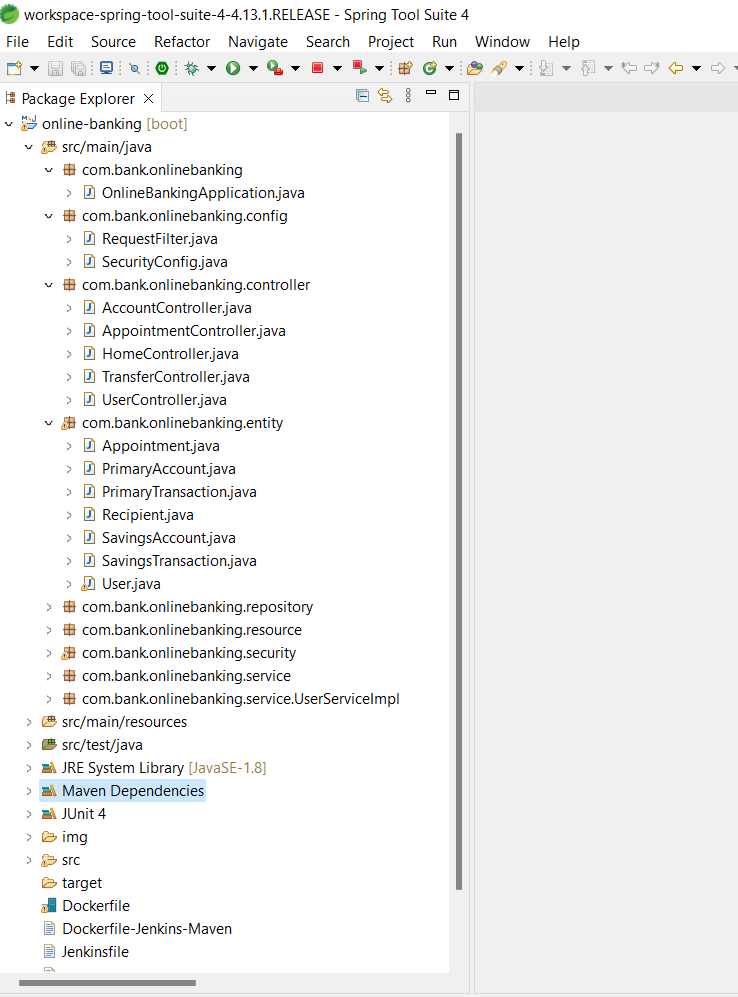
* Registration
* Login
* Account transactions
* Transfers
* Savings details
* Profile settings
* Requesting cheque books

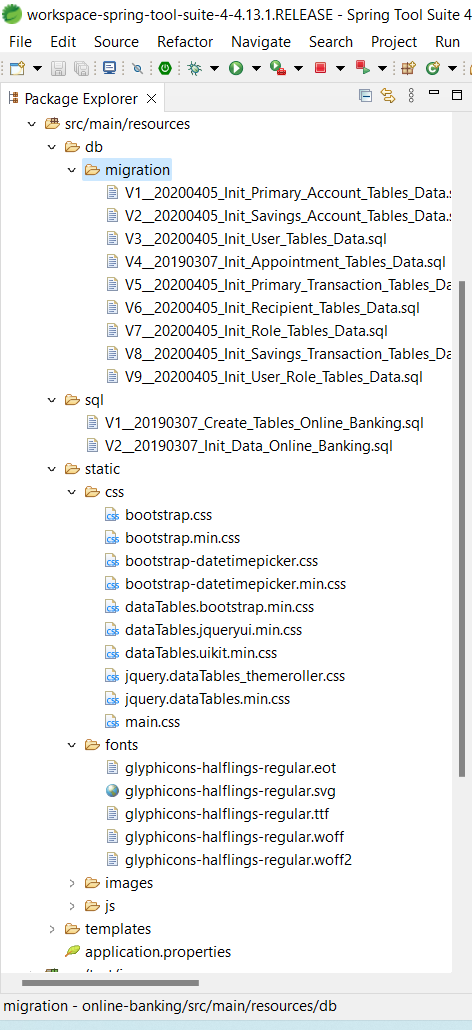
# GitHub Link.

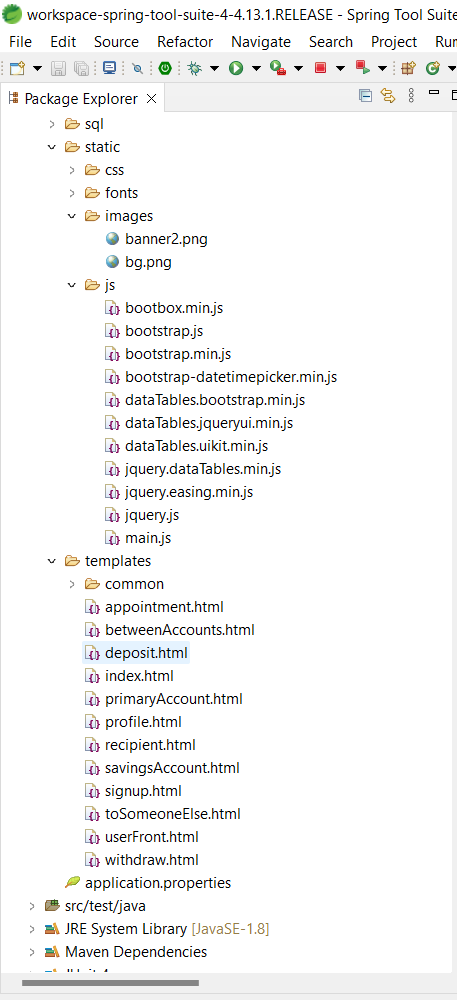
https://github.com/skibrahimsk/Final-Project

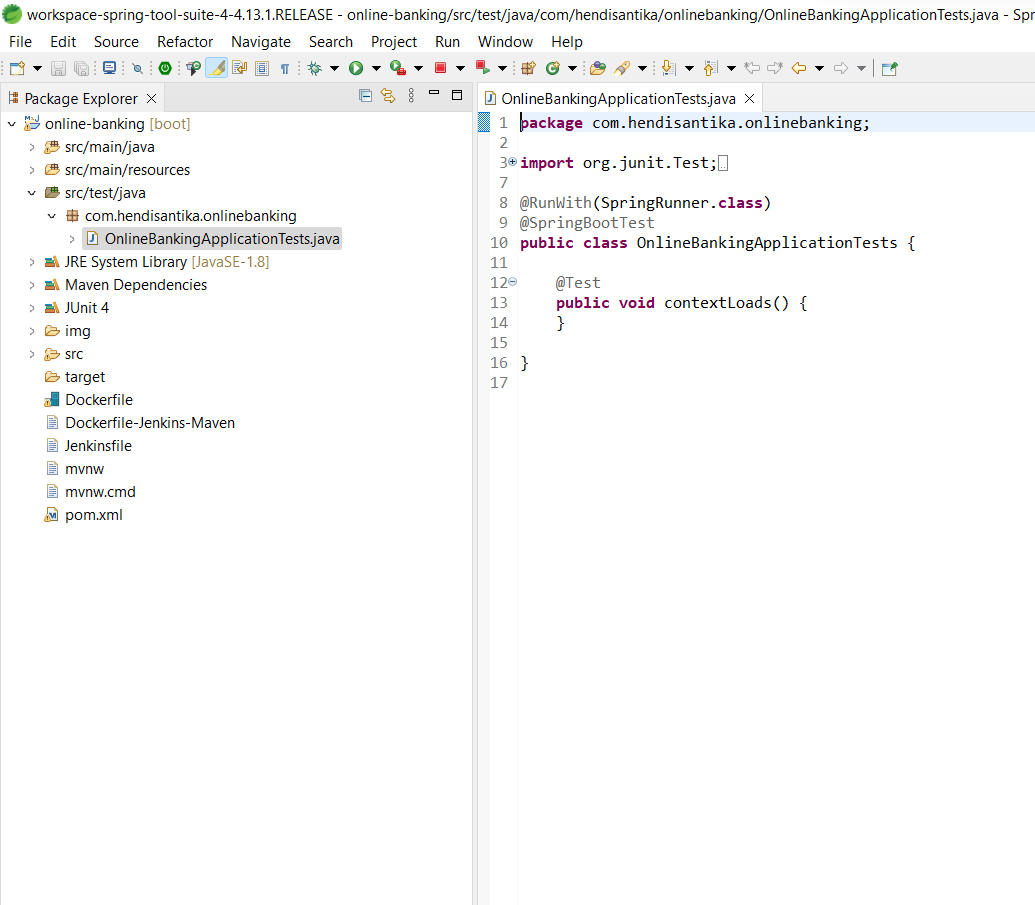
# Folder Structure:

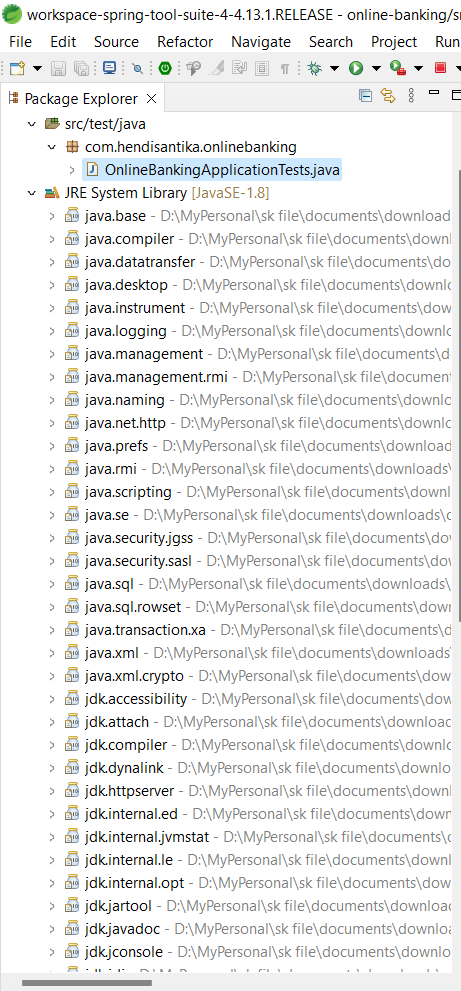
|  |
| --- |
| **Folder Structure** |
|  |

