**ICIN Bank Source Code:**

**Admin Controller.java**

package com.admin.controller;

import java.util.List;

import org.apache.commons.mail.EmailException;

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

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

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

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

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

import com.admin.model.ChequebookRequest;

import com.admin.model.User;

import com.admin.model.UserDisplay;

import com.admin.service.impl.AdminServiceImpl;

import com.admin.service.impl.MailServiceImpl;

@RestController

@CrossOrigin

public class AdminController {

@Autowired

private AdminServiceImpl service;

@Autowired

private MailServiceImpl mailService;

@GetMapping("user/{username}/features/{featureId}")

public void setUserFeatures(@PathVariable("username") String username, @PathVariable("featureId") int featureId) {

service.setUserFeatures(username, featureId);

}

@GetMapping("user/{username}/authorize")

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

try {

service.authorizeUser(username);

mailService.sendAuthorizedEmail(username);

} catch (EmailException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@GetMapping("user/{username}/authorize/cancel")

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

try {

mailService.sendAuthorizeCancelEmail(username);

} catch (EmailException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

service.cancelAuthorization(username);

}

@GetMapping("user/unauthorized/all")

public List<User> getAllUnauthorizedUsers()

{

return service.getAllUnauthorizedUsers();

}

@GetMapping("/user/all")

public List<UserDisplay> getAllUsers()

{

return service.getAllUsers();

}

// @GetMapping("/user/transfers/{account}")

// public List<Transfer> getTransactionList(@PathVariable("account") int account)

// {

// return service.getAllTransactions(account);

// }

@GetMapping("/chequebook/request/all")

public List<ChequebookRequest> getAllChequeBookRequests()

{

return service.getAllChequebookRequests();

}

@GetMapping("/user/{accNo}/chequebook/request/confirm")

public void confirmChequeBookRequest(@PathVariable("accNo") long accNo)

{

service.acceptChequebookRequest(accNo);

}

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

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

{

service.enableUser(username);

}

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

public void disableUser(@PathVariable("username") String username)

{

service.disableUser(username);

}

@GetMapping("search/user/{userDetail}")

public UserDisplay searchUser(@PathVariable("userDetail") String userDetail) {

return service.searchUser(userDetail);

}

}

**AccountRepository.java**

package com.admin.dao;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.Account;

@Repository

public interface AccountRepository extends CrudRepository<Account, Integer>{

public Account findByUsername(String username);

public Account findByAccno(long accno);

}

**CheckbookRequesrRepository.java**

package com.admin.dao;

import java.util.List;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.transaction.annotation.Transactional;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.ChequebookRequest;

@Repository

public interface ChequeBookRequestsRepository extends CrudRepository<ChequebookRequest, Integer>{

@Modifying

@Transactional

@Query("update ChequebookRequest c set c.requestStatus=1 where c.account = ?1")

void setChequebookInfoByAccount(long accNo);

@Query("FROM ChequebookRequest c where c.requestStatus=FALSE")

public List<ChequebookRequest> findAllChequebookRequests();

}

**TransferRepository.java**

package com.admin.dao;

import java.util.List;

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import com.admin.model.Transfer;

@Repository

public interface TransferRepository extends CrudRepository<Transfer, Integer> {

public List<Transfer> findBySaccount(long saccount);

public List<Transfer> findByRaccount(long racoount);

}

**UserDisplayRepository.java**

package com.admin.dao;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.jpa.repository.Query;

import com.admin.model.User;

import com.admin.model.UserDisplay;

public interface UserDisplayRepository extends JpaRepository<User, Integer>{

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=a.username and u.username=s.username")

public List<UserDisplay> getAllUsers();

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE u.username=?1 and u.username=a.username and u.username=s.username")

public UserDisplay getUserDetailsByUsername(String userDetail);

