**ICIN Bank**

**Code:**

Login

**package** Authentication;

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

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

**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.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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("123456p");

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");

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

driver.quit();

}

}

ChequeBookRequest:

**package** UserActions;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.firefox.FirefoxDriver;

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

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

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

**import** org.testng.annotations.Test;

**public** **class** ChequeBookRequest {

@Test

**public** **void** chequeBookRequest() **throws** InterruptedException {

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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(

"/html/body/app-root/app-home/div[1]/h2")));

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

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

driver.findElement(By.xpath("/html/body/app-root/app-cheque-book-request/div/div[2]/select")).click();

driver.findElement(By.xpath("/html/body/app-root/app-cheque-book-request/div/div[2]/select/option[1]")).click();

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

System.out.println("Cheque Book Issued Successfully!!");

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

driver.quit();

}

}

EditProfile:

**package** UserActions;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.firefox.FirefoxDriver;

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

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

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

**import** org.testng.annotations.Test;

**public** **class** EditProfile {

@Test

**public** **void** EditPorfile() **throws** InterruptedException {

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver= **new** ChromeDriver();

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

Thread.sleep(5000);

**try**

{

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

driver.findElement(By.xpath("//\*[@id=\"dropdown04\"]")).click();

driver.findElement(By.xpath("/html/body/app-root/nav/div/ul/li[5]/div/a[1]")).click();

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[1]/input")).sendKeys("7666854389");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/input")).sendKeys("Mumbai");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[1]/input")).sendKeys("novinarayudu.97@gmail.com");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[2]/input")).sendKeys("123P1234");

driver.findElement(By.xpath("/html/body/app-root/app-edit-profile/div[1]/form/div/div[2]/div[3]/input")).sendKeys("123P1234");

System.out.println("Profile edited");

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

driver.findElement(By.xpath("//\*[@id=\"dropdown04\"]")).click();

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

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

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

}

}

Register:

**package** UserActions;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

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

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

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

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

**import** org.testng.annotations.Test;

**public** **class** Register {

@Test

**public** **void** register() **throws** InterruptedException {

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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

Thread.sleep(5000);

**try** {

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

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

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

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[3]/input")).sendKeys("Novina");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[4]/input")).sendKeys("123456p");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[5]/input")).sendKeys("17/08/2020");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[6]/input")).sendKeys("7666854389");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[7]/input")).sendKeys("Mumbai");

Select id=**new** Select(driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[8]/select")));

id.selectByIndex(2);

WebElement fileInput = driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[9]/input"));

fileInput.sendKeys("C:\\Users\\USHA\\Desktop\\MAVURI USHASRI\\KYC\\demo.txt");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[10]/input")).sendKeys("abcde1234");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[11]/input")).sendKeys("novinarayudu.97@gmail.com");

driver.findElement(By.xpath("/html/body/app-root/app-register/div/form/div/div[12]/button")).click();

Thread.sleep(5000);

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

}

**catch**(Exception e) {

System.out.println("Erro in web browser\nPlease have a look");

}

Thread.sleep(5000);

driver.quit();

}

}

TransactionHistory:

**package** UserActions;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.firefox.FirefoxDriver;

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

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

**import** org.testng.annotations.Test;

**public** **class** TransactionHistory {

@Test

**public** **void** TransactionHistory() **throws** InterruptedException {

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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

Thread.sleep(5000);

**try**

{

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

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

System.out.println("Transaction History Displayed");

driver.findElement(By.xpath("//\*[@id=\"navbardrop\"]")).click();

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

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

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

driver.quit();

}

}

TransferHistory:

**package** UserActions;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.openqa.selenium.firefox.FirefoxDriver;

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

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

**import** org.testng.annotations.Test;

**public** **class** TransferHistory {

@Test

**public** **void** TransactionHistory() **throws** InterruptedException {

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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

Thread.sleep(5000);

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

**try**

{

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

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

System.out.println("Transfer History Displayed");

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

driver.quit();

}

}

TransferMoney:

**package** UserActions;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

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

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

**import** org.testng.annotations.Test;

**public** **class** TransferMoney {

@Test

**public** **void** TransferMoney() **throws** InterruptedException{

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

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("/html/body/app-root/app-home/div[1]/h2")));

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

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

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

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

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[3]/input")).sendKeys("123456789");

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[4]/input")).sendKeys("1234");

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[5]/input")).sendKeys("1000");

Thread.sleep(5000);

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

driver.findElement(By.xpath("/html/body/app-root/app-transfer-between-accounts/div/form/div[6]/button")).click();

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/app-root/app-home/div[1]/h2")));