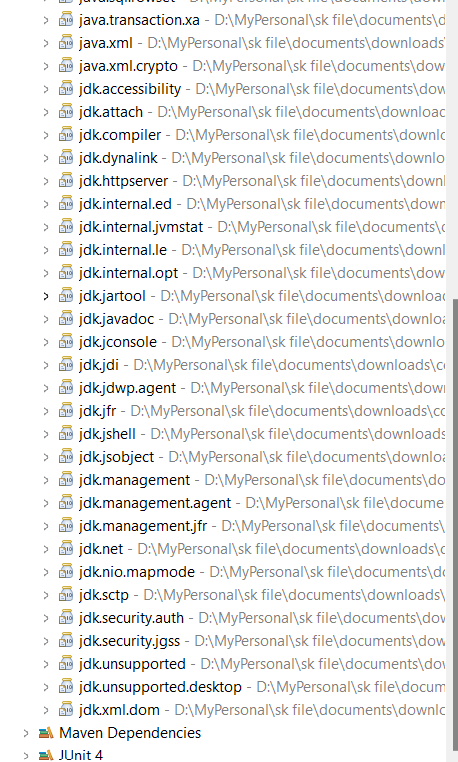




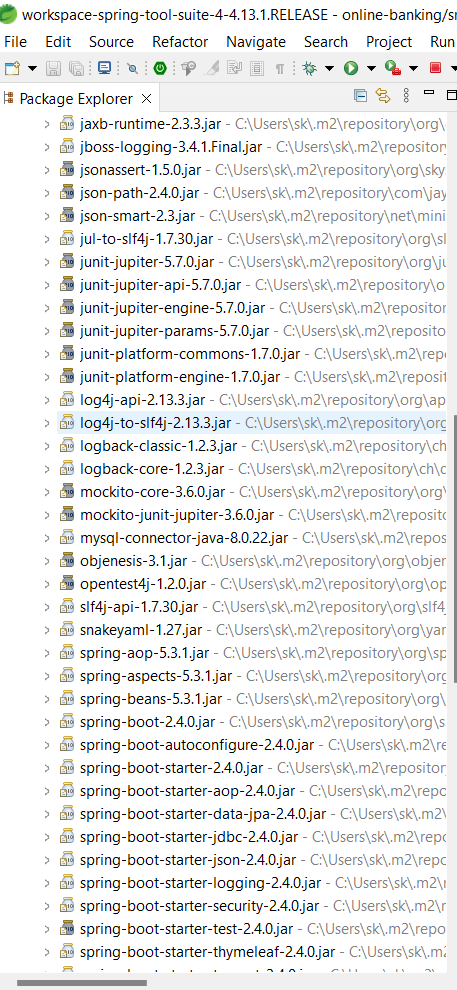


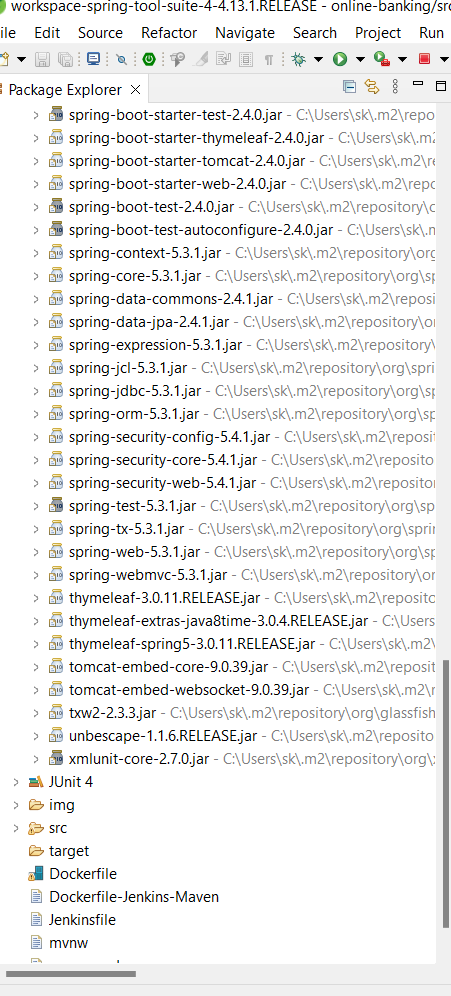


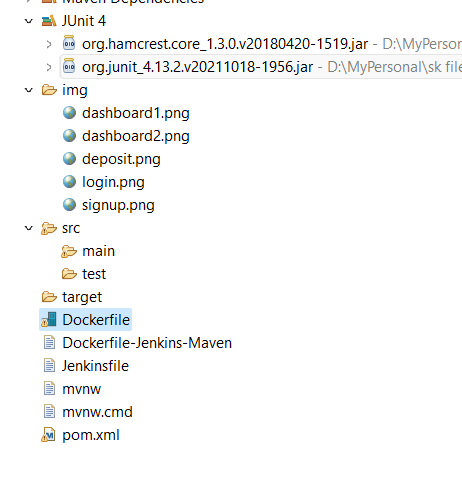












# Project Code:

1. **Pom.Xml**

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

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.hendisantika</groupId>

<artifactId>online-banking</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>online-banking</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.4.0</version>

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-

8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

<dependency

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/org.flywaydb/flyway-core -->

<dependency>

<groupId>org.flywaydb</groupId>

<artifactId>flyway-core</artifactId>

<version>5.2.4</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId> </plugin>

</plugins>

</build>

</project>

1. OnlineBankingApplication:

package com.bank.onlinebanking;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class OnlineBankingApplication {

public static void main(String[] args) {

SpringApplication.run(OnlineBankingApplication.class, args);

}

1. **RequestFilter.java:**

package com.bank.onlinebanking.config;

import org.springframework.core.Ordered;

import org.springframework.core.annotation.Order;

import org.springframework.stereotype.Component;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.27

\* To change this template use File | Settings | File Templates.

\*/

@Component

@Order(Ordered.HIGHEST\_PRECEDENCE)

public class RequestFilter implements Filter {

public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) {

HttpServletResponse response = (HttpServletResponse) res;

HttpServletRequest request = (HttpServletRequest) req;

response.setHeader("Access-Control-Allow-Origin", "http://localhost:4200");

response.setHeader("Access-Control-Allow-Methods", "POST, PUT, GET, OPTIONS, DELETE");

response.setHeader("Access-Control-Allow-Headers", "x-requested-with");

response.setHeader("Access-Control-Max-Age", "3600");

response.setHeader("Access-Control-Allow-Credentials", "true");

if (!(request.getMethod().equalsIgnoreCase("OPTIONS"))) {

try {

chain.doFilter(req, res);

} catch (Exception e) {

e.printStackTrace();

}

} else {

System.out.println("Pre-flight");

response.setHeader("Access-Control-Allow-Methods", "POST,GET,DELETE");

response.setHeader("Access-Control-Max-Age", "3600");

response.setHeader("Access-Control-Allow-Headers", "authorization, content-type," +

"access-control-request-headers,access-control-request-method,accept,origin,authorization,x-requested-with");

response.setStatus(HttpServletResponse.SC\_OK);

}

}

public void init(FilterConfig filterConfig) {

}

public void destroy() {

}

}

1. **SecurityConfig.java:**

package com.bank.onlinebanking.config;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;

import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;

import org.springframework.security.config.annotation.web.builders.HttpSecurity;

import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.web.util.matcher.AntPathRequestMatcher;

import com.bank.onlinebanking.service.UserServiceImpl.UserSecurityService;

import java.security.SecureRandom;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.29

\* To change this template use File | Settings | File Templates.

\*/

@Configuration

@EnableWebSecurity

@EnableGlobalMethodSecurity(prePostEnabled = true)

public class SecurityConfig extends WebSecurityConfigurerAdapter {

// @Autowired

// private Environment env;

private static final String SALT = "salt"; // Salt should be protected carefully

private static final String[] PUBLIC\_MATCHERS = {

"/webjars/\*\*",

"/css/\*\*",

"/js/\*\*",

"/images/\*\*",

"/",

"/about/\*\*",

"/contact/\*\*",

"/error/\*\*/\*",

"/console/\*\*",

"/signup"

};

@Autowired

private UserSecurityService userSecurityService;

@Bean

public BCryptPasswordEncoder passwordEncoder() {

return new BCryptPasswordEncoder(12, new SecureRandom(SALT.getBytes()));

}

@Override

protected void configure(HttpSecurity http) throws Exception {

http

.authorizeRequests().

// antMatchers("/\*\*").

antMatchers(PUBLIC\_MATCHERS).

permitAll().anyRequest().authenticated();

http

.csrf().disable().cors().disable()

.formLogin().failureUrl("/index?error").defaultSuccessUrl("/userFront").loginPage("/index").permitAll()

.and()

.logout().logoutRequestMatcher(new AntPathRequestMatcher("/logout")).logoutSuccessUrl("/index?logout").deleteCookies("remember-me").permitAll()

.and()

.rememberMe();

}

@Autowired

public void configureGlobal(AuthenticationManagerBuilder auth) throws Exception {

// auth.inMemoryAuthentication().withUser("user").password("password").roles("USER"); //This is in-memory authentication

auth.userDetailsService(userSecurityService).passwordEncoder(passwordEncoder());

}

}

1. **AccountController.java:**

package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.PrimaryAccount;

import com.bank.onlinebanking.entity.PrimaryTransaction;

import com.bank.onlinebanking.entity.SavingsAccount;

import com.bank.onlinebanking.entity.SavingsTransaction;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.service.AccountService;

import com.bank.onlinebanking.service.TransactionService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import java.security.Principal;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* To change this template use File | Settings | File Templates.

