STIRLING
EDUCATION

IT 303 – Software Verification, Validation and Testing

TESTING DOCUMENTATION

EatingWell: Healthy Recipes, Healthy Eating

Prepared by:
Lamija Šetić
Hazrin Redžepi

Proposed to:
Samed Jukić, Assist. Prof. Dr.
Adnan Miljković, Teaching Assistant
Benjamin Peljto, Lecture Assistant

12.1.2025.

Contents

1. Introduction

1.1. About the Project

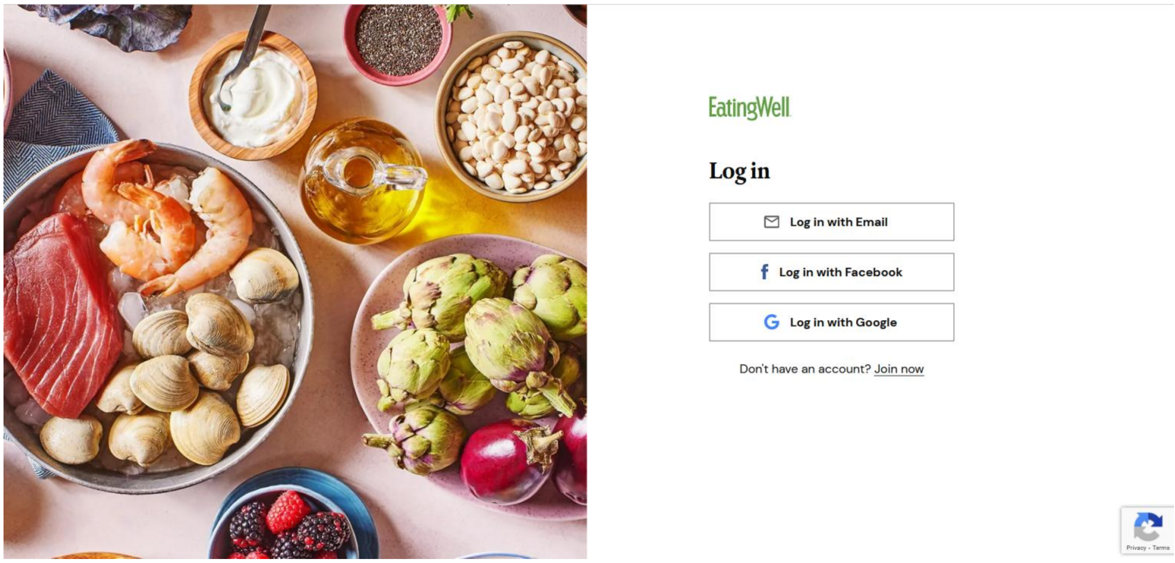
EatingWell.com is a health-focused culinary website offering a rich collection of recipes, meal plans, and cooking tips designed to promote balanced and nutritious eating. It features expert-crafted content tailored to various dietary needs, such as gluten-free, vegan, and heart-healthy options. The platform also includes articles on wellness, sustainable living, and product recommendations, making it a comprehensive resource for healthy lifestyle enthusiasts.

Homepage: https://www.eatingwell.com

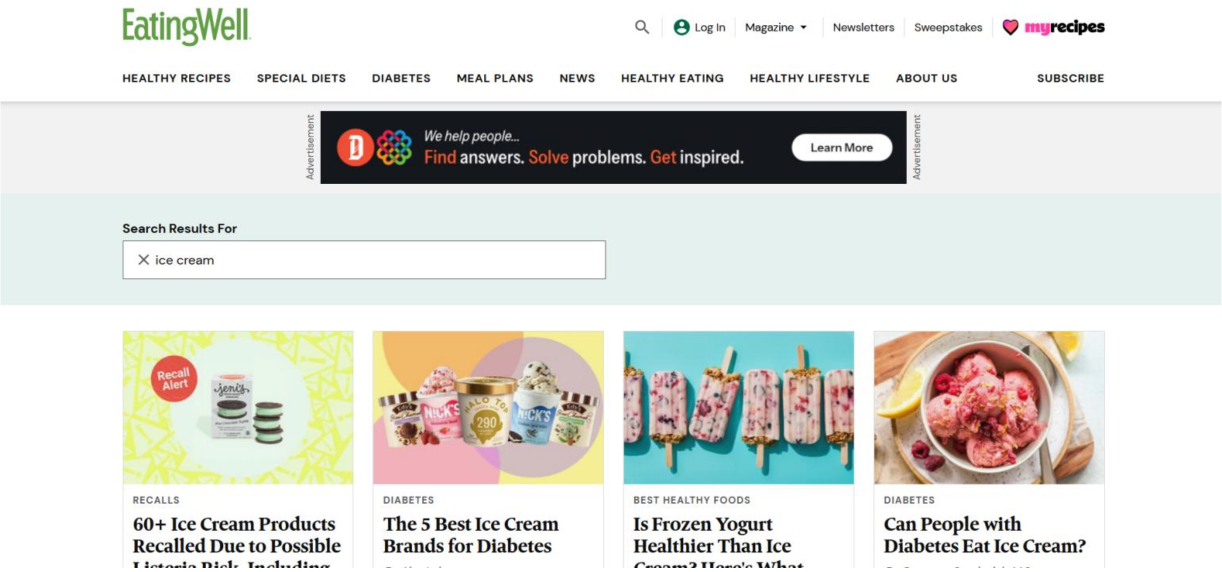
1.2. Project Functionalities and Screenshots

Main features of our project are:

1. Login and register



2. Searching for articles or recipes



EatingWell

Log In | Magazine | Newsletters | Sweepstakes | myrecipes

HEALTHY RECIPES | SPECIAL DIETS | DIABETES | MEAL PLANS | NEWS | HEALTHY EATING | HEALTHY LIFESTYLE | ABOUT US | SUBSCRIBE

Search Results For

ice cream

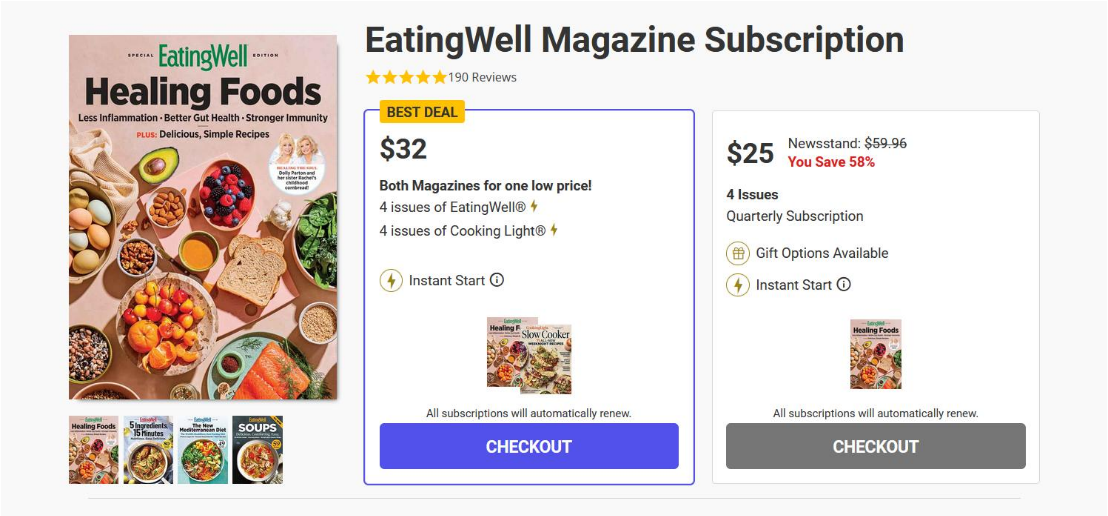
RECALLS
60+ Ice Cream Products Recalled Due to Possible Listeria Risk, Including

DIABETES
The 5 Best Ice Cream Brands for Diabetes

BEST HEALTHY FOODS
Is Frozen Yogurt Healthier Than Ice Cream? Here's What

DIABETES
Can People with Diabetes Eat Ice Cream?

3. Subscription-Based Magazine: Publishes a print magazine featuring in-depth articles, recipes, and special editions focused on specific topics related to healthy eating and lifestyle. There is checkout form also. ***Important notice*:** This page (**https://www.magazines.com/**) is a completely separate page that is located within our main page that we are testing because it's connected to some of the options (**subscribe**) inside our page.



EatingWell Magazine Subscription

★★★★★ 190 Reviews

BEST DEAL

$32

Both Magazines for one low price!

4 issues of EatingWell®

4 issues of Cooking Light®

Instant Start ⓘ

All subscriptions will automatically renew.

CHECKOUT

$25 Newsstand: ~~$59.96~~ You Save 58%

4 Issues

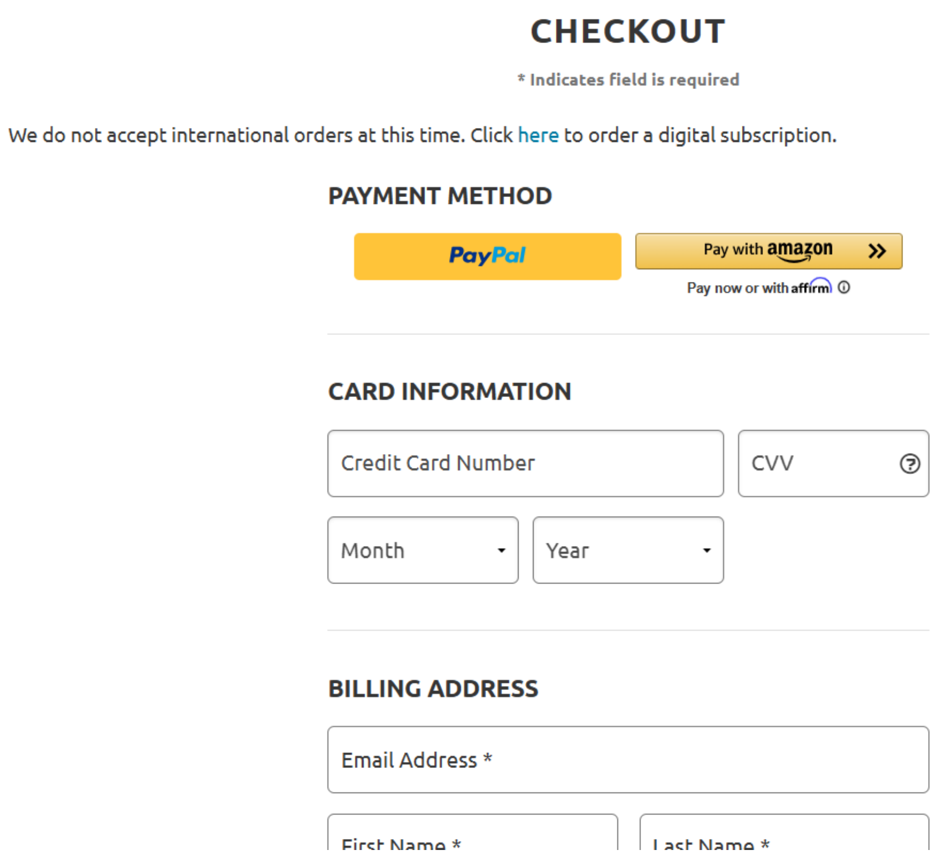
Quarterly Subscription

Gift Options Available

Instant Start ⓘ

All subscriptions will automatically renew.

CHECKOUT



CHECKOUT

* Indicates field is required

We do not accept international orders at this time. Click here to order a digital subscription.

PAYMENT METHOD

CARD INFORMATION

Credit Card Number

CVV

Month

Year

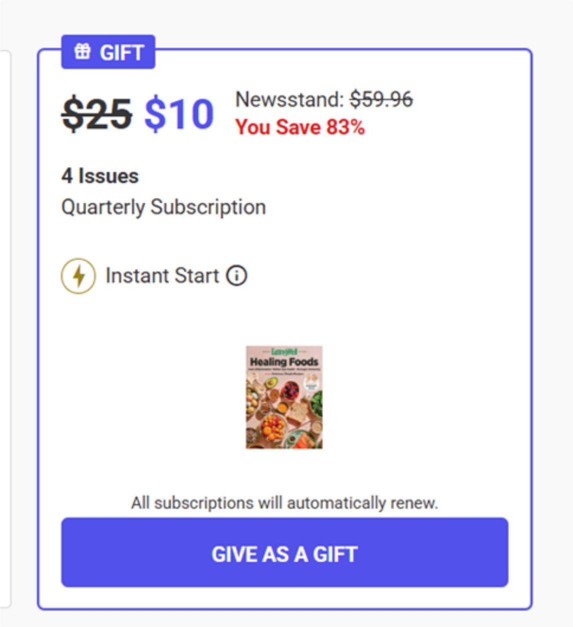
BILLING ADDRESS

Email Address *

First Name *

Last Name *

4. In this section, we also have “give a gift subscription” option and checkout form for this option, too.



GIFT

~~$25~~ $10 Newsstand: ~~$59.96~~

You Save 83%

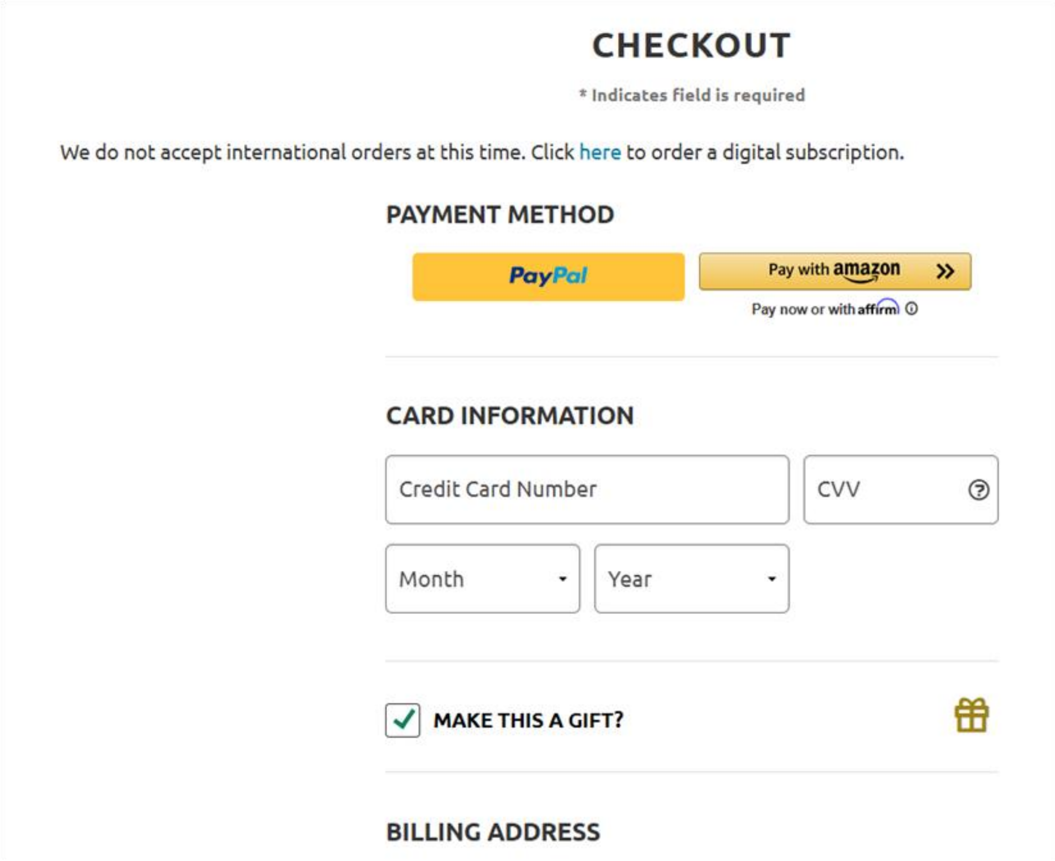
4 Issues

Quarterly Subscription

Instant Start

All subscriptions will automatically renew.