System.out.println("Transfer Money Successfull");

}

**catch**(Exception e)

{

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

}

Thread.sleep(5000);

driver.quit();

}

}

AdminAutomation:

**package** firsttestngpackage;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.Alert;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

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

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.Test;

**public** **class** AdminAutomation {

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

WebDriver driver;

@BeforeSuite

**public** **void** launchBrowser() {

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

System.setProperty("webdriver.chrome.driver", "F:\\Mphasis\\project\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.get(baseUrl);

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

}

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

driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);

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

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

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(){

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

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");

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(){

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

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

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() {

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

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() {

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

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

System.out.println("Authorized");

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() {

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(5, TimeUnit.SECONDS);

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

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

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();

}

}

RequestFilter:

**package** com.icinbank.config;

**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;

**import** org.springframework.core.Ordered;

**import** org.springframework.core.annotation.Order;

**import** org.springframework.stereotype.Component;

@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() {}

}

SecurityConfig:

**package** com.icinbank.config;

**import** java.security.SecureRandom;

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

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.core.env.Environment;

**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.icinbank.service.UserServiceImpl.UserSecurityService;

@Configuration

@EnableWebSecurity

@EnableGlobalMethodSecurity(prePostEnabled=**true**)

**public** **class** SecurityConfig **extends** WebSecurityConfigurerAdapter {

@Autowired

**private** Environment env;

@Autowired

**private** UserSecurityService userSecurityService;

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

@Bean

**public** BCryptPasswordEncoder passwordEncoder() {

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

}

**private** **static** **final** String[] PUBLIC\_MATCHERS = {

"/webjars/\*\*",

"/css/\*\*",

"/js/\*\*",

"/images/\*\*",

"/",

"/about/\*\*",

"/contact/\*\*",

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

"/console/\*\*",

"/signup"

};

@Override

**protected** **void** configure(HttpSecurity http) **throws** Exception {

http

.authorizeRequests().

antMatchers(PUBLIC\_MATCHERS).

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

http

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

.formLogin().failureUrl("/index?error").defaultSuccessUrl("/ICINBank").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.userDetailsService(userSecurityService).passwordEncoder(passwordEncoder());

}

}

<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>AdminPortalAutomation</groupId>

<artifactId>AdminPortalAutomation</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.14.3</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.141.59</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.webjars/jquery -->

<dependency>

<groupId>org.webjars</groupId>

<artifactId>jquery</artifactId>

<version>3.5.1</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src</sourceDirectory>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

<fork>true</fork>

<executable>C:\Program Files\Java\jdk1.8.0\_171\bin\javac.exe</executable>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M3</version>

</plugin>

</plugins>

</build>

</project>

import { AppPage } from './app.po';

import { browser, logging } from 'protractor';

describe('workspace-project App', () **=>** {

**let** page: AppPage;

beforeEach(() **=>** {

page = new AppPage();

});

it('should display welcome message', () **=>** {

page.navigateTo();

expect(page.getTitleText()).toEqual('AdminPortal app is running!');

});

afterEach(**async** () **=>** {

**const** logs = await browser.manage().logs().get(logging.Type.BROWSER);

expect(logs).not.toContain(jasmine.objectContaining({

level: logging.Level.SEVERE,

} as logging.Entry));

});

});

import { browser, by, element } from 'protractor';

export **class** AppPage {

navigateTo(): Promise<unknown> {

return browser.get(browser.baseUrl) as Promise<unknown>;

}

getTitleText(): Promise<string> {

return element(by.css('app-root .content span')).getText() as Promise<string>;

}

}

**const** { SpecReporter } = require('jasmine-spec-reporter');

exports.config = {

allScriptsTimeout: 11000,

specs: [

'./src/\*\*/\*.e2e-spec.ts'

],

capabilities: {

browserName: 'chrome'

},

directConnect: true,

baseUrl: 'http://localhost:4200/',

framework: 'jasmine',

jasmineNodeOpts: {

showColors: true,

defaultTimeoutInterval: 30000,

print: **function**() {}

},

onPrepare() {

require('ts-node').register({

project: require('path').join(\_\_dirname, './tsconfig.json')

});

jasmine.getEnv().addReporter(new SpecReporter({ spec: { displayStacktrace: true } }));

}

};

{

"extends": "../tsconfig.json",

"compilerOptions": {

"outDir": "../out-tsc/e2e",

"module": "commonjs",

"target": "es5",

"types": [

"jasmine",

"jasminewd2",

"node"

]