@Query("SELECT new com.admin.model.UserDisplay(u.fname,u.lname,u.phone,u.username,u.status,u.featureStatus,a.accno,a.balance,s.accno,s.balance)" + "FROM User u ,Account a,Saccount s WHERE s.accno=?1 and u.username=a.username and u.username=s.username")

public UserDisplay getUserDetailsByAccountNo(long accNo);

**UserRepository.java**

import org.springframework.data.repository.CrudRepository;

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import com.admin.model.User;

import java.util.List;

import org.springframework.data.jpa.repository.Modifying;

import org.springframework.data.jpa.repository.Query;

@Repository

public interface UserRepository extends CrudRepository<User, Integer> {

@Query("FROM User u WHERE u.username=?1")

public User findByUsername(String username);

@Modifying

@Transactional

@Query("update User u set u.status=1 where u.username = ?1")

void disableUser(String username);

@Modifying

@Transactional

@Query("update User u set u.status=0 where u.username = ?1")

void enableUser(String username);

@Modifying

@Transactional

@Query("update User u set u.authorizationStatus=1 where u.username = ?1")

void authorizeUser(String username);

@Modifying

@Transactional

@Query("delete from User u where u.username = ?1")

void cancelAuthorization(String username);

@Query("FROM User u where u.authorizationStatus=FALSE")

public List<User> findAllUnauthorizedAccounts();

@Modifying

@Transactional

@Query("update User u set u.featureStatus=?2 where u.username = ?1")

void setUserFeatureStatus(String username, int featureId);

}

**}**

**Admin Service.java**

package com.admin.service;

import java.util.List;

import com.admin.model.ChequebookRequest;

import com.admin.model.Transfer;

import com.admin.model.User;

import com.admin.model.UserDisplay;

public interface AdminService {

public List<UserDisplay> getAllUsers();

public List<Transfer> getAllTransactions(long accountNo);

public List<ChequebookRequest> getAllChequebookRequests();

//public void acceptChequebookRequest(int accNo);

public void enableUser(String username);

public void disableUser(String username);

public void authorizeUser(String username);

public void cancelAuthorization(String username);

public List<User> getAllUnauthorizedUsers();

public void setUserFeatures(String username, int featureId);

public UserDisplay searchUser(String userDetail);

void acceptChequebookRequest(long accNo);

}



**AdminServiceApplicatrion.java**

package com.admin;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class AdminServiceApplication {

public static void main(String[] args) {

SpringApplication.run(AdminServiceApplication.class, args);

}

}

Using frontend application

**Adminporta**l

<!doctype html>

<html lang="en" class="full-height">

<head>

<meta charset="utf-8">

<title>TemplateApp</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.0.4/popper.js"></script>

</head>

<body>

<app-root></app-root>

</body>

</html>

**Userportal**

<!doctype html>

<html lang="en" class="full-height">

<head>

<meta charset="utf-8">

<title>DreamApp</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

</head>

<body>

<app-root></app-root>

</body>

</html>

**Testing using Selenium**

package firsttestngpackage;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.\*;

import org.openqa.selenium.By.ByClassName;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.\*;

public class AdminAutomation {

public String baseUrl = "http://localhost:4201";

//String driverPath = "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe";

String driverPath = "D:\\geckodriver\\geckodriver.exe";

public WebDriver driver ;

@BeforeTest

public void launchBrowser() {

System.out.println("Launching Firefox Browser");

System.setProperty("webdriver.gecko.driver", driverPath);

driver = new FirefoxDriver();

driver.get(baseUrl);

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

}

@Test(priority=0) public void login\_Pass() {

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

driver.findElement(By.name("inputUserName")).sendKeys("madhuri");

driver.findElement(By.name("password")).sendKeys("mad12345");

//Login Button

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/button")).click();

String actualUrl="http://localhost:4200/user-account";

String expectedUrl= driver.getCurrentUrl();

if(actualUrl.equalsIgnoreCase(expectedUrl)) {

System.out.println("Login Successful"); }

driver.manage().window().maximize();