\*/

@Controller

@RequestMapping("/account")

public class AccountController {

@Autowired

private UserService userService;

@Autowired

private AccountService accountService;

@Autowired

private TransactionService transactionService;

@RequestMapping("/primaryAccount")

public String primaryAccount(Model model, Principal principal) {

List<PrimaryTransaction> primaryTransactionList = transactionService.findPrimaryTransactionList(principal.getName());

User user = userService.findByUsername(principal.getName());

PrimaryAccount primaryAccount = user.getPrimaryAccount();

model.addAttribute("primaryAccount", primaryAccount);

model.addAttribute("primaryTransactionList", primaryTransactionList);

return "primaryAccount";

}

@RequestMapping("/savingsAccount")

public String savingsAccount(Model model, Principal principal) {

List<SavingsTransaction> savingsTransactionList = transactionService.findSavingsTransactionList(principal.getName());

User user = userService.findByUsername(principal.getName());

SavingsAccount savingsAccount = user.getSavingsAccount();

model.addAttribute("savingsAccount", savingsAccount);

model.addAttribute("savingsTransactionList", savingsTransactionList);

return "savingsAccount";

}

@RequestMapping(value = "/deposit", method = RequestMethod.GET)

public String deposit(Model model) {

model.addAttribute("accountType", "");

model.addAttribute("amount", "");

return "deposit";

}

@RequestMapping(value = "/deposit", method = RequestMethod.POST)

public String depositPOST(@ModelAttribute("amount") String amount, @ModelAttribute("accountType") String accountType, Principal principal) {

accountService.deposit(accountType, Double.parseDouble(amount), principal);

return "redirect:/userFront";

}

@RequestMapping(value = "/withdraw", method = RequestMethod.GET)

public String withdraw(Model model) {

model.addAttribute("accountType", "");

model.addAttribute("amount", "");

return "withdraw";

}

@RequestMapping(value = "/withdraw", method = RequestMethod.POST)

public String withdrawPOST(@ModelAttribute("amount") String amount, @ModelAttribute("accountType") String accountType, Principal principal) {

accountService.withdraw(accountType, Double.parseDouble(amount), principal);

return "redirect:/userFront";

}

}

1. **AppointmentController.java:**

package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.Appointment;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.service.AppointmentService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import java.security.Principal;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.35

\* To change this template use File | Settings | File Templates.

\*/

@Controller

@RequestMapping("/appointment")

public class AppointmentController {

@Autowired

private AppointmentService appointmentService;

@Autowired

private UserService userService;

@RequestMapping(value = "/create", method = RequestMethod.GET)

public String createAppointment(Model model) {

Appointment appointment = new Appointment();

model.addAttribute("appointment", appointment);

model.addAttribute("dateString", "");

return "appointment";

}

@RequestMapping(value = "/create", method = RequestMethod.POST)

public String createAppointmentPost(@ModelAttribute("appointment") Appointment appointment, @ModelAttribute("dateString") String date, Model model, Principal principal) throws ParseException {

SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd hh:mm");

Date d1 = format1.parse(date);

appointment.setDate(d1);

User user = userService.findByUsername(principal.getName());

appointment.setUser(user);

appointmentService.createAppointment(appointment);

return "redirect:/userFront";

}

}

1. **HomeController.java:**

package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.PrimaryAccount;

import com.bank.onlinebanking.entity.SavingsAccount;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.RoleDao;

import com.bank.onlinebanking.security.UserRole;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import java.security.Principal;

import java.util.HashSet;

import java.util.Set;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.34

\* To change this template use File | Settings | File Templates.

\*/

@Controller

public class HomeController {

@Autowired

private UserService userService;

@Autowired

private RoleDao roleDao;

@RequestMapping("/")

public String home() {

return "redirect:/index";

}

@RequestMapping("/index")

public String index() {

return "index";

}

@GetMapping("/signup")

public String signup(Model model) {

User user = new User();

model.addAttribute("user", user);

return "signup";

}

@PostMapping("/signup")

public String signupPost(@ModelAttribute("user") User user, Model model) {

if (userService.checkUserExists(user.getUsername(), user.getEmail())) {

if (userService.checkEmailExists(user.getEmail())) {

model.addAttribute("emailExists", true);

}

if (userService.checkUsernameExists(user.getUsername())) {

model.addAttribute("usernameExists", true);

}

return "signup";

} else {

Set<UserRole> userRoles = new HashSet<>();

userRoles.add(new UserRole(user, roleDao.findByName("ROLE\_USER")));

userService.createUser(user, userRoles);

return "redirect:/";

}

}

@GetMapping("/userFront")

public String userFront(Principal principal, Model model) {

User user = userService.findByUsername(principal.getName());

PrimaryAccount primaryAccount = user.getPrimaryAccount();

SavingsAccount savingsAccount = user.getSavingsAccount();

model.addAttribute("primaryAccount", primaryAccount);

model.addAttribute("savingsAccount", savingsAccount);

return "userFront";

}

}

1. **TransferController.java**

package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.PrimaryAccount;

import com.bank.onlinebanking.entity.Recipient;

import com.bank.onlinebanking.entity.SavingsAccount;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.service.TransactionService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Controller;

import org.springframework.transaction.annotation.Transactional;

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.security.Principal;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.37

\* To change this template use File | Settings | File Templates.

\*/

@Controller

@RequestMapping("/transfer")

public class TransferController {

@Autowired

private TransactionService transactionService;

@Autowired

private UserService userService;

@RequestMapping(value = "/betweenAccounts", method = RequestMethod.GET)

public String betweenAccounts(Model model) {

model.addAttribute("transferFrom", "");

model.addAttribute("transferTo", "");

model.addAttribute("amount", "");

return "betweenAccounts";

}

@RequestMapping(value = "/betweenAccounts", method = RequestMethod.POST)

public String betweenAccountsPost(

@ModelAttribute("transferFrom") String transferFrom,

@ModelAttribute("transferTo") String transferTo,

@ModelAttribute("amount") String amount,

Principal principal

) throws Exception {

User user = userService.findByUsername(principal.getName());

PrimaryAccount primaryAccount = user.getPrimaryAccount();

SavingsAccount savingsAccount = user.getSavingsAccount();

transactionService.betweenAccountsTransfer(transferFrom, transferTo, amount, primaryAccount, savingsAccount);

return "redirect:/userFront";

}

@RequestMapping(value = "/recipient", method = RequestMethod.GET)

public String recipient(Model model, Principal principal) {

List<Recipient> recipientList = transactionService.findRecipientList(principal);

Recipient recipient = new Recipient();

model.addAttribute("recipientList", recipientList);

model.addAttribute("recipient", recipient);

return "recipient";

}

@RequestMapping(value = "/recipient/save", method = RequestMethod.POST)

public String recipientPost(@ModelAttribute("recipient") Recipient recipient, Principal principal) {

User user = userService.findByUsername(principal.getName());

recipient.setUser(user);

transactionService.saveRecipient(recipient);

return "redirect:/transfer/recipient";

}

@RequestMapping(value = "/recipient/edit", method = RequestMethod.GET)

public String recipientEdit(@RequestParam(value = "recipientName") String recipientName, Model model, Principal principal) {

Recipient recipient = transactionService.findRecipientByName(recipientName);

List<Recipient> recipientList = transactionService.findRecipientList(principal);

model.addAttribute("recipientList", recipientList);

model.addAttribute("recipient", recipient);

return "recipient";

}

@RequestMapping(value = "/recipient/delete", method = RequestMethod.GET)

@Transactional

public String recipientDelete(@RequestParam(value = "recipientName") String recipientName, Model model, Principal principal) {

transactionService.deleteRecipientByName(recipientName);

List<Recipient> recipientList = transactionService.findRecipientList(principal);

Recipient recipient = new Recipient();

model.addAttribute("recipient", recipient);

model.addAttribute("recipientList", recipientList);

return "recipient";

}

@RequestMapping(value = "/toSomeoneElse", method = RequestMethod.GET)

public String toSomeoneElse(Model model, Principal principal) {

List<Recipient> recipientList = transactionService.findRecipientList(principal);

model.addAttribute("recipientList", recipientList);

model.addAttribute("accountType", "");

return "toSomeoneElse";

}

@RequestMapping(value = "/toSomeoneElse", method = RequestMethod.POST)

public String toSomeoneElsePost(@ModelAttribute("recipientName") String recipientName, @ModelAttribute("accountType") String accountType, @ModelAttribute("amount") String amount, Principal principal) {

User user = userService.findByUsername(principal.getName());

Recipient recipient = transactionService.findRecipientByName(recipientName);

transactionService.toSomeoneElseTransfer(recipient, accountType, amount, user.getPrimaryAccount(), user.getSavingsAccount());

return "redirect:/userFront";

}

}

1. **UserController.java**

package com.bank.onlinebanking.controller;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import java.security.Principal;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.38

\* To change this template use File | Settings | File Templates.

\*/

@Controller

@RequestMapping("/user")

public class UserController {

@Autowired

private UserService userService;

@GetMapping("/profile")

public String profile(Principal principal, Model model) {

User user = userService.findByUsername(principal.getName());

model.addAttribute("user", user);

return "profile";

}

@PostMapping("/profile")

public String profilePost(@ModelAttribute("user") User newUser, Model model) {

User user = userService.findByUsername(newUser.getUsername());

user.setUsername(newUser.getUsername());

user.setFirstName(newUser.getFirstName());

user.setLastName(newUser.getLastName());

user.setEmail(newUser.getEmail());

user.setPhone(newUser.getPhone());

model.addAttribute("user", user);

userService.saveUser(user);

return "profile";

}

}

1. **Appointment.java:**

**package com.bank.onlinebanking.entity;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.JoinColumn;**

**import javax.persistence.ManyToOne;**

**import java.util.Date;**

**/\*\***

**\* Created by IntelliJ IDEA.**

**\* Project : online-banking**

**\* Time: 06.46**

**\* To change this template use File | Settings | File Templates.**

**\*/**

**@Entity**

**public class Appointment {**

**@Id**

**@GeneratedValue(strategy = GenerationType.IDENTITY)**

**private Long id;**

**private Date date;**

**private String location;**

**private String description;**

**private boolean confirmed;**

**@ManyToOne**

**@JoinColumn(name = "user\_id")**

**private User user;**

**public Long getId() {**

**return id;**

**}**

**public void setId(Long id) {**

**this.id = id;**

**}**

**public Date getDate() {**

**return date;**

**}**

**public void setDate(Date date) {**

**this.date = date;**

**}**

**public String getLocation() {**

**return location;**

**}**

**public void setLocation(String location) {**

**this.location = location;**

**}**

**public String getDescription() {**

**return description;**

**}**

**public void setDescription(String description) {**

**this.description = description;**

**}**

**public User getUser() {**

**return user;**

**}**

**public void setUser(User user) {**

**this.user = user;**

**}**

**public boolean isConfirmed() {**

**return confirmed;**

**}**

**public void setConfirmed(boolean confirmed) {**

**this.confirmed = confirmed;**

**}**

**@Override**

**public String toString() {**

**return "Appointment{" +**

**"id=" + id +**

**", date=" + date +**

**", location='" + location + '\'' +**

**", description='" + description + '\'' +**

**", user=" + user +**

**'}';**

**}**

**}**

1. **PrimaryAccount.java:**

package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import java.math.BigDecimal;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.49