GIVE AS A GIFT



CHECKOUT

* Indicates field is required

We do not accept international orders at this time. Click here to order a digital subscription.

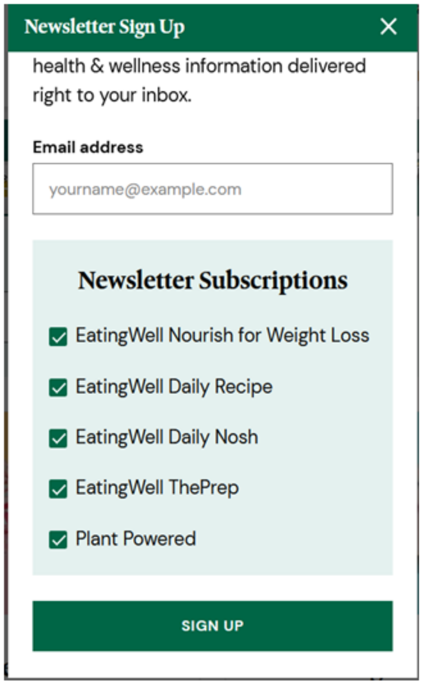
PAYMENT METHOD

CARD INFORMATION

☑ MAKE THIS A GIFT?

BILLING ADDRESS

5. Newsletter sign up



Newsletter Sign Up

health & wellness information delivered right to your inbox.

Email address

Newsletter Subscriptions

☑ EatingWell Nourish for Weight Loss

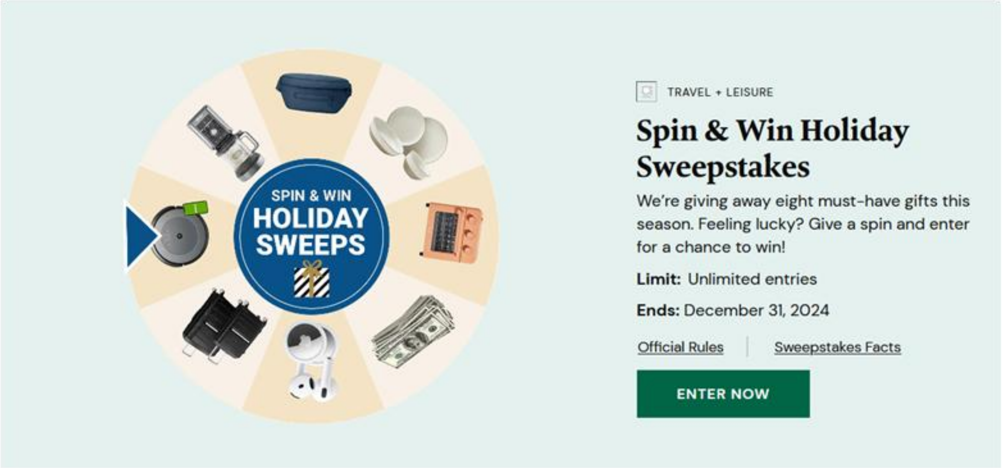
☑ EatingWell Daily Recipe

☑ EatingWell Daily Nosh

☑ EatingWell ThePrep

☑ Plant Powered

6. Sweepstakes



TRAVEL + LEISURE

Spin & Win Holiday Sweepstakes

We're giving away eight must-have gifts this season. Feeling lucky? Give a spin and enter for a chance to win!

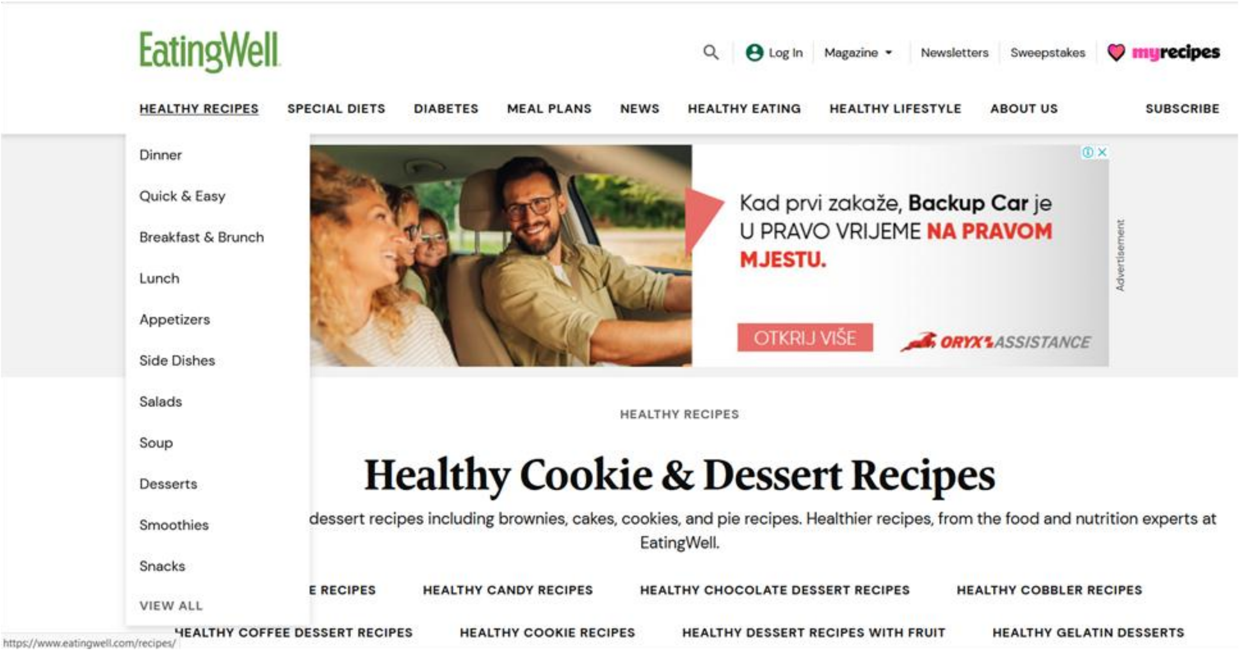
Limit: Unlimited entries

Ends: December 31, 2024

Official Rules | Sweepstakes Facts

ENTER NOW

7. Extensive Recipe Collection in navbar: Offers a wide array of healthy recipes categorized by meal type, dietary preference, and cuisine, assisting users in finding suitable meal options.



Kad prvi zakaže, **Backup Car** je U PRAVO VRIJEME **NA PRAVOM MJESTU.**

HEALTHY RECIPES

Healthy Cookie & Dessert Recipes

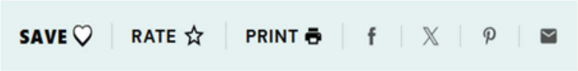
dessert recipes including brownies, cakes, cookies, and pie recipes. Healthier recipes, from the food and nutrition experts at EatingWell.

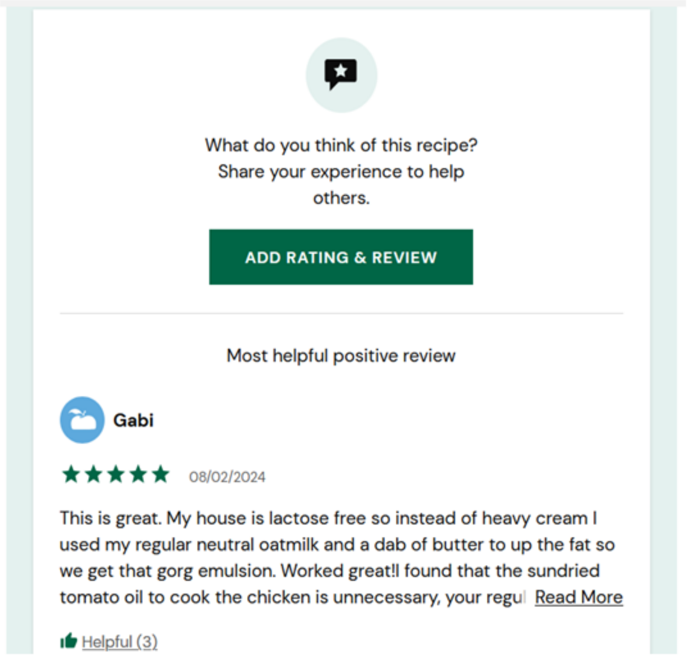
E RECIPES | HEALTHY CANDY RECIPES | HEALTHY CHOCOLATE DESSERT RECIPES | HEALTHY COBBLER RECIPES

HEALTHY COFFEE DESSERT RECIPES | HEALTHY COOKIE RECIPES | HEALTHY DESSERT RECIPES WITH FRUIT | HEALTHY GELATIN DESSERTS

https://www.eatingwell.com/recipes/

8. Adding, rating and sharing recipes/articles reviews and recommendations




What do you think of this recipe? Share your experience to help others.

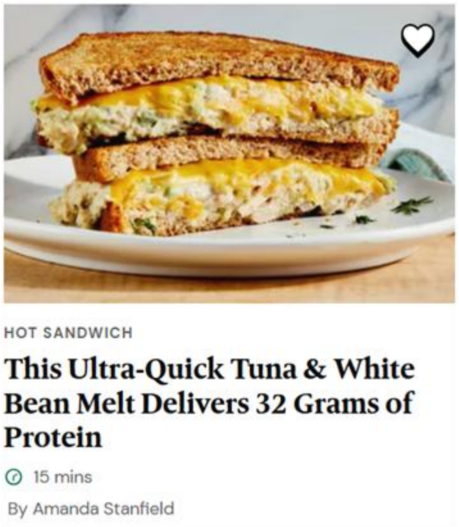
Most helpful positive review

Gabi

★★★★★ 08/02/2024

This is great. My house is lactose free so instead of heavy cream I used my regular neutral oatmilk and a dab of butter to up the fat so we get that gorg emulsion. Worked great!! found that the sundried tomato oil to cook the chicken is unnecessary, your regu Read More

9. Adding hearts/Saving recipes in our favorites. ***Important notice*:** This page (**https://www.myrecipes.com/**) is a completely separate page that is located within our main page that we are testing because it's connected to some of the options (**adding hearts to recipes**) inside our page. So we tested that too because it has an important impact on our main page.

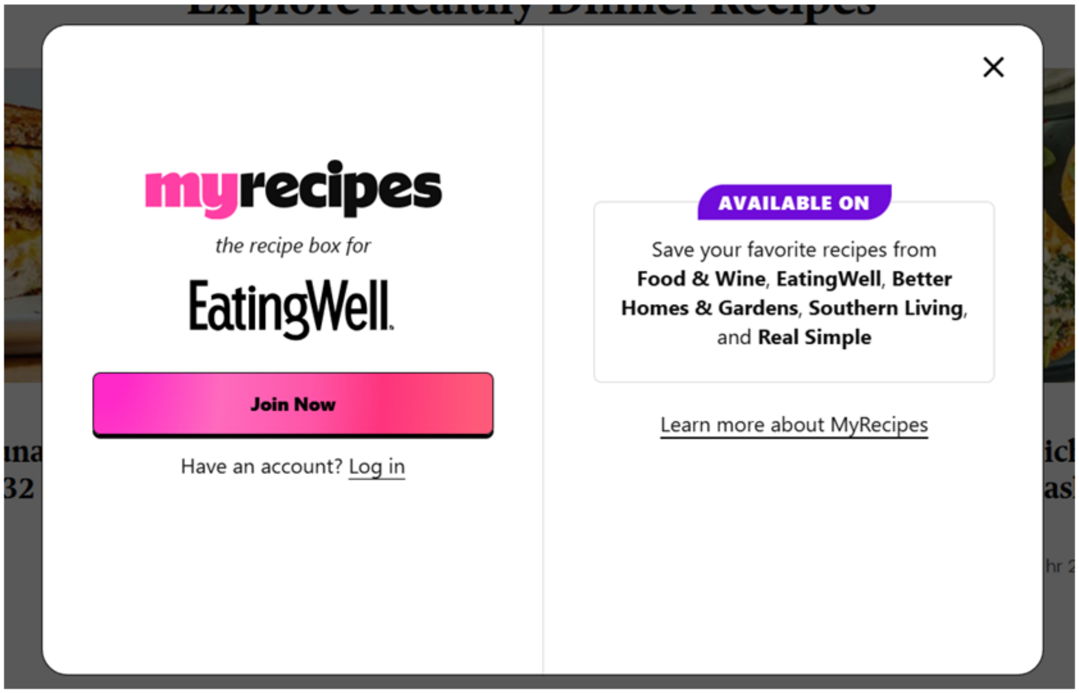

HOT SANDWICH

This Ultra-Quick Tuna & White Bean Melt Delivers 32 Grams of Protein

15 mins

By Amanda Stanfield

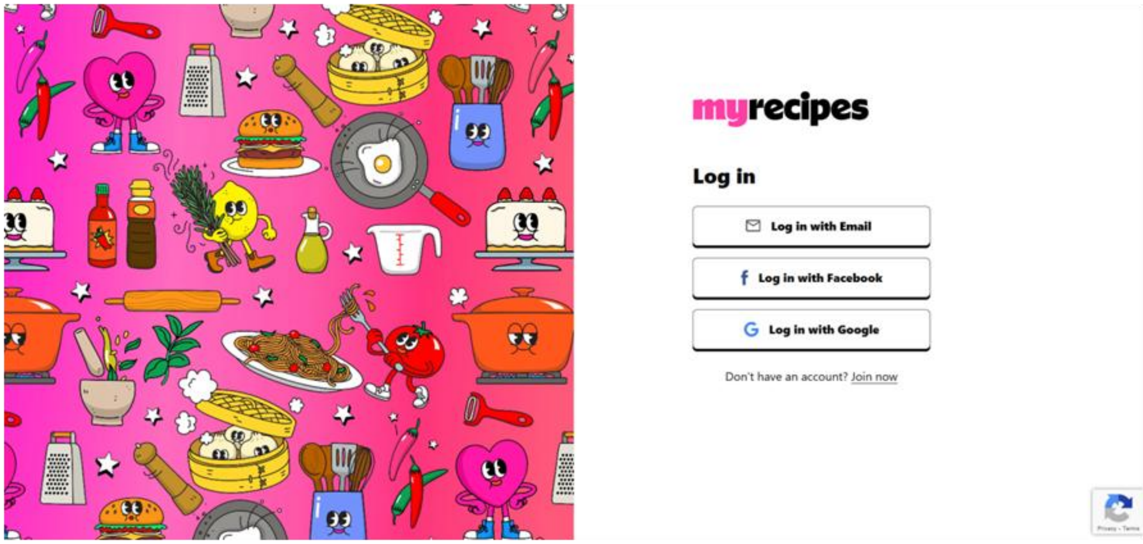
10. After adding heart, there is this pop up for **login/registration** for this page



the recipe box for EatingWell.