}

@Test(priority=1) public void useraccount\_login\_enabling(){

//User Account Hyperlink

driver.findElement(By.xpath("/html/body/app-root/div/nav/div/ul/li[1]/a")).click();

//Enable Button

driver.findElement(By.xpath("/html/body/app-root/app-user-account/table/tbody/tr[1]/td[6]/button")).click();

System.out.println("Enabled Login Feature");

//Disable Button

driver.findElement(By.xpath("/html/body/app-root/app-user-account/table/tbody/tr[2]/td[7]/button")).click();

System.out.println("Disabled Login Feature");

}

@Test(priority=2) public void useraccount\_features(){

//Click on the dropdown

driver.findElement(By.xpath("/html/body/app-root/app-user-account/table/tbody/tr/td[9]/select")).click();

//Select the first option

driver.findElement(By.xpath("/html/body/app-root/app-user-account/table/tbody/tr/td[9]/select/option[1]")).click();

//Set button

driver.findElement(By.xpath("/html/body/app-root/app-user-account/table/tbody/tr/td[9]/button")).click();

System.out.println("User Roles Changed");

}

@Test(priority=4) public void checkbookRequests() {

//Checkbook Request Hyperlink

driver.findElement(By.xpath("/html/body/app-root/div/nav/div/ul/li[2]/a")).click();

//Confirm Request Button

driver.findElement(By.xpath("/html/body/app-root/app-checkbook-requests/table/tbody/tr[1]/td[7]/button")).click();

System.out.println("Request Confirmed");

}

@Test(priority=3) public void authorization() {

//Authorization link

driver.findElement(By.xpath("/html/body/app-root/div/nav/div/ul/li[3]/a")).click();

//Create Account Button

driver.findElement(By.xpath("/html/body/app-root/app-authorize-registration/table/tbody/tr/td[9]/button")).click();

System.out.println("Authorized");

//Cancel Button

driver.findElement(By.xpath("/html/body/app-root/app-authorize-registration/table/tbody/tr[2]/td[10]/button")).click();

System.out.println(" Not Authorized");

}

@Test(priority=5) public void logout() {

//LogOut Button

driver.findElement(By.xpath("/html/body/app-root/div/nav/div/ul/li[4]/a")).

click(); System.out.println("Logged Out");

}

@Test(priority=6) public void login\_Fail() {

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

driver.findElement(By.name("inputUserName")).sendKeys("madhuri");

driver.findElement(By.name("password")).sendKeys("mad");

//Login Button

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/button")).click();

Alert alert = driver.switchTo().alert();

alert.accept();

String actualUrl="http://localhost:4200/user-account";

String expectedUrl= driver.getCurrentUrl();

if(!actualUrl.equalsIgnoreCase(expectedUrl)) {

System.out.println("Login UnSuccessful"); }

driver.manage().window().maximize();

}

}

**User Authentication**

package Authentication;

import org.openqa.selenium.\*;

import org.openqa.selenium.chrome.\*;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import java.util.concurrent.TimeUnit;

import org.junit.Test;

public class Login {

@Test

public void LoginTest() throws InterruptedException {

System.setProperty("webdriver.gecko.driver", "C:\\Users\\Nagaraj\\Downloads\\firefoxDriver\\geckodriver.exe");

WebDriver driver = new FirefoxDriver();

driver.get("http://localhost:4200/login");

Thread.sleep(5000);

driver.manage().timeouts().implicitlyWait(14, TimeUnit.SECONDS);

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[1]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[2]/input")).sendKeys("Novina123");

driver.findElement(By.xpath("/html/body/app-root/app-login/div/form/div[3]/button")).click();

try

{

WebDriverWait wait=new WebDriverWait(driver, 14);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("http://localhost:4200/home")));

System.out.println("Login Successfull");

// driver.findElement(By.xpath("/html/body/app-root/nav/ul/li[5]/div/a[2]")).click();

// wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-login/div/form/h3")));

// System.out.println("Signed Out");

}

catch(Exception e)

{

System.out.println("Error in browser!!\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