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class PrimaryAccount {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private int accountNumber;

private BigDecimal accountBalance;

@OneToMany(mappedBy = "primaryAccount", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

@JsonIgnore

private List<PrimaryTransaction> primaryTransactionList;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public int getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(int accountNumber) {

this.accountNumber = accountNumber;

}

public BigDecimal getAccountBalance() {

return accountBalance;

}

public void setAccountBalance(BigDecimal accountBalance) {

this.accountBalance = accountBalance;

}

public List<PrimaryTransaction> getPrimaryTransactionList() {

return primaryTransactionList;

}

public void setPrimaryTransactionList(List<PrimaryTransaction> primaryTransactionList) {

this.primaryTransactionList = primaryTransactionList;

}

}

1. **PrimaryTransaction.java:**

package com.bank.onlinebanking.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import java.math.BigDecimal;

import java.util.Date;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.49

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class PrimaryTransaction {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private Date date;

private String description;

private String type;

private String status;

private double amount;

private BigDecimal availableBalance;

@ManyToOne

@JoinColumn(name = "primary\_account\_id")

private PrimaryAccount primaryAccount;

public PrimaryTransaction() {

}

public PrimaryTransaction(Date date, String description, String type, String status, double amount, BigDecimal availableBalance, PrimaryAccount primaryAccount) {

this.date = date;

this.description = description;

this.type = type;

this.status = status;

this.amount = amount;

this.availableBalance = availableBalance;

this.primaryAccount = primaryAccount;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public double getAmount() {

return amount;

}

public void setAmount(double amount) {

this.amount = amount;

}

public BigDecimal getAvailableBalance() {

return availableBalance;

}

public void setAvailableBalance(BigDecimal availableBalance) {

this.availableBalance = availableBalance;

}

public PrimaryAccount getPrimaryAccount() {

return primaryAccount;

}

public void setPrimaryAccount(PrimaryAccount primaryAccount) {

this.primaryAccount = primaryAccount;

}

}

1. **Recipient.java:**

package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.54

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class Recipient {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

private String phone;

private String accountNumber;

private String description;

@ManyToOne

@JoinColumn(name = "user\_id")

@JsonIgnore

private User user;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public String getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(String accountNumber) {

this.accountNumber = accountNumber;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

}

1. **SavingsAccount.java:**

package com.bank.onlinebanking.entity;

import com.fasterxml.jackson.annotation.JsonIgnore;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import java.math.BigDecimal;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.51

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class SavingsAccount {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private int accountNumber;

private BigDecimal accountBalance;

@OneToMany(mappedBy = "savingsAccount", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

@JsonIgnore

private List<SavingsTransaction> savingsTransactionList;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public int getAccountNumber() {

return accountNumber;

}

public void setAccountNumber(int accountNumber) {

this.accountNumber = accountNumber;

}

public BigDecimal getAccountBalance() {

return accountBalance;

}

public void setAccountBalance(BigDecimal accountBalance) {

this.accountBalance = accountBalance;

}

public List<SavingsTransaction> getSavingsTransactionList() {

return savingsTransactionList;

}

public void setSavingsTransactionList(List<SavingsTransaction> savingsTransactionList) {

this.savingsTransactionList = savingsTransactionList;

}

}

1. **SavingsTransaction.java:**

package com.bank.onlinebanking.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import java.math.BigDecimal;

import java.util.Date;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.52

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class SavingsTransaction {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private Date date;

private String description;

private String type;

private String status;

private double amount;

private BigDecimal availableBalance;

@ManyToOne

@JoinColumn(name = "savings\_account\_id")

private SavingsAccount savingsAccount;

public SavingsTransaction() {

}

public SavingsTransaction(Date date, String description, String type, String status, double amount, BigDecimal availableBalance, SavingsAccount savingsAccount) {

this.date = date;

this.description = description;

this.type = type;

this.status = status;

this.amount = amount;

this.availableBalance = availableBalance;

this.savingsAccount = savingsAccount;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public Date getDate() {

return date;

}

public void setDate(Date date) {

this.date = date;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getStatus() {

return status;

}

public void setStatus(String status) {

this.status = status;

}

public double getAmount() {

return amount;

}

public void setAmount(double amount) {

this.amount = amount;

}

public BigDecimal getAvailableBalance() {

return availableBalance;

}

public void setAvailableBalance(BigDecimal availableBalance) {

this.availableBalance = availableBalance;

}

public SavingsAccount getSavingsAccount() {

return savingsAccount;

}

public void setSavingsAccount(SavingsAccount savingsAccount) {

this.savingsAccount = savingsAccount;

}

}

1. **User.java**

package com.bank.onlinebanking.entity;

import com.bank.onlinebanking.security.Authority;

import com.bank.onlinebanking.security.UserRole;

import com.fasterxml.jackson.annotation.JsonIgnore;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.userdetails.UserDetails;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.OneToOne;

import java.util.Collection;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.47

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class User implements UserDetails {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "userId", nullable = false, updatable = false)

private Long userId;

private String username;

private String password;

private String firstName;

private String lastName;

@Column(name = "email", nullable = false, unique = true)

private String email;

private String phone;

private boolean enabled = true;

@OneToOne

private PrimaryAccount primaryAccount;

@OneToOne

private SavingsAccount savingsAccount;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

@JsonIgnore

private List<Appointment> appointmentList;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

private List<Recipient> recipientList;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, fetch = FetchType.EAGER)

@JsonIgnore

private Set<UserRole> userRoles = new HashSet<>();

public Set<UserRole> getUserRoles() {

return userRoles;

}

public void setUserRoles(Set<UserRole> userRoles) {

this.userRoles = userRoles;

}

public Long getUserId() {

return userId;

}

public void setUserId(Long userId) {

this.userId = userId;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

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 getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPhone() {

return phone;

}

public void setPhone(String phone) {

this.phone = phone;

}

public List<Appointment> getAppointmentList() {

return appointmentList;

}

public void setAppointmentList(List<Appointment> appointmentList) {

this.appointmentList = appointmentList;

}

public List<Recipient> getRecipientList() {

return recipientList;

}

public void setRecipientList(List<Recipient> recipientList) {

this.recipientList = recipientList;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public PrimaryAccount getPrimaryAccount() {

return primaryAccount;

}

public void setPrimaryAccount(PrimaryAccount primaryAccount) {

this.primaryAccount = primaryAccount;

}

public SavingsAccount getSavingsAccount() {

return savingsAccount;

}

public void setSavingsAccount(SavingsAccount savingsAccount) {

this.savingsAccount = savingsAccount;

}

@Override

public String toString() {

return "User{" +

"userId=" + userId +

", username='" + username + '\'' +

", password='" + password + '\'' +

", firstName='" + firstName + '\'' +

", lastName='" + lastName + '\'' +

", email='" + email + '\'' +

", phone='" + phone + '\'' +

", appointmentList=" + appointmentList +

", recipientList=" + recipientList +

", userRoles=" + userRoles +

'}';

}

@Override

public Collection<? extends GrantedAuthority> getAuthorities() {

Set<GrantedAuthority> authorities = new HashSet<>();

userRoles.forEach(ur -> authorities.add(new Authority(ur.getRole().getName())));

return authorities;

}

@Override

public boolean isAccountNonExpired() {

// TODO Auto-generated method stub

return true;

}

@Override

public boolean isAccountNonLocked() {

// TODO Auto-generated method stub

return true;

}

@Override

public boolean isCredentialsNonExpired() {

// TODO Auto-generated method stub

return true;

}

@Override

public boolean isEnabled() {

return enabled;

}

public void setEnabled(boolean enabled) {

this.enabled = enabled;

}

}

1. **AppointmentResource.java:**

package com.example.datamodel.repository;

import com.example.datamodel.entity.User;

import com.example.datamodel.entity.UserOrder;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.stereotype.Repository;

import javax.persistence.EntityManager;

import javax.persistence.Query;

import javax.persistence.TypedQuery;

import javax.transaction.Transactional;

import java.util.List;

import java.util.Queue;

@Repository

@Transactional

public class UserRepo {

@Autowired

EntityManager em;

private Logger log = LoggerFactory.getLogger(this.getClass());

/\*

\* find user by user id

\* using entity manager's find method

\* return type is user object\*/

public User findUserById(Long userId) {

User user = em.find(User.class, userId);

log.info("user object returned -> {}", user);

return user;

}

public User userUpdate(Long userId, String oldPassword, String newPassword) {

User user = em.find(User.class, userId);

if (user.getPassword() == oldPassword) {

user.setPassword(newPassword);

if (user.getPassword() == newPassword) {

return user;

} else {

return null;

}

} else {

return null;

}

}

public User addNewUser(User user) {

em.persist(user);

return user;

}

public List<User> getUsers() {

TypedQuery<User> query = em.createNamedQuery("get\_all\_users",User.class);

List<User> result = query.getResultList();

return result;

}

public User verifyUser(String email, String password) {

User user = null;

List<User> users = getUsers();

User newuser = null;

for (int i = 0; i < users.size(); i++) {

user = users.get(i);

if (user.getEmail().equals(email)) {

if (user.getPassword().equals(password)) {

newuser = user;

break;

}

}

}

log.info("user -> {}", user);

return newuser;

}

public List<User> searchUserByName(String name) {

Query query = em.createNativeQuery("Select \* from user where name = :name",User.class);

query.setParameter("name",name);

List result = query.getResultList();

return result;

}

public void addUserOrderToUser(Long id, UserOrder userOrder) {

User user = em.find(User.class,id);

user.setUserOrder(userOrder);

em.merge(user);

}

public UserOrder verifyUserToOrder(Long userId) {

User user = em.find(User.class,userId);

return user.getUserOrder();

}

}

1. **UserResource.css:**

package com.bank.onlinebanking.resource;

import com.bank.onlinebanking.entity.PrimaryTransaction;

import com.bank.onlinebanking.entity.SavingsTransaction;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.service.TransactionService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.39

\* To change this template use File | Settings | File Templates.