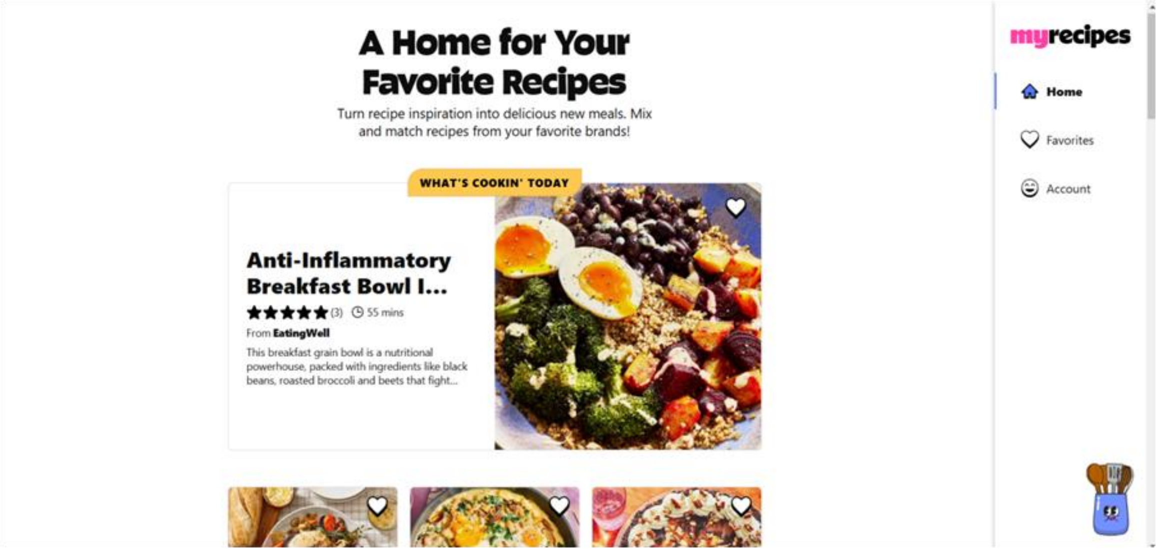
Have an account? Log in

Save your favorite recipes from **Food & Wine**, **EatingWell**, **Better Homes & Gardens**, **Southern Living**, and **Real Simple**



Don't have an account? Join now

11. Homepage and navbar of myrecipes page



A Home for Your Favorite Recipes

Turn recipe inspiration into delicious new meals. Mix and match recipes from your favorite brands!

WHAT'S COOKIN' TODAY

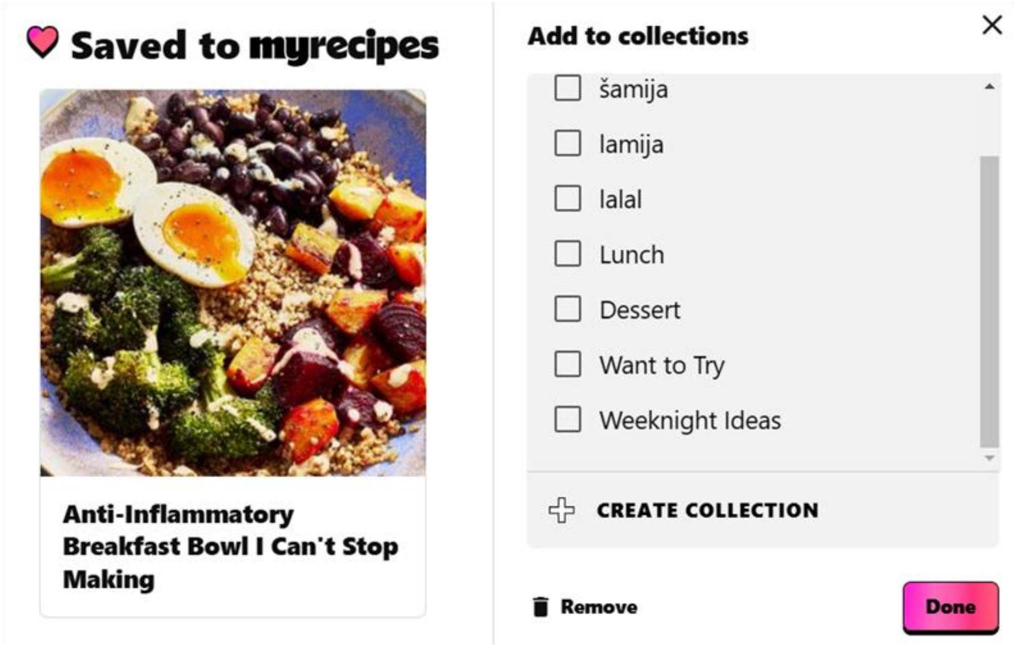
Anti-Inflammatory Breakfast Bowl I...

★★★★★ (3) 55 mins

From **EatingWell**

This breakfast grain bowl is a nutritional powerhouse, packed with ingredients like black beans, roasted broccoli and beets that fight...

12. Adding/removing recipes to favorites or collection



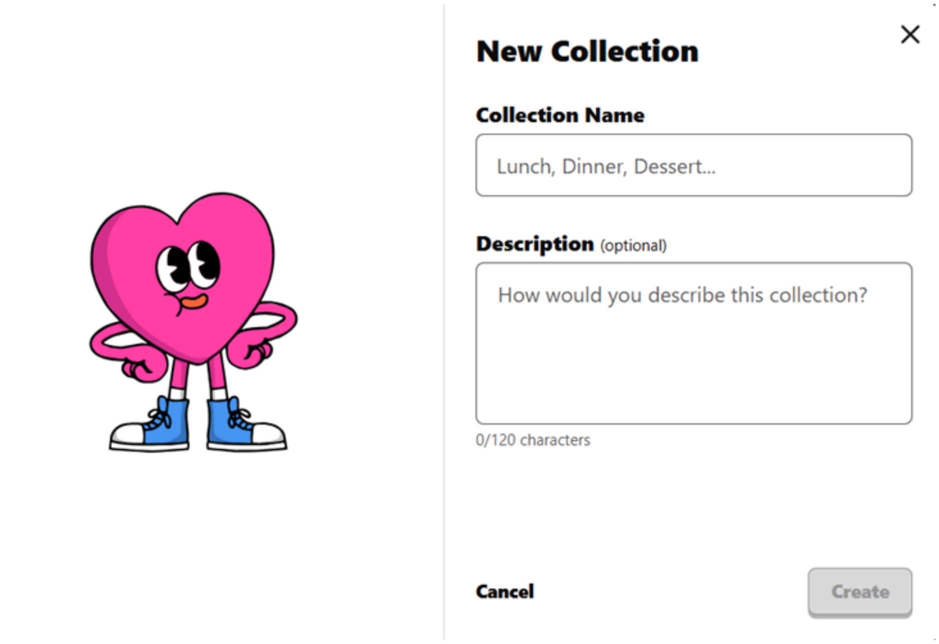
Saved to myrecipes

Anti-Inflammatory Breakfast Bowl I Can't Stop Making

Add to collections

□ šamija
□ lamija
□ lalal
□ Lunch
□ Dessert
□ Want to Try
□ Weeknight Ideas

13. Making new collection of recipes



New Collection

Collection Name

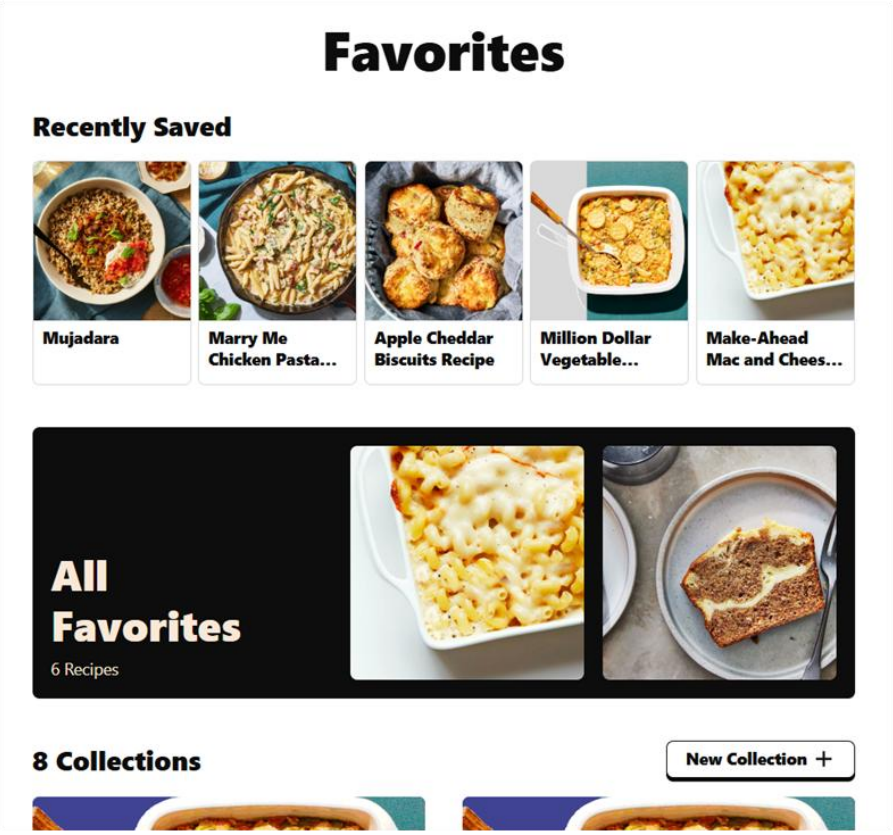
Lunch, Dinner, Dessert...

Description (optional)

How would you describe this collection?

0/120 characters

14. Collections with saved recipes



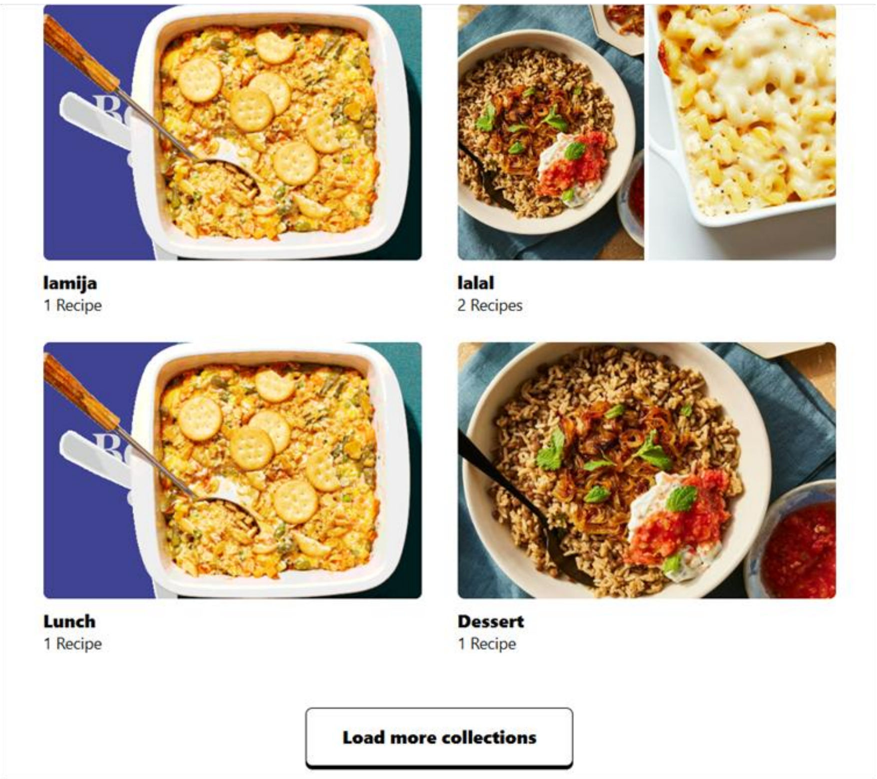
Favorites

Recently Saved

- Mujadara
- Marry Me Chicken Pasta...
- Apple Cheddar Biscuits Recipe
- Million Dollar Vegetable...
- Make-Ahead Mac and Chees...

All Favorites
6 Recipes

8 Collections



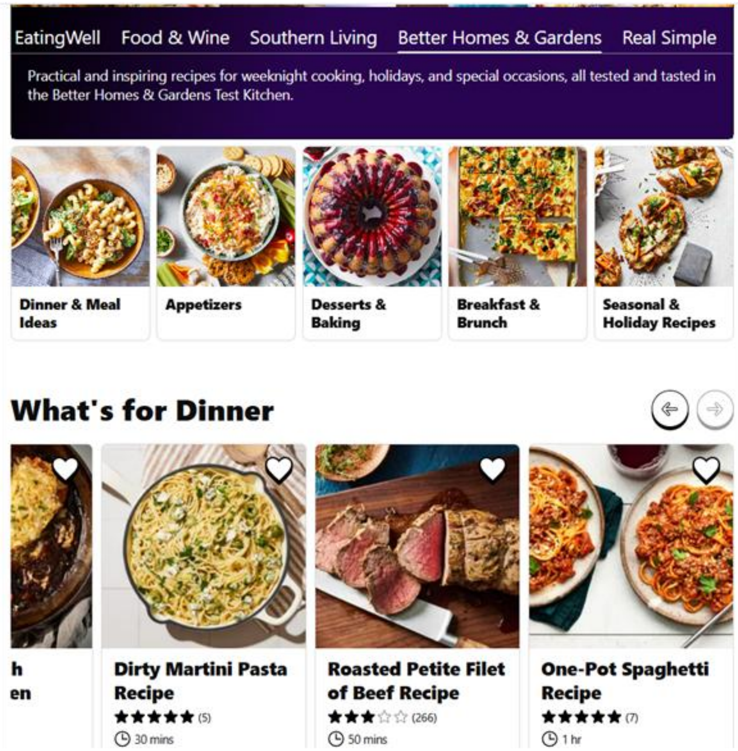
lamija
1 Recipe

lalal
2 Recipes

Lunch
1 Recipe

Dessert
1 Recipe

15. Searching through carousel of recipes



Practical and inspiring recipes for weeknight cooking, holidays, and special occasions, all tested and tasted in the Better Homes & Gardens Test Kitchen.