}

<h1>Cheque Book Requests</h1>

<table class=*"table table-striped"*>

<thead>

<tr>

<th>Request Id</th>

<th>User Name</th>

<th>Description</th>

<th>Confirmed?</th>

<th>Action</th>

</tr>

</thead>

<tbody>

<tr \*ngFor=*"let chequeBookRequest of chequeBookRequestList"*>

<td>{{chequeBookRequest.id}}</td>

<td>{{chequeBookRequest.user.username}}</td>

<td>{{chequeBookRequest.description}}</td>

<td>{{chequeBookRequest.confirmed}}</td>

<td [hidden]=*"chequeBookRequest.confirmed"*><a (click)=*"confirmChequeBookRequest(chequeBookRequest.id)"* style="cursor: *pointer*;">Confirm</a></td>

<!-- <td [hidden]="!user.enabled"><a (click)="disableUser(user.username)" style="cursor: pointer;">Disable</a></td> -->

</tr>

</tbody>

</table>

/\* tslint:disable:no-unused-variable \*/

import { TestBed, async } from '@angular/core/testing';

import { ChequeBookRequestComponent } from './chequeBookRequest.component';

describe('Component: ChequeBookRequest', () **=>** {

it('should create an instance', () **=>** {

**let** component = new ChequeBookRequestComponent();

expect(component).toBeTruthy();

});

});

import { Component, OnInit } from '@angular/core';

import {ChequeBookRequestService} from '../chequeBookRequest.service';

@Component({

selector: 'app-chequeBookRequest',

templateUrl: './chequeBookRequest.component.html',

styleUrls: ['./chequeBookRequest.component.css']

})

export **class** ChequeBookRequestComponent **implements** OnInit {

chequeBookRequestList: Object[];

**constructor**(**private** chequeBookRequestService: ChequeBookRequestService) {

this.getChequeBookRequestList();

}

getChequeBookRequestList() {

this.chequeBookRequestService.getChequeBookRequestList().subscribe(

res **=>** {

this.chequeBookRequestList = JSON.parse(JSON.stringify(res));

},

error **=>** console.log(error)

)

}

confirmChequeBookRequest(id: number) {

this.chequeBookRequestService.confirmChequeBookRequest(id).subscribe();

location.reload();

}

ngOnInit() {}

}

*.form-signin*{

background-color: *#f5f5f2*;

}

<div class=*"wrapper"*>

<form class=*"form-signin"* (ngSubmit)=*"onSubmit()"* [hidden]=*"loggedIn"*>

<h2 class=*"clean-font"*>Please login</h2>

<input type=*"text"* class=*"form-control"* name=*"username"* [(ngModel)]=*"username"* placeholder=*"Username"* name=*"username"* required autofocus=*""* />

<br />

<input type=*"password"* class=*"form-control"* name=*"password"* [(ngModel)]=*"password"* placeholder=*"Password"* required />

<div class=*"form-group"*>

<br/>

<label >

<input type=*"checkbox"* name=*"remember-me"* id=*"remember-me"* />&nbsp;<span class=*"clean-font"*>Remember me</span>

</label>

</div>

<button class=*"btn btn-primary btn-block"* type=*"submit"*>Login</button>

</form>

<div [hidden]=*"!loggedIn"*>

<h2>Welcome to Admin Portal!</h2>

</div>

</div>

import { TestBed, async } from '@angular/core/testing';

import { LoginComponent } from './login.component';

describe('Component: Login', () **=>** {

it('should create an instance', () **=>** {

**let** component = new LoginComponent();

expect(component).toBeTruthy();

});

});

import { Component, OnInit } from '@angular/core';

import {Observable} from 'rxjs/Observable';

import {LoginService} from '../login.service';

@Component({

selector: 'app-login',

templateUrl: './login.component.html',

styleUrls: ['./login.component.css']

})

export **class** LoginComponent **implements** OnInit {

loggedIn: boolean;

username: string;

password: string;

**constructor** (**private** loginService: LoginService) {

if(localStorage.getItem('PortalAdminHasLoggedIn') == '' || localStorage.getItem('PortalAdminHasLoggedIn') == null) {

this.loggedIn = false;

} else {

this.loggedIn = true;

}

}

onSubmit() {

this.loginService.sendCredential(this.username, this.password).subscribe(

res **=>** {

this.loggedIn=true;

localStorage.setItem('PortalAdminHasLoggedIn', 'true');

location.reload();

},

err **=>** console.log(err)

);

}

ngOnInit() {}

}

*.navbar-clean*{

background-color: *#f5f5f2*;