\*/

@RestController

@RequestMapping("/api")

@PreAuthorize("hasRole('ADMIN')")

public class UserResource {

@Autowired

private UserService userService;

@Autowired

private TransactionService transactionService;

@RequestMapping(value = "/user/all", method = RequestMethod.GET)

public List<User> userList() {

return userService.findUserList();

}

@RequestMapping(value = "/user/primary/transaction", method = RequestMethod.GET)

public List<PrimaryTransaction> getPrimaryTransactionList(@RequestParam("username") String username) {

return transactionService.findPrimaryTransactionList(username);

}

@RequestMapping(value = "/user/savings/transaction", method = RequestMethod.GET)

public List<SavingsTransaction> getSavingsTransactionList(@RequestParam("username") String username) {

return transactionService.findSavingsTransactionList(username);

}

@RequestMapping("/user/{username}/enable")

public void enableUser(@PathVariable("username") String username) {

userService.enableUser(username);

}

@RequestMapping("/user/{username}/disable")

public void diableUser(@PathVariable("username") String username) {

userService.disableUser(username);

}

}

1. **role.java**

package com.bank.onlinebanking.security;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import java.util.HashSet;

import java.util.Set;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 06/08/18

\* Time: 20.45

\* To change this template use File | Settings | File Templates.

\*/

@Entity

public class Role {

@Id

private int roleId;

private String name;

@OneToMany(mappedBy = "role", cascade = CascadeType.ALL, fetch = FetchType.LAZY)

private Set<UserRole> userRoles = new HashSet<>();

public Role() {

}

public int getRoleId() {

return roleId;

}

public void setRoleId(int roleId) {

this.roleId = roleId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Set<UserRole> getUserRoles() {

return userRoles;

}

public void setUserRoles(Set<UserRole> userRoles) {

this.userRoles = userRoles;

}

}

1. **userrole.java**

package com.bank.onlinebanking.security;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

import com.bank.onlinebanking.entity.User;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 06/08/18

\* Time: 20.44

\* To change this template use File | Settings | File Templates.

\*/

@Entity

@Table(name = "user\_role")

public class UserRole {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private long userRoleId;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "user\_id")

private User user;

@ManyToOne(fetch = FetchType.EAGER)

@JoinColumn(name = "role\_id")

private Role role;

public UserRole(User user, Role role) {

this.user = user;

this.role = role;

}

public UserRole() {

}

public long getUserRoleId() {

return userRoleId;

}

public void setUserRoleId(long userRoleId) {

this.userRoleId = userRoleId;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

public Role getRole() {

return role;

}

public void setRole(Role role) {

this.role = role;

}

}

1. **AccountServiceImpl.java**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.PrimaryAccount;

import com.bank.onlinebanking.entity.PrimaryTransaction;

import com.bank.onlinebanking.entity.SavingsAccount;

import com.bank.onlinebanking.entity.SavingsTransaction;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.PrimaryAccountDao;

import com.bank.onlinebanking.repository.SavingsAccountDao;

import com.bank.onlinebanking.service.AccountService;

import com.bank.onlinebanking.service.TransactionService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Service;

import java.math.BigDecimal;

import java.security.Principal;

import java.util.Date;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 09/08/18

\* Time: 04.36

\* To change this template use File | Settings | File Templates.

\*/

@Service

public class AccountServiceImpl implements AccountService {

private static int nextAccountNumber = 11223101;

@Autowired

private PrimaryAccountDao primaryAccountDao;

@Autowired

private SavingsAccountDao savingsAccountDao;

@Autowired

private UserService userService;

@Autowired

private TransactionService transactionService;

public PrimaryAccount createPrimaryAccount() {

PrimaryAccount primaryAccount = new PrimaryAccount();

primaryAccount.setAccountBalance(new BigDecimal(0.0));

primaryAccount.setAccountNumber(accountGen());

primaryAccountDao.save(primaryAccount);

return primaryAccountDao.findByAccountNumber(primaryAccount.getAccountNumber());

}

public SavingsAccount createSavingsAccount() {

SavingsAccount savingsAccount = new SavingsAccount();

savingsAccount.setAccountBalance(new BigDecimal(0.0));

savingsAccount.setAccountNumber(accountGen());

savingsAccountDao.save(savingsAccount);

return savingsAccountDao.findByAccountNumber(savingsAccount.getAccountNumber());

}

public void deposit(String accountType, double amount, Principal principal) {

User user = userService.findByUsername(principal.getName());

if (accountType.equalsIgnoreCase("Primary")) {

PrimaryAccount primaryAccount = user.getPrimaryAccount();

primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().add(new BigDecimal(amount)));

primaryAccountDao.save(primaryAccount);

Date date = new Date();

PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Deposit to Primary Account", "Account", "Finished", amount, primaryAccount.getAccountBalance(), primaryAccount);

transactionService.savePrimaryDepositTransaction(primaryTransaction);

} else if (accountType.equalsIgnoreCase("Savings")) {

SavingsAccount savingsAccount = user.getSavingsAccount();

savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().add(new BigDecimal(amount)));

savingsAccountDao.save(savingsAccount);

Date date = new Date();

SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Deposit to savings Account", "Account", "Finished", amount, savingsAccount.getAccountBalance(), savingsAccount);

transactionService.saveSavingsDepositTransaction(savingsTransaction);

}

}

public void withdraw(String accountType, double amount, Principal principal) {

User user = userService.findByUsername(principal.getName());

if (accountType.equalsIgnoreCase("Primary")) {

PrimaryAccount primaryAccount = user.getPrimaryAccount();

primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new BigDecimal(amount)));

primaryAccountDao.save(primaryAccount);

Date date = new Date();

PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Withdraw from Primary Account", "Account", "Finished", amount, primaryAccount.getAccountBalance(), primaryAccount);

transactionService.savePrimaryWithdrawTransaction(primaryTransaction);

} else if (accountType.equalsIgnoreCase("Savings")) {

SavingsAccount savingsAccount = user.getSavingsAccount();

savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new BigDecimal(amount)));

savingsAccountDao.save(savingsAccount);

Date date = new Date();

SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Withdraw from savings Account", "Account", "Finished", amount, savingsAccount.getAccountBalance(), savingsAccount);

transactionService.saveSavingsWithdrawTransaction(savingsTransaction);

}

}

private int accountGen() {

return ++nextAccountNumber;

}

}

1. **AppointmentServiceImpl.java:**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.Appointment;

import com.bank.onlinebanking.repository.AppointmentDao;

import com.bank.onlinebanking.service.AppointmentService;

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

import org.springframework.stereotype.Service;

import java.util.List;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 09/08/18

\* Time: 04.38

\* To change this template use File | Settings | File Templates.

\*/

@Service

public class AppointmentServiceImpl implements AppointmentService {

@Autowired

private AppointmentDao appointmentDao;

public Appointment createAppointment(Appointment appointment) {

return appointmentDao.save(appointment);

}

public List<Appointment> findAll() {

return appointmentDao.findAll();

}

public Appointment findAppointment(Long id) {

return appointmentDao.findById(id).get();

}

public void confirmAppointment(Long id) {

Appointment appointment = findAppointment(id);

appointment.setConfirmed(true);

appointmentDao.save(appointment);

}

}

1. **TransactionServiceImpl.java:**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.PrimaryAccount;

import com.bank.onlinebanking.entity.PrimaryTransaction;

import com.bank.onlinebanking.entity.Recipient;

import com.bank.onlinebanking.entity.SavingsAccount;

import com.bank.onlinebanking.entity.SavingsTransaction;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.PrimaryAccountDao;

import com.bank.onlinebanking.repository.PrimaryTransactionDao;

import com.bank.onlinebanking.repository.RecipientDao;

import com.bank.onlinebanking.repository.SavingsAccountDao;

import com.bank.onlinebanking.repository.SavingsTransactionDao;

import com.bank.onlinebanking.service.TransactionService;

import com.bank.onlinebanking.service.UserService;

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

import org.springframework.stereotype.Service;

import java.math.BigDecimal;

import java.security.Principal;

import java.util.Date;

import java.util.List;

import java.util.stream.Collectors;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 10/08/18

\* Time: 06.21

\* To change this template use File | Settings | File Templates.

\*/

@Service

public class TransactionServiceImpl implements TransactionService {

@Autowired

private UserService userService;

@Autowired

private PrimaryTransactionDao primaryTransactionDao;

@Autowired

private SavingsTransactionDao savingsTransactionDao;

@Autowired

private PrimaryAccountDao primaryAccountDao;

@Autowired

private SavingsAccountDao savingsAccountDao;

@Autowired

private RecipientDao recipientDao;

public List<PrimaryTransaction> findPrimaryTransactionList(String username) {

User user = userService.findByUsername(username);

List<PrimaryTransaction> primaryTransactionList = user.getPrimaryAccount().getPrimaryTransactionList();

return primaryTransactionList;

}

public List<SavingsTransaction> findSavingsTransactionList(String username) {

User user = userService.findByUsername(username);

List<SavingsTransaction> savingsTransactionList = user.getSavingsAccount().getSavingsTransactionList();

return savingsTransactionList;

}

public void savePrimaryDepositTransaction(PrimaryTransaction primaryTransaction) {

primaryTransactionDao.save(primaryTransaction);

}

public void saveSavingsDepositTransaction(SavingsTransaction savingsTransaction) {

savingsTransactionDao.save(savingsTransaction);

}

public void savePrimaryWithdrawTransaction(PrimaryTransaction primaryTransaction) {

primaryTransactionDao.save(primaryTransaction);

}

public void saveSavingsWithdrawTransaction(SavingsTransaction savingsTransaction) {

savingsTransactionDao.save(savingsTransaction);