Dinner & Meal Ideas

Appetizers

Desserts & Baking

Breakfast & Brunch

Seasonal & Holiday Recipes

What's for Dinner

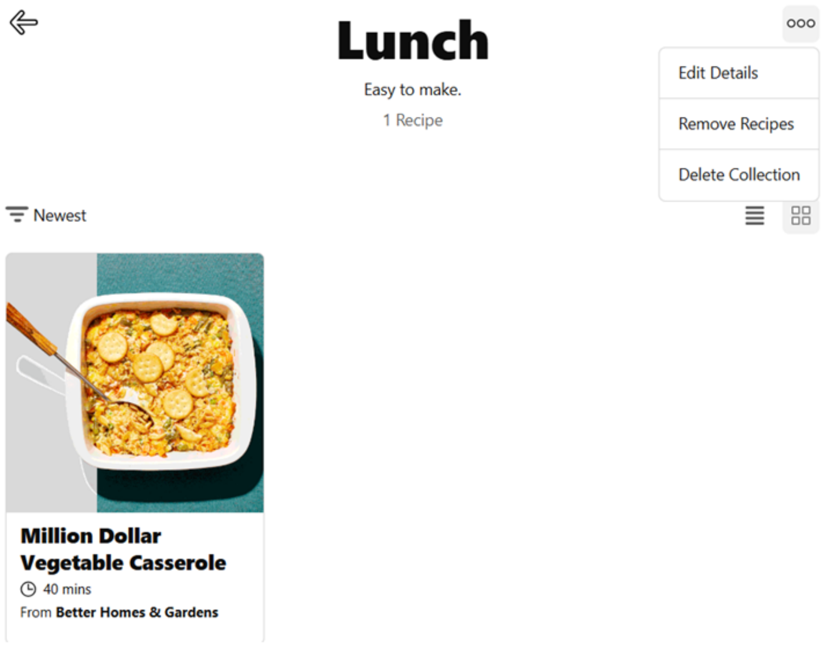
h
en

Dirty Martini Pasta Recipe
★★★★★ (5)
30 mins

Roasted Petite Filet of Beef Recipe
★★★☆☆ (266)
50 mins

One-Pot Spaghetti Recipe
★★★★★ (7)
1 hr

16. Edit, remove or delete recipes/collection



Lunch

Easy to make.

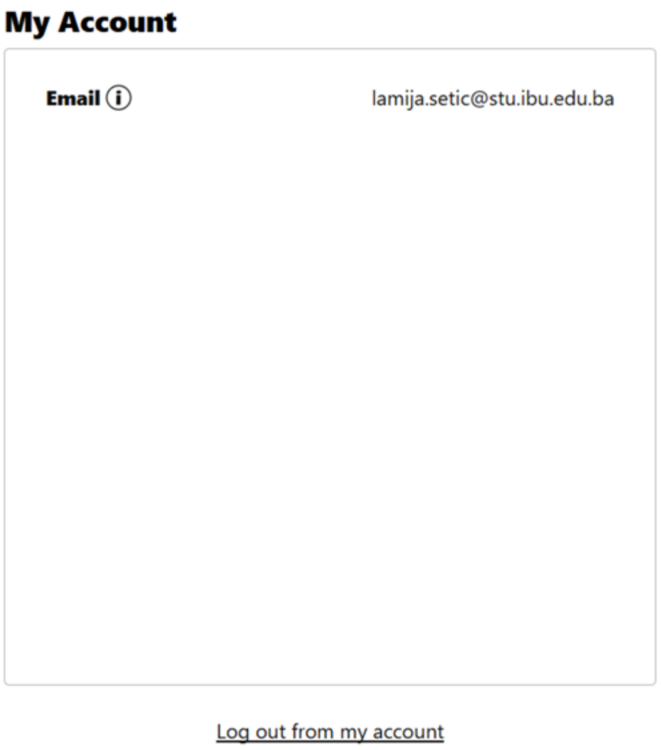
1 Recipe

Million Dollar Vegetable Casserole

40 mins

From Better Homes & Gardens

17. Log out option and account info



My Account

Email: lamija.setic@stu.ibu.edu.ba

2. Test Plan

2.1. Scope

Our website consists of two additional pages, each with its own functionalities. However, the main goal of our project was to test the critical functionalities of our main website, as well as some additional functionalities related to it, which were included in the tests and test cases.

The first additional page is magazines.com. It is a separate page but is connected to our website due to the subscribe option, which offers various subscription types for users. This makes it partly a critical functionality of our website. However, it is located on this additional page. Therefore, we tested the subscription-related functionalities, but we did not test the specific functionalities of magazines.com itself because we considered it additional work that was not directly relevant to our main website.

The second page is the My Recipes page, which is integrated into our website. On our main site, users can add recipes and create recipe collections, which are saved on the My Recipes page. Users must create a separate account on My Recipes, complete with authentication and authorization processes. While it is a standalone page, it is closely connected to our website because it shares critical functionalities like saving recipes and creating collections. Even though the account on My Recipes is entirely separate from the account on our main site, it shares these critical functionalities. Therefore, we extensively tested all important functionalities of this page.

As for our main website, we did not test the Sweepstakes section because it requires entering an email to participate in a sweepstakes, which we avoided to protect our privacy. Similarly, we did not test parts that require entering credit card information for the same reason. However, we tested everything that involves entering an email for verification purposes.

To summarize, we tested all critical functionalities of our main website, along with some additional features. Sensitive data entry, such as email for the sweepstakes or credit card information, was excluded from our tests.

2.2. Testing Environment and Tools

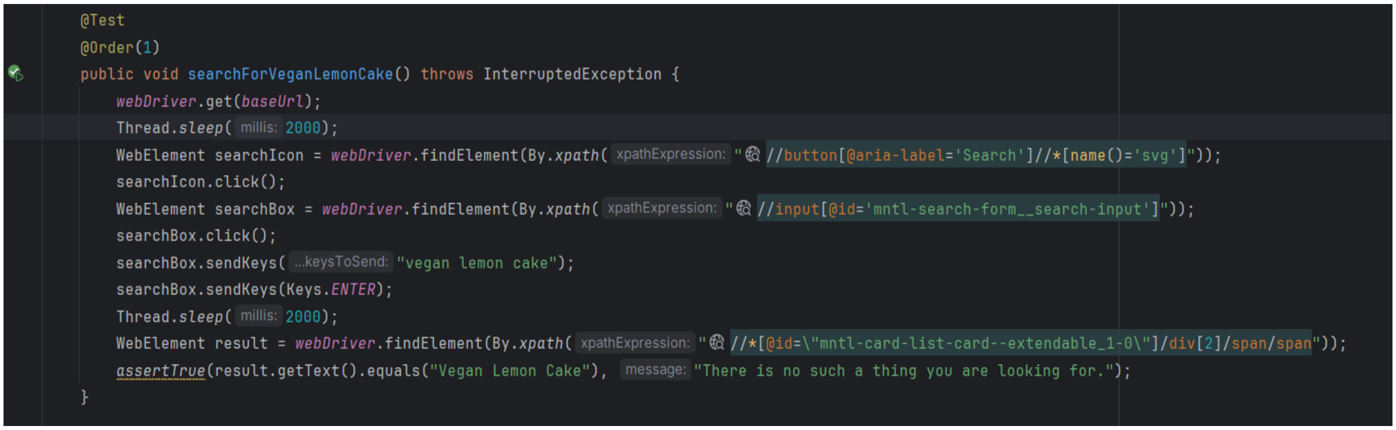
During our testing, we will be using IntelliJ IDEA as the integrated development environment (IDE) and Java as the primary programming language. For test automation, we will utilize Selenium WebDriver, a powerful tool for automating web browsers, enabling us to perform functional and regression testing efficiently. These tools and frameworks will provide a robust environment for developing and executing our test cases.

3. Test Execution

3.1. Test Scenario Name: Search test

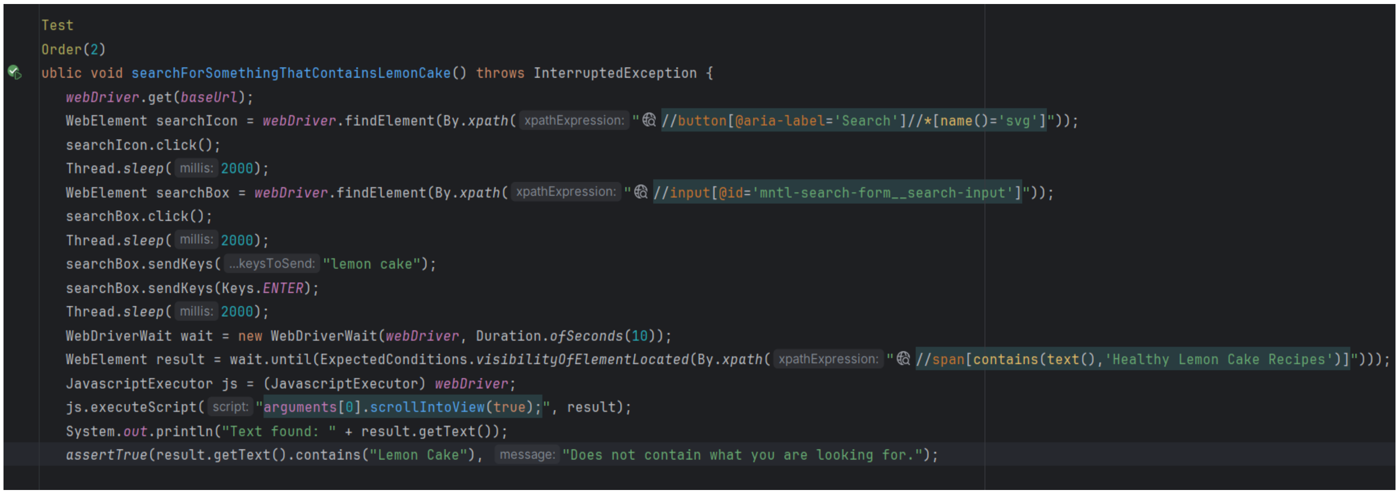
This test scenario is about testing search and its functionalities.

Test Name: Search for “vegan lemon cake”				
Description: Check if searching for “vegan lemon cake” gives you the right result				
Pre-condition(s):				
Test Steps: 1. Wait until page loads. 2. Go to search bar. 3. Click on it. 4. Type “vegan lemon cake”. 5. Click ENTER. 6. Check if the result is equal to “vegan lemon cake” text.	**Test Data:** Input data in input field for searching is “vegan lemon cake”.	**Expected Result:** A user can see the result that is equal to what he/she was looking for.	**Actual Result:** A user can see the result that is equal to what he/she was looking for.	**Status:** PASS
Notes: The page is case-sensitive, so we have to write matching text in the test, exactly like one on the page.				



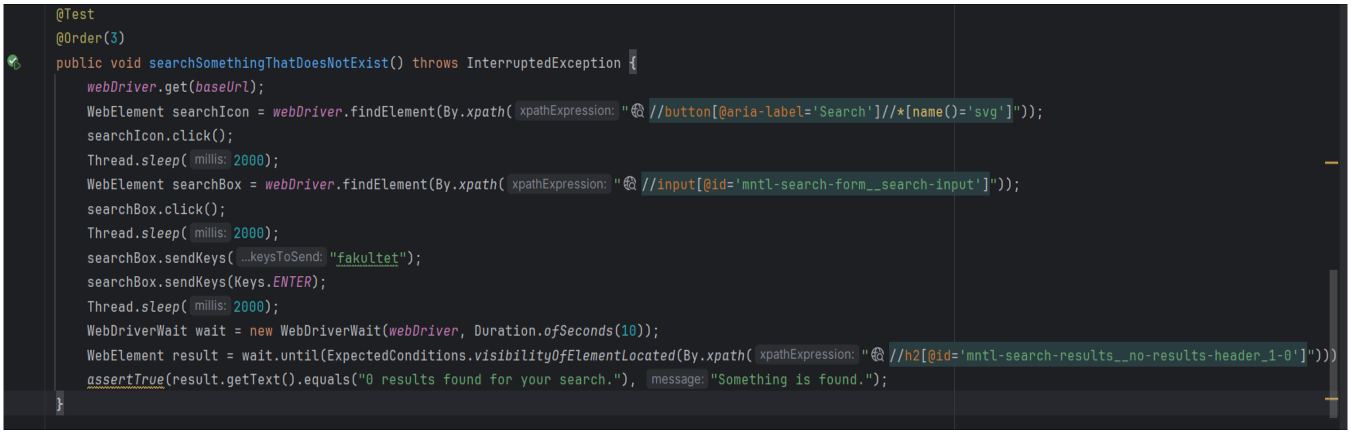
```
@Test
@Order(1)
public void searchForVeganLemonCake() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep( millis: 2000);
    WebElement searchIcon = webDriver.findElement(By.xpath( xpathExpression: "//button[@aria-label='Search']//*[name()='svg']"));
    searchIcon.click();
    WebElement searchBox = webDriver.findElement(By.xpath( xpathExpression: "//input[@id='mntl-search-form__search-input']"));
    searchBox.click();
    searchBox.sendKeys( ...keysToSend: "vegan lemon cake");
    searchBox.sendKeys(Keys.ENTER);
    Thread.sleep( millis: 2000);
    WebElement result = webDriver.findElement(By.xpath( xpathExpression: "//*[@id=\"mntl-card-list-card--extendable_1-0\"]/div[2]/span/span"));
    assertTrue(result.getText().equals("Vegan Lemon Cake"),  message: "There is no such a thing you are looking for.");
}
```

<table>
<tr><td colspan="5"><b>Test Name:</b> Search for something that contains “lemon cake”</td></tr>
<tr><td colspan="5"><b>Description:</b> Check if searching for “lemon cake” gives you the result that also contains “lemon cake”, which should</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b></td></tr>
<tr><td><b>Test Steps:</b>
1. Wait until page loads.
2. Go to search bar.
3. Click on it.
4. Type “vegan lemon cake”.
5. Click ENTER.
6. Wait for visibility of result.
7. Scroll to that result.
8. Check if the result contains “lemon cake” text.</td><td><b>Test Data:</b>
Input data is “lemon cake”.</td><td><b>Expected Result:</b>
A user can see the result that contains what he/she was looking for.</td><td><b>Actual Result:</b>
A user can see the result that contains what he/she was looking for.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b> The page is case-sensitive, so we have to write matching text in the test, exactly like one on the page.</td></tr>
</table>