}

<nav class=*"navbar navbar-clean"*>

<div class=*"container-fluid"*>

<div class=*"navbar-header"*>

<button type=*"button"* class=*"navbar-toggle collapsed"* data-toggle=*"collapse"* data-target=*"#bs-example-navbar-collapse-1"*>

<span class=*"sr-only"*>Toggle navigation</span>

<span class=*"icon-bar"*></span>

<span class=*"icon-bar"*></span>

<span class=*"icon-bar"*></span>

</button>

<a class=*"navbar-brand"* routerLink=*"/login"* routerLinkActive=*"active"*>Admin Portal</a>

</div>

<div class=*"collapse navbar-collapse"* id=*"bs-example-navbar-collapse-1"*>

<ul class=*"nav navbar-nav"*>

<li [style.display]=*"getDisplay()"*><a routerLink=*"/userAccount"* routerLinkActive=*"active"*> User Account <span class=*"sr-only"*>(current)</span></a></li>

<li [style.display]=*"getDisplay()"*><a routerLink=*"/chequeBookRequest"* routerLinkActive=*"active"*> Cheque Book Requests <span class=*"sr-only"*>(current)</span></a></li>

</ul>

<ul class=*"nav navbar-nav navbar-right"*>

<li [style.display]=*"getDisplay()"*><a (click)=*"logout()"* style="cursor: *pointer*;">Logout</a></li>

</ul>

</div>

</div>

</nav>

import { TestBed, async } from '@angular/core/testing';

import { NavbarComponent } from './navbar.component';

describe('Component: Navbar', () **=>** {

it('should create an instance', () **=>** {

**let** component = new NavbarComponent();

expect(component).toBeTruthy();

});

});

import { Component, OnInit } from '@angular/core';

import { LoginService } from '../login.service';

import { Router } from '@angular/router';

@Component({

selector: 'app-navbar',

templateUrl: './navbar.component.html',

styleUrls: ['./navbar.component.css']

})

export **class** NavbarComponent **implements** OnInit {

loggedIn: boolean;

**constructor**(**private** loginService: LoginService, **private** router : Router) {

if(localStorage.getItem('PortalAdminHasLoggedIn') == '') {

this.loggedIn = false;

} else {

this.loggedIn = true;

}

}

logout(){

this.loginService.logout().subscribe(

res **=>** {

localStorage.setItem('PortalAdminHasLoggedIn', '');

this.loggedIn=false;

},

err **=>** console.log(err)

);

location.reload();

this.router.navigate(['/login']);

}

getDisplay() {

if(!this.loggedIn){

return "none";

} else {

return "";

}

}

ngOnInit() {

}

}

<h1>Primary Account Transaction List</h1>

<table class=*"table table-striped"*>

<thead>

<tr>

<th>Post Date</th>

<th>Description</th>

<th>Type</th>

<th>Status</th>

<th>Amount</th>

<th>Available Balance</th>

</tr>

</thead>

<tbody>

<tr \*ngFor=*"let primaryTransaction of primaryTransactionList"*>

<td>{{primaryTransaction.date | date: 'MM/dd/yyyy'}} </td>

<td>{{primaryTransaction.description}}</td>

<td>{{primaryTransaction.type}}</td>

<td>{{primaryTransaction.status}}</td>

<td>{{primaryTransaction.amount}}</td>

<td>{{primaryTransaction.availableBalance}}</td>

</tr>

</tbody>

</table>

/\* tslint:disable:no-unused-variable \*/

import { TestBed, async } from '@angular/core/testing';

import { PrimaryTransactionComponent } from './primary-transaction.component';

describe('Component: PrimaryTransaction', () **=>** {

it('should create an instance', () **=>** {

**let** component = new PrimaryTransactionComponent();

expect(component).toBeTruthy();

});

});

import { Component, OnInit } from '@angular/core';

import {UserService} from '../user.service';

import { ActivatedRoute, Params } from '@angular/router';

@Component({

selector: 'app-primary-transaction',

templateUrl: './primary-transaction.component.html',

styleUrls: ['./primary-transaction.component.css']

})

export **class** PrimaryTransactionComponent **implements** OnInit {

username:string;

primaryTransactionList: Object[];

**constructor**(**private** route: ActivatedRoute, **private** userService: UserService) {

this.route.params.forEach((params: Params) **=>** {

this.username = params['username'];

});

this.getPrimaryTransactionList();

}

getPrimaryTransactionList() {

this.userService.getPrimaryTransactionList(this.username).subscribe(

res **=>** {

console.log(JSON.parse(JSON.stringify(res)).\_body);

this.primaryTransactionList = JSON.parse(JSON.stringify(res));

},

error **=>** console.log(error)

)

}

ngOnInit() {}

}