}

public void betweenAccountsTransfer(String transferFrom, String transferTo, String amount, PrimaryAccount primaryAccount, SavingsAccount savingsAccount) throws Exception {

if (transferFrom.equalsIgnoreCase("Primary") && transferTo.equalsIgnoreCase("Savings")) {

primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new BigDecimal(amount)));

savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().add(new BigDecimal(amount)));

primaryAccountDao.save(primaryAccount);

savingsAccountDao.save(savingsAccount);

Date date = new Date();

PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Between account transfer from " + transferFrom + " to " + transferTo, "Account", "Finished", Double.parseDouble(amount), primaryAccount.getAccountBalance(), primaryAccount);

primaryTransactionDao.save(primaryTransaction);

} else if (transferFrom.equalsIgnoreCase("Savings") && transferTo.equalsIgnoreCase("Primary")) {

primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().add(new BigDecimal(amount)));

savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new BigDecimal(amount)));

primaryAccountDao.save(primaryAccount);

savingsAccountDao.save(savingsAccount);

Date date = new Date();

SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Between account transfer from " + transferFrom + " to " + transferTo, "Transfer", "Finished", Double.parseDouble(amount), savingsAccount.getAccountBalance(), savingsAccount);

savingsTransactionDao.save(savingsTransaction);

} else {

throw new Exception("Invalid Transfer");

}

}

public List<Recipient> findRecipientList(Principal principal) {

String username = principal.getName();

List<Recipient> recipientList = recipientDao.findAll().stream()

.filter(recipient -> username.equals(recipient.getUser().getUsername()))

.collect(Collectors.toList());

return recipientList;

}

public Recipient saveRecipient(Recipient recipient) {

return recipientDao.save(recipient);

}

public Recipient findRecipientByName(String recipientName) {

return recipientDao.findByName(recipientName);

}

public void deleteRecipientByName(String recipientName) {

recipientDao.deleteByName(recipientName);

}

public void toSomeoneElseTransfer(Recipient recipient, String accountType, String amount, PrimaryAccount primaryAccount, SavingsAccount savingsAccount) {

if (accountType.equalsIgnoreCase("Primary")) {

primaryAccount.setAccountBalance(primaryAccount.getAccountBalance().subtract(new BigDecimal(amount)));

primaryAccountDao.save(primaryAccount);

Date date = new Date();

PrimaryTransaction primaryTransaction = new PrimaryTransaction(date, "Transfer to recipient " + recipient.getName(), "Transfer", "Finished", Double.parseDouble(amount), primaryAccount.getAccountBalance(), primaryAccount);

primaryTransactionDao.save(primaryTransaction);

} else if (accountType.equalsIgnoreCase("Savings")) {

savingsAccount.setAccountBalance(savingsAccount.getAccountBalance().subtract(new BigDecimal(amount)));

savingsAccountDao.save(savingsAccount);

Date date = new Date();

SavingsTransaction savingsTransaction = new SavingsTransaction(date, "Transfer to recipient " + recipient.getName(), "Transfer", "Finished", Double.parseDouble(amount), savingsAccount.getAccountBalance(), savingsAccount);

savingsTransactionDao.save(savingsTransaction);

}

}

}

1. **UserSecurityService.java:**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.UserDao;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import org.springframework.stereotype.Service;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Time: 06.30

\* To change this template use File | Settings | File Templates.

\*/

@Service

public class UserSecurityService implements UserDetailsService {

private static final Logger LOG = LoggerFactory.getLogger(UserSecurityService.class);

@Autowired

private UserDao userDao;

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

User user = userDao.findByUsername(username);

if (null == user) {

LOG.warn("Username {} not found", username);

throw new UsernameNotFoundException("Username " + username + " not found");

}

return user;

}

}

1. **UserSecurityServiceImpl.java:**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.UserDao;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import org.springframework.stereotype.Service;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 10/08/18

\* Time: 06.22

\* To change this template use File | Settings | File Templates.

\*/

@Service

public class UserSecurityServiceImpl implements UserDetailsService {

private static final Logger LOG = LoggerFactory.getLogger(UserSecurityServiceImpl.class);

@Autowired

private UserDao userDao;

@Override

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

User user = userDao.findByUsername(username);

if (null == user) {

LOG.warn("Username {} not found", username);

throw new UsernameNotFoundException("Username " + username + " not found");

}

return user;

}

}

1. **UserServiceImpl.java**

package com.bank.onlinebanking.service.UserServiceImpl;

import com.bank.onlinebanking.entity.User;

import com.bank.onlinebanking.repository.RoleDao;

import com.bank.onlinebanking.repository.UserDao;

import com.bank.onlinebanking.security.UserRole;

import com.bank.onlinebanking.service.AccountService;

import com.bank.onlinebanking.service.UserService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import java.util.List;

import java.util.Set;

/\*\*

\* Created by IntelliJ IDEA.

\* Project : online-banking

\* Date: 10/08/18

\* Time: 06.23

\* To change this template use File | Settings | File Templates.

\*/

@Service

@Transactional

public class UserServiceImpl implements UserService {

private static final Logger LOG = LoggerFactory.getLogger(UserService.class);

@Autowired

private UserDao userDao;

@Autowired

private RoleDao roleDao;

@Autowired

private BCryptPasswordEncoder passwordEncoder;

@Autowired

private AccountService accountService;

public void save(User user) {

userDao.save(user);

}

public User findByUsername(String username) {

return userDao.findByUsername(username);

}

public User findByEmail(String email) {

return userDao.findByEmail(email);

}

public User createUser(User user, Set<UserRole> userRoles) {

User localUser = userDao.findByUsername(user.getUsername());

if (localUser != null) {

LOG.info("User with username {} already exist. Nothing will be done. ", user.getUsername());

} else {

String encryptedPassword = passwordEncoder.encode(user.getPassword());

user.setPassword(encryptedPassword);

for (UserRole ur : userRoles) {

roleDao.save(ur.getRole());

}

user.getUserRoles().addAll(userRoles);

user.setPrimaryAccount(accountService.createPrimaryAccount());

user.setSavingsAccount(accountService.createSavingsAccount());

localUser = userDao.save(user);

}

return localUser;

}

public boolean checkUserExists(String username, String email) {

return checkUsernameExists(username) || checkEmailExists(username);

}

public boolean checkUsernameExists(String username) {

return null != findByUsername(username);

}

public boolean checkEmailExists(String email) {

return null != findByEmail(email);

}

public User saveUser(User user) {

return userDao.save(user);

}

public List<User> findUserList() {

return userDao.findAll();

}

public void enableUser(String username) {

User user = findByUsername(username);

user.setEnabled(true);

userDao.save(user);

}

public void disableUser(String username) {

User user = findByUsername(username);

user.setEnabled(false);

LOG.info("user status : {}", user.isEnabled());

userDao.save(user);

LOG.info(username, "{}, is disabled.");

}

}

1. **bootstrap.css**

<!DOCTYPE html>

<html>

<head>

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

<title>Insert title here</title>

<link rel=*"preconnect"* href=*"https://fonts.googleapis.com"*>

<link rel=*"preconnect"* href=*"https://fonts.gstatic.com"* crossorigin>

<link href=*"https://fonts.googleapis.com/css2?family=Lato:wght@300&display=swap"* rel=*"stylesheet"*>

<link href=*"/css/SignUp.css"* rel=*"stylesheet"* type=*"text/css"* >

</head>

<body>

<div class=*"loginBox"*>

<div class=*"logo"*></div>

<div class=*"title"*>SignUp</div>

<form action=*"UserSignUp"* method=*"post"*>

<div class=*"fields"*>

<div class=*"username"*>

<svg class=*"svg-icon"* viewBox=*"0 0 20 20"*>

<path d=*"M12.075,10.812c1.358-0.853,2.242-2.507,2.242-4.037c0-2.181-1.795-4.618-4.198-4.618S5.921,4.594,5.921,6.775c0,1.53,0.884,3.185,2.242,4.037c-3.222,0.865-5.6,3.807-5.6,7.298c0,0.23,0.189,0.42,0.42,0.42h14.273c0.23,0,0.42-0.189,0.42-0.42C17.676,14.619,15.297,11.677,12.075,10.812 M6.761,6.775c0-2.162,1.773-3.778,3.358-3.778s3.359,1.616,3.359,3.778c0,2.162-1.774,3.778-3.359,3.778S6.761,8.937,6.761,6.775 M3.415,17.69c0.218-3.51,3.142-6.297,6.704-6.297c3.562,0,6.486,2.787,6.705,6.297H3.415z"*></path>

</svg>

<input class=*"user-input"* type=*"text"* placeholder=*"Choose Your UserName"* name=*"userName"* required>

</div>

<div class=*"username"*>

<svg class=*"svg-icon"* viewBox=*"0 0 20 20"*>

<path d=*"M12.075,10.812c1.358-0.853,2.242-2.507,2.242-4.037c0-2.181-1.795-4.618-4.198-4.618S5.921,4.594,5.921,6.775c0,1.53,0.884,3.185,2.242,4.037c-3.222,0.865-5.6,3.807-5.6,7.298c0,0.23,0.189,0.42,0.42,0.42h14.273c0.23,0,0.42-0.189,0.42-0.42C17.676,14.619,15.297,11.677,12.075,10.812 M6.761,6.775c0-2.162,1.773-3.778,3.358-3.778s3.359,1.616,3.359,3.778c0,2.162-1.774,3.778-3.359,3.778S6.761,8.937,6.761,6.775 M3.415,17.69c0.218-3.51,3.142-6.297,6.704-6.297c3.562,0,6.486,2.787,6.705,6.297H3.415z"*></path>

</svg>

<input class=*"user-input"* type=*"email"* placeholder=*"Enter Your email"* name=*"email"* required>