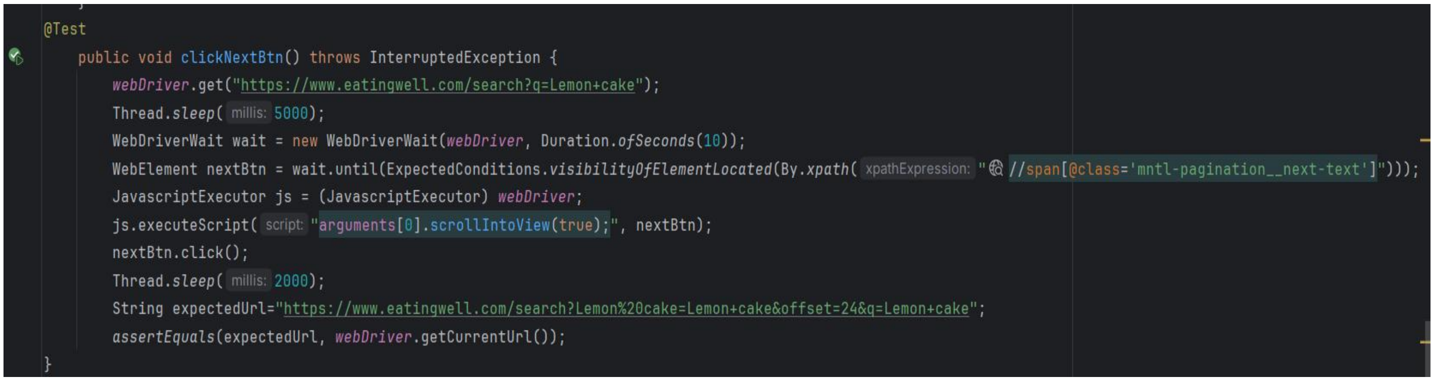
```
Test
Order(2)
ublic void searchForSomethingThatContainsLemonCake() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement searchIcon = webDriver.findElement(By.xpath( xpathExpression: "//button[@aria-label='Search']//*[name()='svg']"));
    searchIcon.click();
    Thread.sleep( millis: 2000);
    WebElement searchBox = webDriver.findElement(By.xpath( xpathExpression: "//input[@id='mntl-search-form__search-input']"));
    searchBox.click();
    Thread.sleep( millis: 2000);
    searchBox.sendKeys( ...keysToSend: "lemon cake");
    searchBox.sendKeys(Keys.ENTER);
    Thread.sleep( millis: 2000);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement result = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//span[contains(text(),'Healthy Lemon Cake Recipes')]")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript( script: "arguments[0].scrollIntoView(true);", result);
    System.out.println("Text found: " + result.getText());
    assertTrue(result.getText().contains("Lemon Cake"),  message: "Does not contain what you are looking for.");
```

<table>
<tr><td colspan="5"><b>Test Name:</b> Search for something that does not exist</td></tr>
<tr><td colspan="5"><b>Description:</b> Check if searching for “fakultet” gives you any result, it shouldn’t.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b> Are there any conditions or other tests that need to be executed before this test; are there any test fixtures? If not, leave this blank.</td></tr>
<tr><td><b>Test Steps:</b>
1. Wait until page loads.
2. Go to search bar.
3. Click on it.
4. Type “ fakultet”.
5. Click ENTER.
6. Wait for visibility of result.
7. Check if the result is equal to “0 results found for your search.” text.</td><td><b>Test Data:</b>
Input data: “fakultet”.</td><td><b>Expected Result:</b>
A user cannot se the result for which he was looking for because it doesn’t exist.</td><td><b>Actual Result:</b>
A user cannot se the result for which he was looking for because it doesn’t exist.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b> Are there any notes about this test you would like to add? If not, leave this blank.</td></tr>
</table>



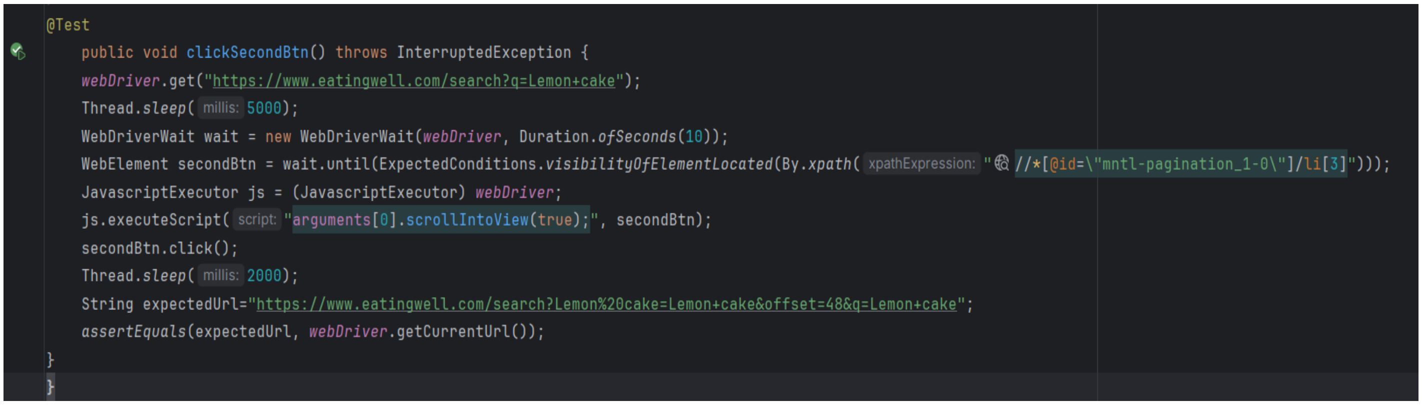
```
@Test
@Order(3)
public void searchSomethingThatDoesNotExist() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement searchIcon = webDriver.findElement(By.xpath( xpathExpression: "//button[@aria-label='Search']//*[name()='svg']"));
    searchIcon.click();
    Thread.sleep( millis: 2000);
    WebElement searchBox = webDriver.findElement(By.xpath( xpathExpression: "//input[@id='mntl-search-form__search-input']"));
    searchBox.click();
    Thread.sleep( millis: 2000);
    searchBox.sendKeys( ...keysToSend: "fakultet");
    searchBox.sendKeys(Keys.ENTER);
    Thread.sleep( millis: 2000);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement result = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//h2[@id='mntl-search-results__no-results-header_1-0']")))
    assertTrue(result.getText().equals("0 results found for your search."),  message: "Something is found.");
}
```

Test Name: Click on the next button				
Description: Check if the next button works and leads to the correct page				
Pre-condition(s): Are there any conditions or other tests that need to be executed before this test; are there any test fixtures? If not, leave this blank.				
Test Steps: 1. Wait until page loads. 2. Wait until finds next button. 3. Scroll to it. 4. Click on it. 5. Check if expected URL is the current one.	**Test Data:**	**Expected Result:** The user is taken to the correct next page when clicking on the next button.	**Actual Result:** The user is taken to the correct next page when clicking on the next button.	**Status:** PASS
Notes: Are there any notes about this test you would like to add? If not, leave this blank.				



```
@Test
    public void clickNextBtn() throws InterruptedException {
        webDriver.get("https://www.eatingwell.com/search?q=Lemon+cake");
        Thread.sleep( millis: 5000);
        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
        WebElement nextBtn = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//span[@class='mntl-pagination__next-text']")));
        JavascriptExecutor js = (JavascriptExecutor) webDriver;
        js.executeScript( script: "arguments[0].scrollIntoView(true);", nextBtn);
        nextBtn.click();
        Thread.sleep( millis: 2000);
        String expectedUrl="https://www.eatingwell.com/search?Lemon%20cake=Lemon+cake&offset=24&q=Lemon+cake";
        assertEquals(expectedUrl, webDriver.getCurrentUrl());
}
```

Test Name: Click on the second(2) button				
Description: Check if the second button works and leads to the correct page				
Pre-condition(s): Are there any conditions or other tests that need to be executed before this test; are there any test fixtures? If not, leave this blank.				
Test Steps: 1. Wait until page loads. 2. Wait until finds the second button. 3. Scroll to it. 4. Click on it. 5. Check if expected URL is the current one.	**Test Data:**	**Expected Result:** The user is taken to the correct second page when clicking on the second button.	**Actual Result:** The user is taken to the correct second page when clicking on the second button.	**Status:** PASS
Notes: By second I mean the button with number "2" at the end of search page.				

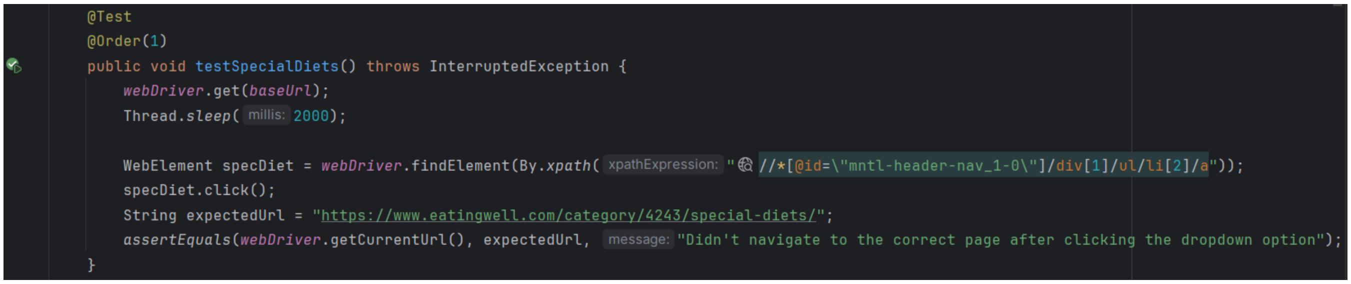


```
@Test
    public void clickSecondBtn() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/search?q=Lemon+cake");
    Thread.sleep( millis: 5000);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement secondBtn = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( xpathExpression: "//*[@id=\"mntl-pagination_1-0\"]/li[3]")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript( script: "arguments[0].scrollIntoView(true);", secondBtn);
    secondBtn.click();
    Thread.sleep( millis: 2000);
    String expectedUrl="https://www.eatingwell.com/search?Lemon%20cake=Lemon+cake&offset=48&q=Lemon+cake";
    assertEquals(expectedUrl, webDriver.getCurrentUrl());
}
}
```

3.2. Test Scenario Name: Navbar links test

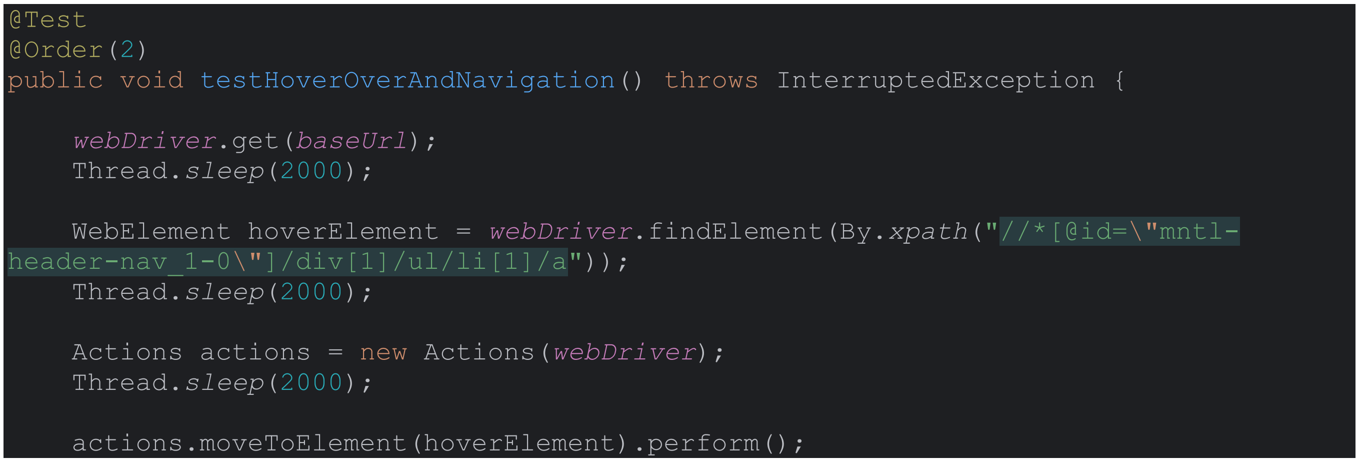
This test scenario is about testing navbar dropdowns, links and their navigation.

Test Name: Test special diets dropdown				
Description: Check if this dropdown link navigates to correct page				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Find and click on dropdown “Special diets”. 3. Check its navigation by comparing current and expected URL.	**Test Data:**	**Expected Result:** The user is taken to the correct page when clicking on the special diets dropdown.	**Actual Result:** The user is taken to the correct page when clicking on the special diets dropdown.	**Status:** PASS
Notes:				



```
@Test
@Order(1)
public void testSpecialDiets() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep( millis: 2000);

    WebElement specDiet = webDriver.findElement(By.xpath( xpathExpression: "//*[@id=\"mntl-header-nav_1-0\"]/div[1]/ul/li[2]/a"));
    specDiet.click();
    String expectedUrl = "https://www.eatingwell.com/category/4243/special-diets/";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl,  message: "Didn't navigate to the correct page after clicking the dropdown option");
}
```

Test Name: Test healthy recipes dropdown option (lunch)				
Description: Check if healthy recipes have dropdown option “lunch” and click on it to check its navigation.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for page to load. 2. Find the element to hover over. 3. Hover over the element. 4. Wait until dropdown menu appears. 5. Verify the option is visible. 6. Click on the dropdown option. 7. Check its navigation by comparing current and expected URL.		The user is taken to the correct page when hovered over healthy recipes and clicked on the lunch in the dropdown.	The user is taken to the correct page when hovered over healthy recipes and clicked on the lunch in the dropdown.	PASS
Notes:				



```
@Test
@Order(2)
public void testHoverOverAndNavigation() throws InterruptedException {

    webDriver.get(baseUrl);
    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//*[@id=\"mntl-
header-nav_1-0\"]/div[1]/ul/li[1]/a"));
    Thread.sleep(2000);

    Actions actions = new Actions(webDriver);
    Thread.sleep(2000);

    actions.moveToElement(hoverElement).perform();
    // Wait until the dropdown menu appears
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//li[@class='mnt
l-header-nav__sublist-item']//a[contains(text(),'Lunch')]")));

    // Verify the option is visible
    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
    Thread.sleep(2000);

    dropdownOption.click();
    Thread.sleep(2000);

    String expectedUrl =
"https://www.eatingwell.com/recipes/17963/mealtimes/lunch/";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the
correct page after clicking the dropdown option");
}
```