</div>

<div class=*"password"*>

<svg class=*"svg-icon"* viewBox=*"0 0 20 20"*>

<path d=*"M17.308,7.564h-1.993c0-2.929-2.385-5.314-5.314-5.314S4.686,4.635,4.686,7.564H2.693c-0.244,0-0.443,0.2-0.443,0.443v9.3c0,0.243,0.199,0.442,0.443,0.442h14.615c0.243,0,0.442-0.199,0.442-0.442v-9.3C17.75,7.764,17.551,7.564,17.308,7.564 M10,3.136c2.442,0,4.43,1.986,4.43,4.428H5.571C5.571,5.122,7.558,3.136,10,3.136 M16.865,16.864H3.136V8.45h13.729V16.864z M10,10.664c-0.854,0-1.55,0.696-1.55,1.551c0,0.699,0.467,1.292,1.107,1.485v0.95c0,0.243,0.2,0.442,0.443,0.442s0.443-0.199,0.443-0.442V13.7c0.64-0.193,1.106-0.786,1.106-1.485C11.55,11.36,10.854,10.664,10,10.664 M10,12.878c-0.366,0-0.664-0.298-0.664-0.663c0-0.366,0.298-0.665,0.664-0.665c0.365,0,0.664,0.299,0.664,0.665C10.664,12.58,10.365,12.878,10,12.878"*></path>

</svg>

<input class=*"user-input"* type=*"text"* placeholder=*"Enter Your name"* name=*"name"* required>

</div>

<div class=*"password"*>

<svg class=*"svg-icon"* viewBox=*"0 0 20 20"*>

<path d=*"M17.308,7.564h-1.993c0-2.929-2.385-5.314-5.314-5.314S4.686,4.635,4.686,7.564H2.693c-0.244,0-0.443,0.2-0.443,0.443v9.3c0,0.243,0.199,0.442,0.443,0.442h14.615c0.243,0,0.442-0.199,0.442-0.442v-9.3C17.75,7.764,17.551,7.564,17.308,7.564 M10,3.136c2.442,0,4.43,1.986,4.43,4.428H5.571C5.571,5.122,7.558,3.136,10,3.136 M16.865,16.864H3.136V8.45h13.729V16.864z M10,10.664c-0.854,0-1.55,0.696-1.55,1.551c0,0.699,0.467,1.292,1.107,1.485v0.95c0,0.243,0.2,0.442,0.443,0.442s0.443-0.199,0.443-0.442V13.7c0.64-0.193,1.106-0.786,1.106-1.485C11.55,11.36,10.854,10.664,10,10.664 M10,12.878c-0.366,0-0.664-0.298-0.664-0.663c0-0.366,0.298-0.665,0.664-0.665c0.365,0,0.664,0.299,0.664,0.665C10.664,12.58,10.365,12.878,10,12.878"*></path>

</svg>

<input class=*"user-input"* type=*"password"* placeholder=*"Enter Your password"* name=*"password"* required>

</div>

<div class=*"signin-button"*>

<input class=*"user-input"* type=*"submit"* value=*"Sign Up"*>

</div>

</div>

</form>

<a href=*""*>Login</a>

</div>

</body>

</html>

1. **datepicker.css**

/\*!

\* Datetimepicker for Bootstrap

\*

\* Copyright 2012 Stefan Petre

\* Improvements by Andrew Rowls

\* Licensed under the Apache License v2.0

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\*/

.datetimepicker {

padding: 4px;

margin-top: 1px;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

border-radius: 4px;

direction: ltr;

}

.datetimepicker-inline {

width: 220px;

}

.datetimepicker.datetimepicker-rtl {

direction: rtl;

}

.datetimepicker.datetimepicker-rtl table tr td span {

float: right;

}

.datetimepicker-dropdown, .datetimepicker-dropdown-left {

top: 0;

left: 0;

}

[class\*=" datetimepicker-dropdown"]:before {

content: '';

display: inline-block;

border-left: 7px solid transparent;

border-right: 7px solid transparent;

border-bottom: 7px solid #cccccc;

border-bottom-color: rgba(0, 0, 0, 0.2);

position: absolute;

}

[class\*=" datetimepicker-dropdown"]:after {

content: '';

display: inline-block;

border-left: 6px solid transparent;

border-right: 6px solid transparent;

border-bottom: 6px solid #ffffff;

position: absolute;

}

[class\*=" datetimepicker-dropdown-top"]:before {

content: '';

display: inline-block;

border-left: 7px solid transparent;

border-right: 7px solid transparent;

border-top: 7px solid #cccccc;

border-top-color: rgba(0, 0, 0, 0.2);

border-bottom: 0;

}

[class\*=" datetimepicker-dropdown-top"]:after {

content: '';

display: inline-block;

border-left: 6px solid transparent;

border-right: 6px solid transparent;

border-top: 6px solid #ffffff;

border-bottom: 0;

}

.datetimepicker-dropdown-bottom-left:before {

top: -7px;

right: 6px;

}

.datetimepicker-dropdown-bottom-left:after {

top: -6px;

right: 7px;

}

.datetimepicker-dropdown-bottom-right:before {

top: -7px;

left: 6px;

}

.datetimepicker-dropdown-bottom-right:after {

top: -6px;

left: 7px;

}

.datetimepicker-dropdown-top-left:before {

bottom: -7px;

right: 6px;

}

.datetimepicker-dropdown-top-left:after {

bottom: -6px;

right: 7px;

}

.datetimepicker-dropdown-top-right:before {

bottom: -7px;

left: 6px;

}

.datetimepicker-dropdown-top-right:after {

bottom: -6px;

left: 7px;

}

.datetimepicker > div {

display: none;

}

.datetimepicker.minutes div.datetimepicker-minutes {

display: block;

}

.datetimepicker.hours div.datetimepicker-hours {

display: block;

}

.datetimepicker.days div.datetimepicker-days {

display: block;

}

.datetimepicker.months div.datetimepicker-months {

display: block;

}

.datetimepicker.years div.datetimepicker-years {

display: block;

}

.datetimepicker table {

margin: 0;

}

.datetimepicker td,

.datetimepicker th {

text-align: center;

width: 20px;

height: 20px;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

border-radius: 4px;

border: none;

}

.table-striped .datetimepicker table tr td,

.table-striped .datetimepicker table tr th {

background-color: transparent;

}

.datetimepicker table tr td.minute:hover {

background: #eeeeee;

cursor: pointer;

}

.datetimepicker table tr td.hour:hover {

background: #eeeeee;

cursor: pointer;

}

.datetimepicker table tr td.day:hover {

background: #eeeeee;

cursor: pointer;

}

.datetimepicker table tr td.old,

.datetimepicker table tr td.new {

color: #999999;

}

.datetimepicker table tr td.disabled,

.datetimepicker table tr td.disabled:hover {

background: none;

color: #999999;

cursor: default;

}

.datetimepicker table tr td.today,

.datetimepicker table tr td.today:hover,

.datetimepicker table tr td.today.disabled,

.datetimepicker table tr td.today.disabled:hover {

background-color: #fde19a;

background-image: -moz-linear-gradient(top, #fdd49a, #fdf59a);

background-image: -ms-linear-gradient(top, #fdd49a, #fdf59a);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#fdd49a), to(#fdf59a));

background-image: -webkit-linear-gradient(top, #fdd49a, #fdf59a);

background-image: -o-linear-gradient(top, #fdd49a, #fdf59a);

background-image: linear-gradient(top, #fdd49a, #fdf59a);

background-repeat: repeat-x;

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#fdd49a', endColorstr='#fdf59a', GradientType=0);

border-color: #fdf59a #fdf59a #fbed50;

border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);

filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);

}

.datetimepicker table tr td.today:hover,

.datetimepicker table tr td.today:hover:hover,

.datetimepicker table tr td.today.disabled:hover,

.datetimepicker table tr td.today.disabled:hover:hover,

.datetimepicker table tr td.today:active,

.datetimepicker table tr td.today:hover:active,

.datetimepicker table tr td.today.disabled:active,

.datetimepicker table tr td.today.disabled:hover:active,

.datetimepicker table tr td.today.active,

.datetimepicker table tr td.today:hover.active,

.datetimepicker table tr td.today.disabled.active,

.datetimepicker table tr td.today.disabled:hover.active,

.datetimepicker table tr td.today.disabled,

.datetimepicker table tr td.today:hover.disabled,

.datetimepicker table tr td.today.disabled.disabled,

.datetimepicker table tr td.today.disabled:hover.disabled,

.datetimepicker table tr td.today[disabled],

.datetimepicker table tr td.today:hover[disabled],

.datetimepicker table tr td.today.disabled[disabled],

.datetimepicker table tr td.today.disabled:hover[disabled] {

background-color: #fdf59a;

}

.datetimepicker table tr td.today:active,

.datetimepicker table tr td.today:hover:active,

.datetimepicker table tr td.today.disabled:active,

.datetimepicker table tr td.today.disabled:hover:active,

.datetimepicker table tr td.today.active,

.datetimepicker table tr td.today:hover.active,

.datetimepicker table tr td.today.disabled.active,

.datetimepicker table tr td.today.disabled:hover.active {

background-color: #fbf069;

}

.datetimepicker table tr td.active,

.datetimepicker table tr td.active:hover,

.datetimepicker table tr td.active.disabled,

.datetimepicker table tr td.active.disabled:hover {

background-color: #006dcc;

background-image: -moz-linear-gradient(top, #0088cc, #0044cc);

background-image: -ms-linear-gradient(top, #0088cc, #0044cc);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#0088cc), to(#0044cc));

background-image: -webkit-linear-gradient(top, #0088cc, #0044cc);

background-image: -o-linear-gradient(top, #0088cc, #0044cc);

background-image: linear-gradient(top, #0088cc, #0044cc);

background-repeat: repeat-x;

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#0088cc', endColorstr='#0044cc', GradientType=0);

border-color: #0044cc #0044cc #002a80;

border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);

filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);

color: #ffffff;

text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);

}

.datetimepicker table tr td.active:hover,

.datetimepicker table tr td.active:hover:hover,

.datetimepicker table tr td.active.disabled:hover,

.datetimepicker table tr td.active.disabled:hover:hover,

.datetimepicker table tr td.active:active,

.datetimepicker table tr td.active:hover:active,

.datetimepicker table tr td.active.disabled:active,

.datetimepicker table tr td.active.disabled:hover:active,

.datetimepicker table tr td.active.active,

.datetimepicker table tr td.active:hover.active,

.datetimepicker table tr td.active.disabled.active,

.datetimepicker table tr td.active.disabled:hover.active,

.datetimepicker table tr td.active.disabled,

.datetimepicker table tr td.active:hover.disabled,

.datetimepicker table tr td.active.disabled.disabled,

.datetimepicker table tr td.active.disabled:hover.disabled,

.datetimepicker table tr td.active[disabled],

.datetimepicker table tr td.active:hover[disabled],

.datetimepicker table tr td.active.disabled[disabled],

.datetimepicker table tr td.active.disabled:hover[disabled] {

background-color: #0044cc;