Test Name: Test “view all” button and visibility of “dinner plans” option				
Description: Check if “view all” button when clicked, makes “dinner plans” option visible				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Find and click the “Meal Plans” button. 3. Wait for the “View all” button to become clickable. 4. Scroll the element into view. 5. Click on the button. 6. Wait for the additional options to become visible. 7. Check if “dinner plans” option is visible.	**Test Data:**	**Expected Result:** The user can see “dinner plans” option by clicking on button “View all”.	**Actual Result:** The user can see “dinner plans” option by clicking on button “View all”.	**Status:** PASS
Notes:				

```
@Test
@Order(3)
public void testViewAllButtonAndDinnerPlansOption() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement mealPlansBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-
header-nav_1-0\"]/div[1]/ul/li[4]/a"));
    mealPlansBtn.click();
    Thread.sleep(2000);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement viewAllBtn =
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id=\"mntl-
taxonomy-nodes__chop-text_1-0\"]")));


    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", viewAllBtn);
    Thread.sleep(2000);

    js.executeScript("arguments[0].click();", viewAllBtn);
    Thread.sleep(2000);

    WebElement dinnerPlansOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"mntl-
taxonomy-nodes__link_20-0\"]/span")));

    assertTrue(dinnerPlansOption.isDisplayed(), "Dinner plan option is not
visible");
}
```

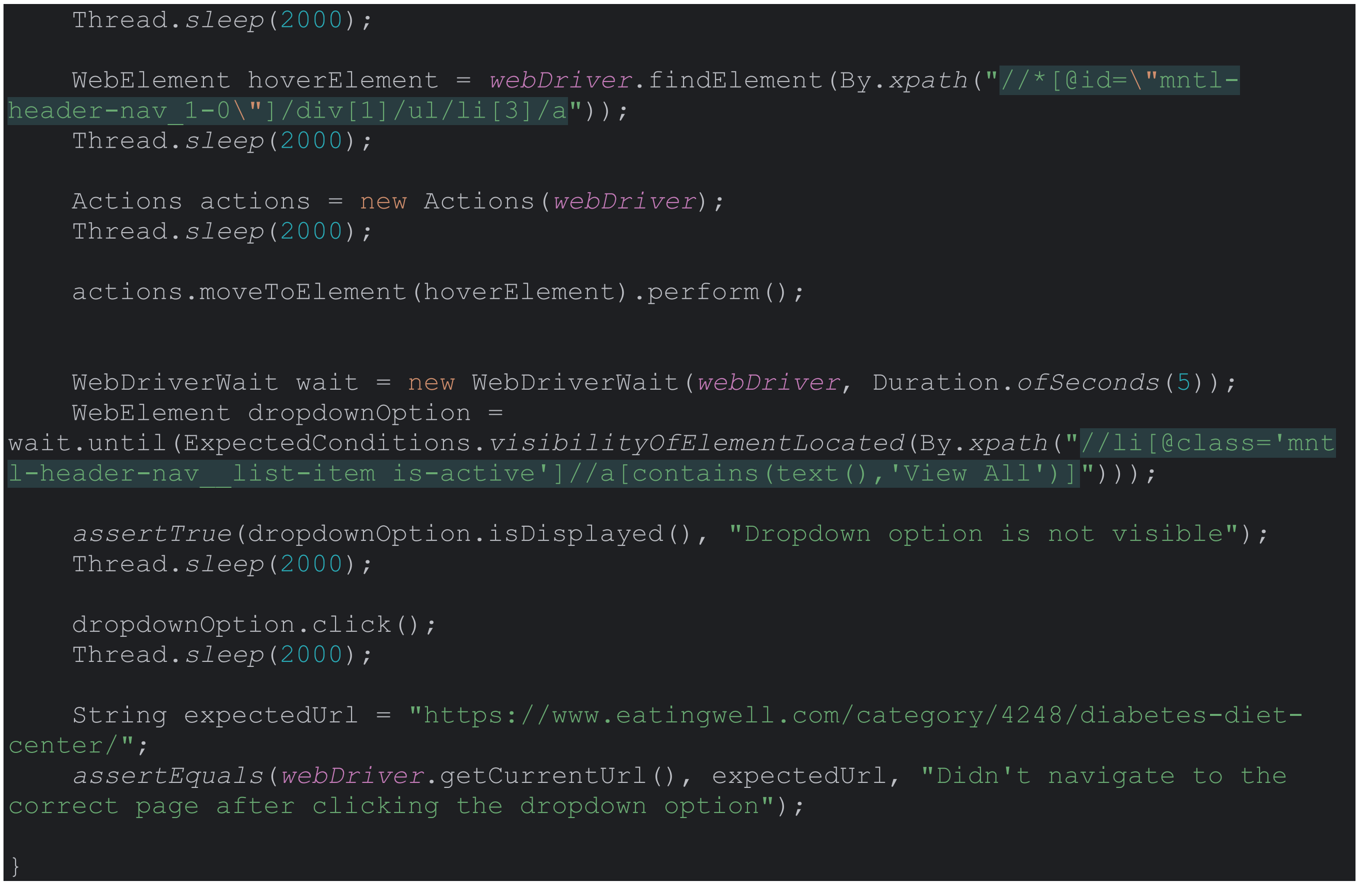
Test Name: Test “healthy eating for kids” navigation				
Description: Check if “healthy eating for kids” link when clicked, goes to expected url				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for page to load. 2. Find and click the “Healthy eating” in navbar. 3. Click on the button. 4. Find “healthy eating for kids” option. 5. Click on it. 6. Check if expected and current url are the same.		The user is sent to the correct page when he clicked on “healthy eating for kids” option.	The user is sent to the correct page when he clicked on “healthy eating for kids” option.	PASS
Notes:				

```
@Test
@Order(4)
```

```
public void healthyEatingForKidsNavigation() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);

    WebElement healthyEatingBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-header-nav_1-0\"]/div[1]/ul/li[6]/a"));
    healthyEatingBtn.click();

    WebElement healthyEatingForKids =
webDriver.findElement(By.xpath("//*[@id=\"mntl-taxonomy-nodes__link_4-0\"]/span"));
    healthyEatingForKids.click();
    String expectedUrl = "https://www.eatingwell.com/category/4311/healthy-eating-for-kids/";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the correct page after clicking this option");

}
```

Test Name: Test “view all” option’s navigation in “diabetes” dropdown				
Description: Check if “view all” option’s navigation in “diabetes” dropdown is correct				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Find the element to hover over. 3. Hover over the element. 4. Wait until dropdown menu appears. 5. Verify the option is visible. 6. Click on the dropdown option. 7. Check its navigation by comparing current and expected url.	**Test Data:**	**Expected Result:** The user is taken to the correct page when hovered over “diabetes” and clicked on the “view all” in the dropdown.	**Actual Result:** The user is taken to the correct page when hovered over “diabetes” and clicked on the “view all” in the dropdown.	**Status:** PASS
Notes:				

```
@Test
@Order(5)
public void testHoverOverAndClickViewAllNavigation() throws InterruptedException {

    webDriver.get(baseUrl);
```



```
    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//*[@id=\"mntl-
header-nav_1-0\"]/div[1]/ul/li[3]/a"));
    Thread.sleep(2000);

    Actions actions = new Actions(webDriver);
    Thread.sleep(2000);

    actions.moveToElement(hoverElement).perform();


    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//li[@class='mnt
l-header-nav__list-item is-active']//a[contains(text(),'View All')]")));

    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
    Thread.sleep(2000);

    dropdownOption.click();
    Thread.sleep(2000);

    String expectedUrl = "https://www.eatingwell.com/category/4248/diabetes-diet-
center/";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the
correct page after clicking the dropdown option");

}
```

Test Name: Test meal plans dropdown option (high-protein)				
Description: Check if meal plans have dropdown option “high-protein” and click on it to check its navigation.				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Find the element to hover over. 3. Hover over the element. 4. Wait until dropdown menu appears. 5. Verify the option is visible. 6. Click on the dropdown option.	**Test Data:**	**Expected Result:** The user is taken to the correct page when hovered over meal plans and clicked on the high-protein in the dropdown.	**Actual Result:** The user is taken to the correct page when hovered over meal plans and clicked on the high-protein in the dropdown.	**Status:** PASS

7. Check its navigation by comparing current and expected url.				

Notes:

```
@Test
@Order(6)
public void testHoverOverAndClickNavigation1() throws InterruptedException {

    webDriver.get(baseUrl);
    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//*[@id=\"mntl-header-nav_1-0\"]/div[1]/ul/li[4]/a"));
    Thread.sleep(2000);

    Actions actions = new Actions(webDriver);
    Thread.sleep(2000);

    actions.moveToElement(hoverElement).perform();


    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//li[@class='mntl-header-nav__list-item is-active']//a[contains(text(),'High-Protein')]")));


    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
    Thread.sleep(2000);

    dropdownOption.click();
    Thread.sleep(2000);

    String expectedUrl = "https://www.eatingwell.com/high-protein-meal-plans-8660078";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the correct page after clicking the dropdown option");
}
```

Test Name: Test FB link and its navigation on meal plans				
Description: Check if facebook link when clicked, goes to official fb login page				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

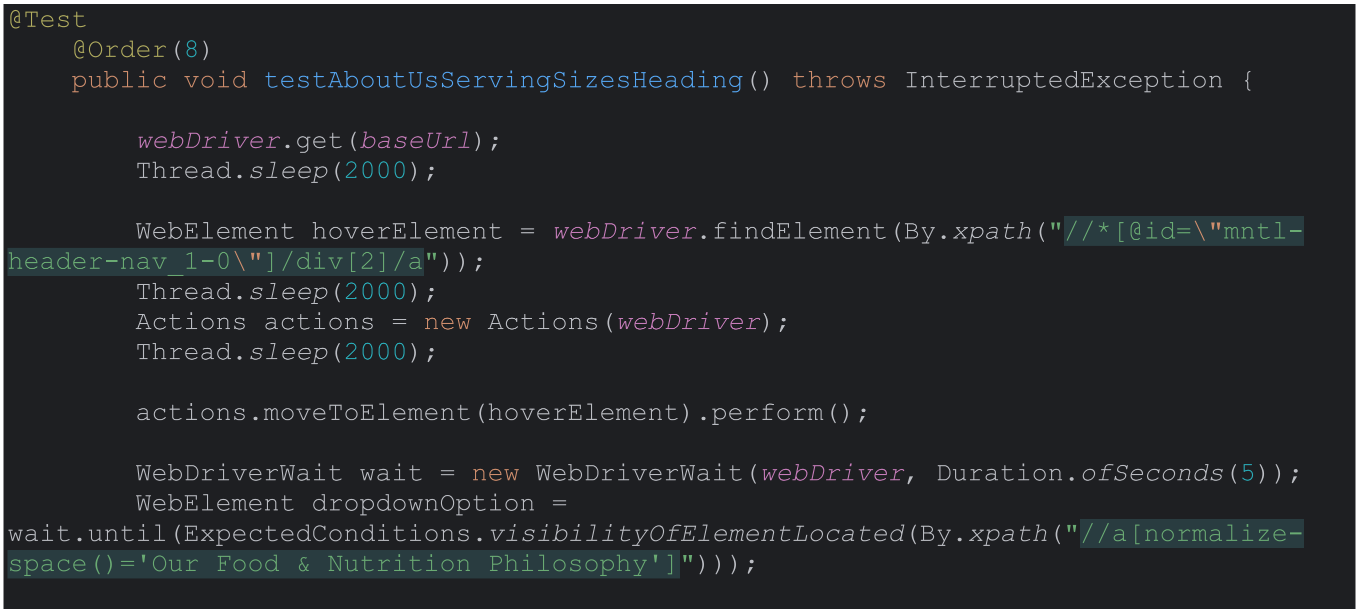
1. Wait for page to load. 2. Find the element to hover over. 3. Hover over the element. 4. Wait until dropdown menu appears. 5. Verify the option is visible. 6. Click on the dropdown option. 7. Find and click on fb link element. 8. Handle windows switching 9. Check its navigation by comparing current URL in new window and expected URL. 10. Switch back to the original window.		The user must be directed to the fb signup/login page after clicking on fb link on meal plans page.	The user is directed to the fb signup/login page after clicking on fb link on meal plans page.	PASS

Notes:

```
@Test
@Order(7)
public void testLinkFbOnMealPlans() throws InterruptedException {

    webDriver.get(baseUrl);
    Thread.sleep(2000);


    WebElement hoverElement = webDriver.findElement(By.xpath("//*[@id=\"mntl-header-nav_1-0\"]/div[1]/ul/li[4]/a"));
    Thread.sleep(2000);


    Actions actions = new Actions(webDriver);
    Thread.sleep(2000);


    actions.moveToElement(hoverElement).perform();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//li[@class='mnt
l-header-nav__list-item is-active']//a[contains(text(),'High-Protein')]")));


    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
    Thread.sleep(2000);


    dropdownOption.click();
    Thread.sleep(2000);


    WebElement fbLink = webDriver.findElement(By.xpath("//*[@id=\"taxonomysc-
header__social_1-0\"]/li[2]/span"));
    fbLink.click();
    Thread.sleep(2000);

    // Store the current window handle (original window)
    String originalWindow = webDriver.getWindowHandle();

    // Get all window handles
    Set<String> windowHandles = webDriver.getWindowHandles();

    // Iterate through the window handles and switch to the new window (tab)
    for (String handle : windowHandles) {
        if (!handle.equals(originalWindow)) {
            webDriver.switchTo().window(handle);
            break;
        }
    }

    // Verify the URL in the new window
    String expectedUrl =
"https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.eatingwell.com%2Fhi
gh-protein-meal-plans-
8660078%3Futm_source%3Dfacebook.com%26utm_medium%3Dsocial%26utm_campaign%3Dsocial-
share-article";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the
correct page in the new window");

    // Switch back to the original window
webDriver.switchTo().window(originalWindow);
    }
```

Test Name: Test Serving sizes heading in About us in navbar				
Description: Check if article in About us has “Serving sizes” section which has heading “Serving sizes”				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for page to load. 2. Find the element to hover over. 3. Hover over the element. 4. Wait until dropdown menu appears. 5. Verify the option is visible. 6. Click on the dropdown option. 7. Find and click on Serving sizes content section element. 8. Find heading element of Serving sizes section. 9. Check if heading of “Serving sizes” section is “Serving sizes”.		Serving sizes is actually a heading of that section that has the same name as heading, so based on that, we know that we are successfully directed to the content of that section.	We are indeed successfully directed because a heading of Serving sizes section is indeed serving sizes.	PASS
Notes:				



```
@Test
    @Order(8)
    public void testAboutUsServingSizesHeading() throws InterruptedException {

        webDriver.get(baseUrl);
        Thread.sleep(2000);

        WebElement hoverElement = webDriver.findElement(By.xpath("//*[@id=\"mntl-header-nav_1-0\"]/div[2]/a"));
        Thread.sleep(2000);
        Actions actions = new Actions(webDriver);
        Thread.sleep(2000);

        actions.moveToElement(hoverElement).perform();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(5));
        WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//a[normalize-space()='Our Food & Nutrition Philosophy']")));
```

```
        assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
        Thread.sleep(2000);

        dropdownOption.click();
        Thread.sleep(2000);

        WebElement servingSizes=
webDriver.findElement(By.xpath("//span[@class='link__wrapper'][normalize-
space()='Serving Sizes']"));
        servingSizes.click();
        Thread.sleep(2000);
        WebElement servingSizesHeading=
webDriver.findElement(By.xpath("//span[@class='mntl-sc-block-
heading__text'][normalize-space()='Serving Sizes']"));
        Thread.sleep(2000);
        assertTrue(servingSizesHeading.getText().equals("Serving Sizes"),"There is
no this heading here." );
    }
}
```

3.3. Test Scenario Name: Subscription test

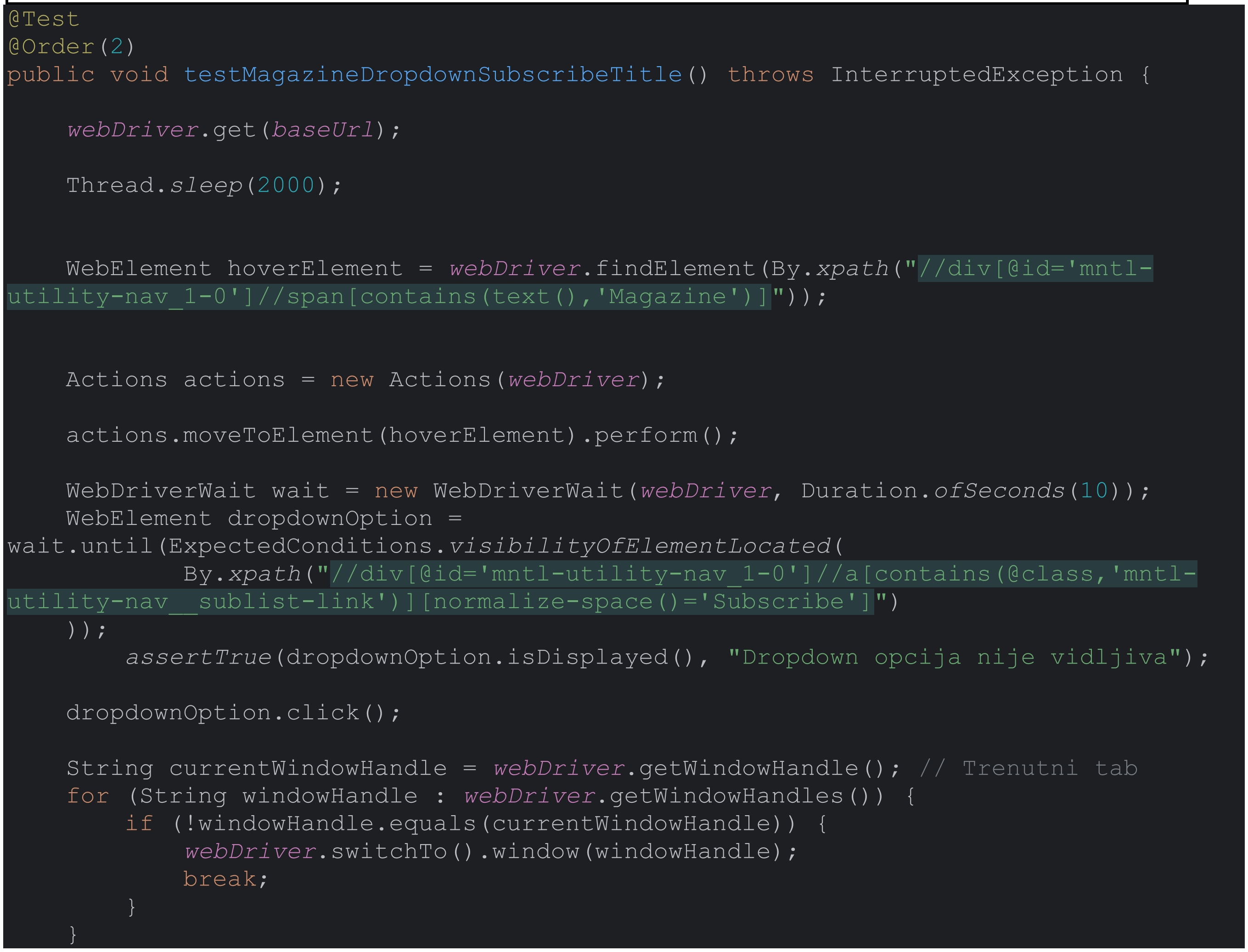
This test scenario is about testing all subscribe links and their navigation, including checking input fields in their content.

Test Name: Test subscribe option in magazine dropdown				
Description: Check if this dropdown link navigates to correct page				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Hover over dropdown “Magazine”. 3. Wait for visibility of dropdown menu. 4. Check if option is displayed. 5. Click on option. 6. Handle windows switching. 7. Check its navigation by comparing current URL in new window and expected URL.	**Test Data:**	**Expected Result:** The user is taken to the correct page when clicking on the subscribe option in magazine dropdown.	**Actual Result:** The user is taken to the correct page when clicking on the subscribe option in magazine dropdown.	**Status:** PASS
Notes:				

```
@Test
@Order(1)
public void testMagazineDropdownSubscribeNavigation() throws InterruptedException {

    webDriver.get(baseUrl);

    Thread.sleep(2000);


    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-utility-nav_1-0']//span[contains(text(),'Magazine')]"));


    Actions actions = new Actions(webDriver);


    actions.moveToElement(hoverElement).perform();


    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption = wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));

    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");


    dropdownOption.click();

        String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }


    String expectedUrl = "https://www.magazines.com/eatingwell-magazine.html?utm_source=eatingwell.com&utm_medium=owned&utm_campaign=ad409etr1w3226b";

    assertEquals(expectedUrl, webDriver.getCurrentUrl(), "Page is not properly directed after clicking on 'Subscribe'");
}
```

Test Name: Test title in content of subscribe option in magazine dropdown
Description: Check if this dropdown link navigates to correct page and then test title of its content.
Pre-condition(s):

Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for page to load. 2. Hover over dropdown “Magazine”. 3. Wait for visibility of dropdown menu. 4. Check if option is displayed. 5. Click on option. 6. Handle windows switching. 7. Find title of its content. 8. Check if title really is: “EatingWell Magazine Subscription”		The title is “EatingWell Magazine Subscription” so user is taken to the correct page when clicking on the subscribe option in magazine dropdown.	The title is indeed “EatingWell Magazine Subscription” so user is taken to the correct page when clicking on the subscribe option in magazine dropdown.	PASS
Notes:				

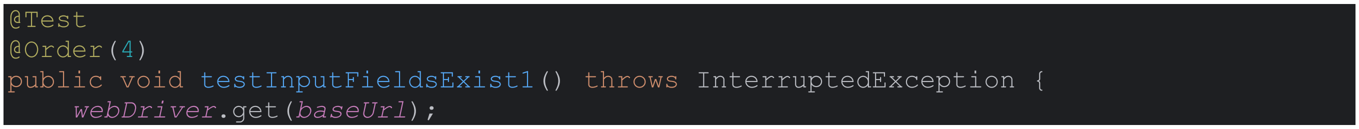


```
@Test
@Order(2)
public void testMagazineDropdownSubscribeTitle() throws InterruptedException {

    webDriver.get(baseUrl);

    Thread.sleep(2000);


    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));


    Actions actions = new Actions(webDriver);

    actions.moveToElement(hoverElement).perform();

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));
        assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");

    dropdownOption.click();

    String currentWindowHandle = webDriver.getWindowHandle(); // Trenutni tab
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement subscriptionTitle =
webDriver.findElement(By.xpath("//h1[@class='productTitle']"));
    assertEquals("EatingWell Magazine Subscription", subscriptionTitle.getText(),
"Title text doesn't match");
}
```

Test Name: Test checkout button				
Description: Check if heading for checkout button is present when button is clicked.				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Hover over dropdown “Magazine”. 3. Wait for visibility of dropdown menu. 4. Check if option is displayed. 5. Click on option. 6. Handle windows switching. 7. Wait, scroll and find checkout button. 8. Wait for it to be clickable. 9. Click on it. 10. Find its heading. 11. Check if clicked, it directs the user to the checkout form with “checkout heading”.	**Test Data:**	**Expected Result:** The title is “Checkout” so user is taken to the correct page with form when clicking on the checkout button from subscribe option in magazine dropdown.	**Actual Result:** The title is “Checkout” so user is taken to the correct page with form when clicking on the checkout button from subscribe option in magazine dropdown.	**Status:** PASS
Notes:				

```
@Test
@Order(3)
public void testMagazineDropdownCheckoutBtn() throws InterruptedException {

    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));

        Actions actions = new Actions(webDriver);

        actions.moveToElement(hoverElement).perform();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(15));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));


    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible.");

    dropdownOption.click();

    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {

            webDriver.switchTo().window(windowHandle);
            break;
        }
    }

    WebElement checkoutButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//button[@aria-label='Checkout with $20.00
selection']//span[@class='checkoutBtnTxt'][normalize-space()='CHECKOUT']")
    ));

        assertTrue(checkoutButton.isDisplayed(), "Button ‘checkout’ is not visible
on the page.");
    Thread.sleep(2000);

    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", checkoutButton);

        wait.until(ExpectedConditions.elementToBeClickable(checkoutButton));


    checkoutButton.click();
    Thread.sleep(3000);

    WebElement checkoutHeading =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"storeC
heckoutForm\"]/h2")));
    assertTrue(checkoutHeading.getText().equals("CHECKOUT"), "There is no heading
‘Checkout’ on the page.");
}
```

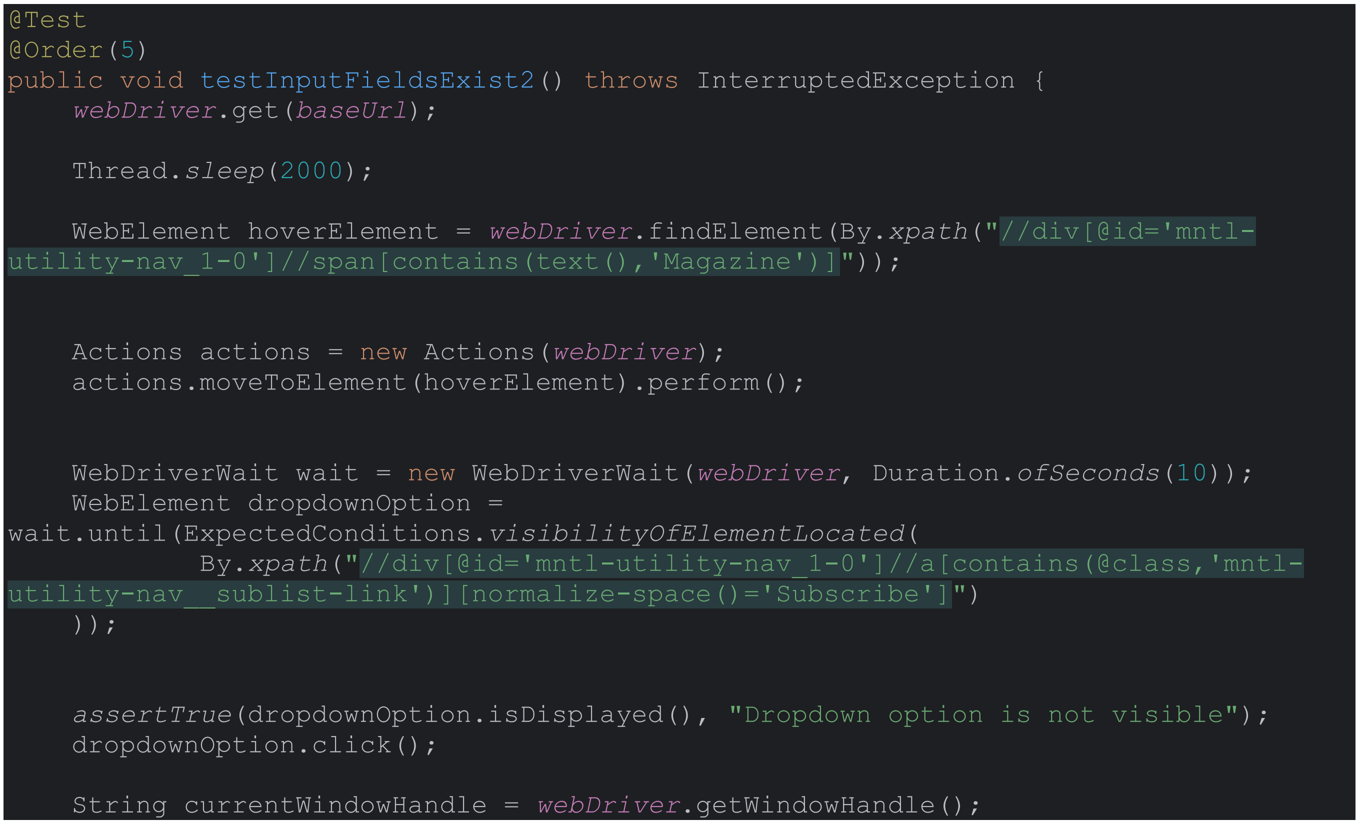
Test Name: Test input field in checkout form				
Description: Check if input field “Card information” is displayed after clicking checkout button in checkout form.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for page to load. 2. Hover over dropdown “Magazine”. 3. Wait for visibility of dropdown menu. 4. Check if option is displayed. 5. Click on option. 6. Handle windows switching. 7. Wait, scroll and find checkout button. 8. Wait for it to be clickable. 9. Click on it. 10. Find “Card information” input field. 11. Check if that input field is actually displayed.		The input field “Card information” is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	The input field “Card information” is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	PASS
Notes:				



```
@Test
@Order(4)
public void testInputFieldsExist1() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));

    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));

    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible.");

    dropdownOption.click();

    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }

    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-
label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-
space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);

    WebElement cardInformationField =
webDriver.findElement(By.xpath("//p[@class='checkout__international']"));

    assertTrue(cardInformationField.isDisplayed(), "Card Information input field is
not displayed on page.");

}
```

Test Name: Test input field in checkout form				
Description: Check if input field “Email address” is displayed after clicking checkout button in checkout form.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