}

.datetimepicker table tr td.active:active,

.datetimepicker table tr td.active:hover:active,

.datetimepicker table tr td.active.disabled:active,

.datetimepicker table tr td.active.disabled:hover:active,

.datetimepicker table tr td.active.active,

.datetimepicker table tr td.active:hover.active,

.datetimepicker table tr td.active.disabled.active,

.datetimepicker table tr td.active.disabled:hover.active {

background-color: #003399;

}

.datetimepicker table tr td span {

display: block;

width: 23%;

height: 54px;

line-height: 54px;

float: left;

margin: 1%;

cursor: pointer;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

border-radius: 4px;

}

.datetimepicker .datetimepicker-hours span {

height: 26px;

line-height: 26px;

}

.datetimepicker .datetimepicker-hours table tr td span.hour\_am,

.datetimepicker .datetimepicker-hours table tr td span.hour\_pm {

width: 14.6%;

}

.datetimepicker .datetimepicker-hours fieldset legend,

.datetimepicker .datetimepicker-minutes fieldset legend {

margin-bottom: inherit;

line-height: 30px;

}

.datetimepicker .datetimepicker-minutes span {

height: 26px;

line-height: 26px;

}

.datetimepicker table tr td span:hover {

background: #eeeeee;

}

.datetimepicker table tr td span.disabled,

.datetimepicker table tr td span.disabled:hover {

background: none;

color: #999999;

cursor: default;

}

.datetimepicker table tr td span.active,

.datetimepicker table tr td span.active:hover,

.datetimepicker table tr td span.active.disabled,

.datetimepicker table tr td span.active.disabled:hover {

background-color: #006dcc;

background-image: -moz-linear-gradient(top, #0088cc, #0044cc);

background-image: -ms-linear-gradient(top, #0088cc, #0044cc);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#0088cc), to(#0044cc));

background-image: -webkit-linear-gradient(top, #0088cc, #0044cc);

background-image: -o-linear-gradient(top, #0088cc, #0044cc);

background-image: linear-gradient(top, #0088cc, #0044cc);

background-repeat: repeat-x;

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#0088cc', endColorstr='#0044cc', GradientType=0);

border-color: #0044cc #0044cc #002a80;

border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);

filter: progid:DXImageTransform.Microsoft.gradient(enabled=false);

color: #ffffff;

text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);

}

.datetimepicker table tr td span.active:hover,

.datetimepicker table tr td span.active:hover:hover,

.datetimepicker table tr td span.active.disabled:hover,

.datetimepicker table tr td span.active.disabled:hover:hover,

.datetimepicker table tr td span.active:active,

.datetimepicker table tr td span.active:hover:active,

.datetimepicker table tr td span.active.disabled:active,

.datetimepicker table tr td span.active.disabled:hover:active,

.datetimepicker table tr td span.active.active,

.datetimepicker table tr td span.active:hover.active,

.datetimepicker table tr td span.active.disabled.active,

.datetimepicker table tr td span.active.disabled:hover.active,

.datetimepicker table tr td span.active.disabled,

.datetimepicker table tr td span.active:hover.disabled,

.datetimepicker table tr td span.active.disabled.disabled,

.datetimepicker table tr td span.active.disabled:hover.disabled,

.datetimepicker table tr td span.active[disabled],

.datetimepicker table tr td span.active:hover[disabled],

.datetimepicker table tr td span.active.disabled[disabled],

.datetimepicker table tr td span.active.disabled:hover[disabled] {

background-color: #0044cc;

}

.datetimepicker table tr td span.active:active,

.datetimepicker table tr td span.active:hover:active,

.datetimepicker table tr td span.active.disabled:active,

.datetimepicker table tr td span.active.disabled:hover:active,

.datetimepicker table tr td span.active.active,

.datetimepicker table tr td span.active:hover.active,

.datetimepicker table tr td span.active.disabled.active,

.datetimepicker table tr td span.active.disabled:hover.active {

background-color: #003399;

}

.datetimepicker table tr td span.old {

color: #999999;

}

.datetimepicker th.switch {

width: 145px;

}

.datetimepicker th span.glyphicon {

pointer-events: none;

}

.datetimepicker thead tr:first-child th,

.datetimepicker tfoot th {

cursor: pointer;

}

.datetimepicker thead tr:first-child th:hover,

.datetimepicker tfoot th:hover {

background: #eeeeee;

}

.input-append.date .add-on i,

.input-prepend.date .add-on i,

.input-group.date .input-group-addon span {

cursor: pointer;

width: 14px;

height: 14px;

}

1. **main.css**

body {

background-color: transparent;

}

html {

background: url("/images/111sbg.png") no-repeat center center fixed;

-webkit-background-size: cover;

-moz-background-size: cover;

-o-background-size: cover;

background-size: cover;

}

img {

display: block;

margin-left: auto;

margin-right: auto;

}

.form-basic {

max-width: 640px;

margin: 0 auto;

padding: 55px;

box-sizing: border-box;

background-color: #ffffff;

box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.1);

font: bold 14px sans-serif;

text-align: center;

}

.form-basic .form-row {

text-align: left;

margin-bottom: 22px;

}

.form-basic .form-title-row {

text-align: center;

margin-bottom: 55px;

}

/\* The form title \*/

.form-basic h1 {

display: inline-block;

box-sizing: border-box;

color: #4c565e;

font-size: 24px;

padding: 0 10px 15px;

border-bottom: 2px solid #6caee0;

margin: 0;

}

.form-basic .form-row > label span {

display: inline-block;

box-sizing: border-box;

color: #5F5F5F;

width: 180px;

text-align: right;

vertical-align: top;

padding: 12px 25px;

}

.form-basic input {

color: #5f5f5f;

box-sizing: border-box;

width: 240px;

box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);

padding: 12px;

border: 1px solid #dbdbdb;

}

.form-basic input[type=radio], .form-basic input[type=checkbox] {

box-shadow: none;

width: auto;

}

.form-basic input[type=checkbox] {

margin-top: 13px;

}

.form-basic select {

background-color: #ffffff;

color: #5f5f5f;

box-sizing: border-box;

max-width: 240px;

box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);

padding: 12px 8px;

border: 1px solid #dbdbdb;

}

.form-basic textarea {

color: #5f5f5f;

box-sizing: border-box;

width: 240px;

height: 80px;

box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);

font: normal 13px sans-serif;

padding: 12px;

border: 1px solid #dbdbdb;

resize: vertical;

}

.form-basic .form-radio-buttons {

display: inline-block;

vertical-align: top;

}

.form-basic .form-radio-buttons > div {

margin-top: 10px;

}

.form-basic .form-radio-buttons label span {

margin-left: 8px;

color: #5f5f5f;

font-weight: normal;

}

.form-basic .form-radio-buttons input {

width: auto;

}

.form-basic button {

display: block;

border-radius: 2px;

background-color: #6caee0;

color: #ffffff;

font-weight: bold;

box-shadow: 1px 2px 4px 0 rgba(0, 0, 0, 0.08);

padding: 14px 22px;

border: 0;

margin: 40px 183px 0;

}

/\* Making the form responsive. Remove this media query

if you don't need the form to work on mobile devices. \*/

@media ( max-width: 600px) {

.form-basic {

padding: 30px;

max-width: 480px;

}

.form-basic .form-row {

max-width: 300px;

margin: 25px auto;

text-align: left;

}

.form-basic .form-title-row {

margin-bottom: 50px;

}

.form-basic .form-row > label span {

display: block;

text-align: left;

padding: 0 0 15px;

}

.form-basic select {

width: 240px;

}

.form-basic input[type=checkbox] {

margin-top: 0;

}

.form-basic .form-radio-buttons > div {

margin: 0 0 10px;

}

.form-basic button {

margin: 0;

}

}

.main {

margin-top: 70px;

}

.main-center {

margin-top: 30px;

margin: 0 auto;

max-width: 330px;

padding: 40px 40px;

}

.responstable {

margin: 1em 0;

width: 100%;

overflow: hidden;

background: #FFF;

color: #024457;

border-radius: 10px;

border: 1px solid #167F92;

}

.responstable tr {

border: 1px solid #D9E4E6;

}

.responstable tr:nth-child(odd) {

background-color: #EAF3F3;

}

.responstable th {

display: none;

border: 1px solid #FFF;

background-color: #167F92;

color: #FFF;

padding: 1em;

}

.responstable th:first-child {

display: table-cell;

text-align: center;

}

.responstable th:nth-child(2) {

display: table-cell;

}

.responstable th:nth-child(2) span {

display: none;

}

.responstable th:nth-child(2):after {

content: attr(data-th);

}

@media ( min-width: 480px) {

.responstable th:nth-child(2) span {

display: block;

}

.responstable th:nth-child(2):after {

display: none;

}

}

.responstable td {

display: block;

word-wrap: break-word;

max-width: 7em;

}

.responstable td:first-child {

display: table-cell;

text-align: center;

border-right: 1px solid #D9E4E6;

}

@media ( min-width: 480px) {

.responstable td {

border: 1px solid #D9E4E6;

}

}

.responstable th, .responstable td {

text-align: left;

margin: .5em 1em;

}

@media ( min-width: 480px) {

.responstable th, .responstable td {

display: table-cell;

padding: 1em;

}

}

body {

padding: 0 2em;

font-family: Arial, sans-serif;

color: #024457;

/\*background: #f2f2f2;\*/

}

h1 {

font-family: Verdana;

font-weight: normal;

/\*color: #024457;\*/

}

h1 span {

color: #167F92;

}

.box {

border: 2px solid red;

}

hr {

-moz-border-bottom-colors: none;

-moz-border-image: none;

-moz-border-left-colors: none;

-moz-border-right-colors: none;

-moz-border-top-colors: none;

border-color: #EEEEEE -moz-use-text-color #FFFFFF;

border-style: solid none;

border-width: 1px 0;

margin: 18px 0;

border: 0;

height: 1px;

background: #333;

background-image: linear-gradient(to right, #ccc, #333, #ccc);

}

**==========================THE END=====================**