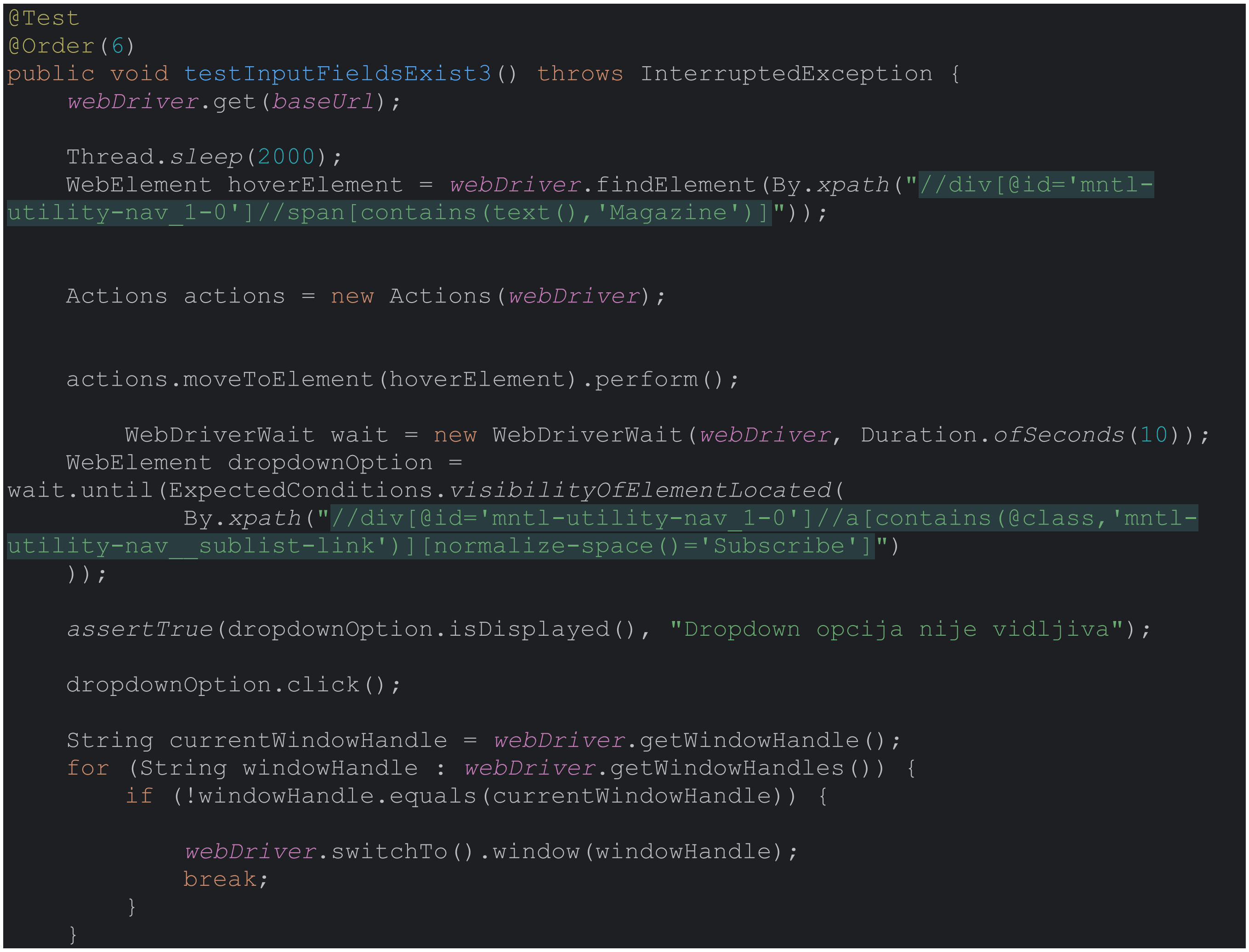
12. Wait for page to load. 13. Hover over dropdown “Magazine”. 14. Wait for visibility of dropdown menu. 15. Check if option is displayed. 16. Click on option. 17. Handle windows switching. 18. Wait, scroll and find checkout button. 19. Wait for it to be clickable. 20. Click on it. 21. Find “Email address” input field. 22. Check if that input field is actually displayed.		The input field “Email address” is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	The input field “Email address” is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	PASS
Notes:				



```
@Test
@Order(5)
public void testInputFieldsExist2() throws InterruptedException {
    webDriver.get(baseUrl);

    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));


    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();


    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));


    assertTrue(dropdownOption.isDisplayed(), "Dropdown option is not visible");
    dropdownOption.click();

    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {

            webDriver.switchTo().window(windowHandle);
            break;
        }
    }


    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-
label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-
space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);
    WebElement emailAddress =
webDriver.findElement(By.xpath("//input[@id='bill_to_email']"));

    assertTrue(emailAddress.isDisplayed(), "Email address input field is not
displayed on page.");


}
```

Test Name: Test input field in checkout form				
Description: Check if input field "Enter number" is displayed after clicking checkout button in checkout form.				
Pre-condition(s):				
Test Steps: 23. Wait for page to load. 24. Hover over dropdown "Magazine". 25. Wait for visibility of dropdown menu. 26. Check if option is displayed. 27. Click on option. 28. Handle windows switching. 29. Wait, scroll and find checkout button. 30. Wait for it to be clickable. 31. Click on it. 32. Find and scroll to "Enter number" input	**Test Data:**	**Expected Result:** The input field "Enter number" is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	**Actual Result:** The input field "Enter number" is displayed when clicking on the checkout button from subscribe option in magazine dropdown.	**Status:** PASS

field. 33. Check if that input field is actually displayed.				
Notes:				

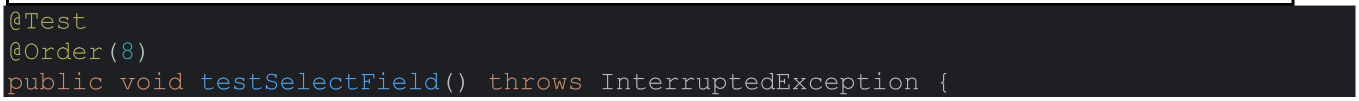


```
@Test
@Order(6)
public void testInputFieldsExist3() throws InterruptedException {
    webDriver.get(baseUrl);

    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));


    Actions actions = new Actions(webDriver);


    actions.moveToElement(hoverElement).perform();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));

    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");

    dropdownOption.click();

    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {

            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);

    WebElement enterNumberField = webDriver.findElement(By.xpath("//input[@id='giftCardCode']"));

    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", enterNumberField);

    assertTrue(enterNumberField.isDisplayed(), "Enter number input field nije prisutan na stranici");


}
```

Test Name: Test subscribe linked button				
Description: Check if you go to the correct page by clicking on subscribe linked button.				
Pre-condition(s):				
Test Steps: 1. Wait for page to load. 2. Find a subscribe button. 3. Click on that button. 4. Handle windows switching. 5. Check if that URL in new window is expected URL.	**Test Data:**	**Expected Result:** By clicking on subscribe button it goes to the expected URL in new window.	**Actual Result:** By clicking on subscribe button it goes to the expected URL in new window.	**Status:** PASS
Notes:				

```
@Test
@Order(7)
public void testLinkedSubscribeButton() throws InterruptedException {

    webDriver.get(baseUrl);
    Thread.sleep(3000);

    WebElement subscribeButton = webDriver.findElement(By.xpath("//span[normalize-space()='Subscribe']"));
    subscribeButton.click();
    Thread.sleep(3000);

    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {

            webDriver.switchTo().window(windowHandle);
            break;
        }
    }

    String expectedUrl = "https://www.magazines.com/eatingwell-
magazine.html?utm_source=eatingwell.com&utm_medium=owned&utm_campaign=ad409etr1w322
6b";
    assertEquals(expectedUrl, webDriver.getCurrentUrl(), "This is not the right URL
in the new tab.");

    webDriver.switchTo().window(currentWindowHandle);
}
```

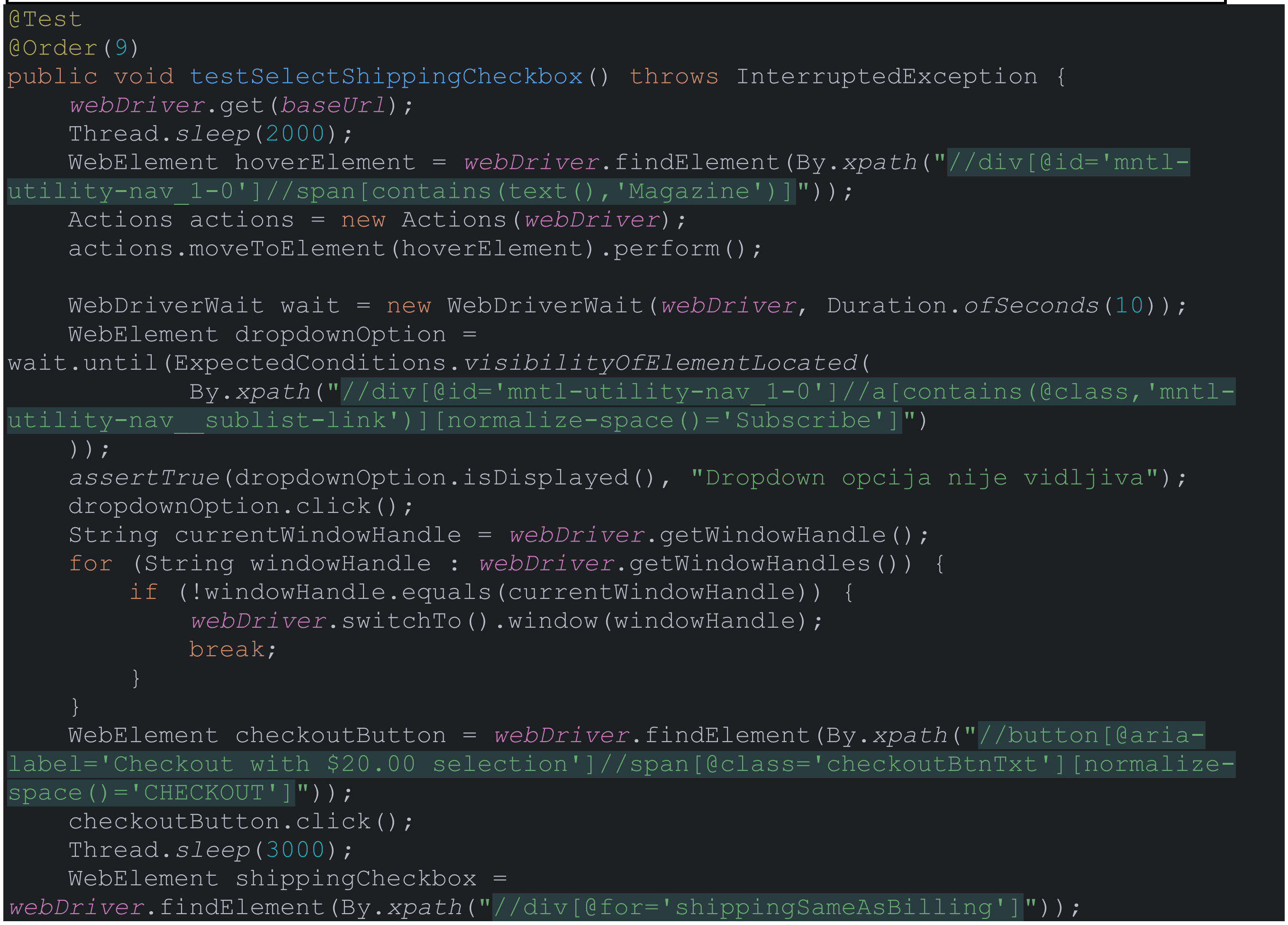
Test Name: Test select field				
Description: Check if the "Colorado" state is correctly selected from the billing address dropdown on the checkout page.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for page to load. 2. Find and hover over “Magazine” menu. 3. Wait for the dropdown menu to appear. 4. Verify that the “Subscribe” option in the dropdown is visible. 5. Click on the “Subscribe” button. 6. Switch to the newly opened tab. 7. Locate and click the “Checkout” button. 8. Wait for the checkout page to load. 9. Select “Colorado” from the “State” dropdown. 10. Check if the selected state really is “Colorado”.		By clicking on the "Subscribe" button, it opens the expected URL in a new window, and "Colorado" is successfully selected in the dropdown.	By clicking on the "Subscribe" button, it opens the expected URL in a new window, and "Colorado" is successfully selected in the dropdown.	PASS
Notes:				



```
@Test
@Order(8)
public void testSelectField() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);
    Select stateDropdown = new
Select(webDriver.findElement(By.xpath("//select[@id='billingAddressState']")));
    stateDropdown.selectByVisibleText("Colorado");
    Thread.sleep(3000);
    WebElement selectedOption = stateDropdown.getFirstSelectedOption();
    assertTrue(selectedOption.getText().equals("Colorado"), "State 'Colorado' was not selected.");
}
```

Test Name: Test select shipping checkbox				
Description: Check if the "Shipping Same As Billing" checkbox is visible on the checkout page and remains unselected after clicking.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find and hover over “Magazine” menu. 3. Wait for the dropdown menu to appear. 4. Verify that the “Subscribe” option in	**Test Data:**	**Expected Result:** By clicking the "Subscribe" button, it opens the expected URL in a new window, and the "Shipping Same As Billing"	**Actual Result:** By clicking the "Subscribe" button, it opens the expected URL in a new window, and the "Shipping Same As Billing" checkbox is visible	**Status:** PASS

the dropdown is visible. 5. Click on the "Subscribe" button. 6. Switch to the newly opened tab. 7. Locate and click the "Checkout" button. 8. Wait for the checkout page to load. 9. Locate the "Shipping Same as Billing" checkbox and scroll it into view and click on the checkbox. 10. Verify that the checkbox is not selected after clicking.		checkbox is visible but remains unselected after clicking.	but remains unselected after clicking.	
Notes:				



```
@Test
@Order(9)
public void testSelectShippingCheckbox() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-
label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-
space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);
    WebElement shippingCheckbox =
webDriver.findElement(By.xpath("//div[@for='shippingSameAsBilling']"));
```

```
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", shippingCheckbox);
    Thread.sleep(3000);
    shippingCheckbox.click();
    Thread.sleep(3000);
    assertFalse(shippingCheckbox.isSelected(), "Checkbox was selected.");
}
```

Test Name: Test order subtotal				
Description: Check if the subtotal price on the checkout page is correctly displayed as "$20.00".				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find and hover over “Magazine” menu. 3. Wait for the dropdown menu to appear. 4. Ensure the “Subscribe” button in the dropdown is visible and click on it. 5. Switch to the newly opened tab. 6. Locate and click the “Checkout” button. 7. Wait for the checkout page to load. 8. Locate the “Subtotal” element and scroll it into view using JavaScript. 9. Verify that he subtotal text matches the expected value of “$20.00”.	**Test Data:**	**Expected Result:** By clicking the "Subscribe" button, it opens the expected URL in a new window, and the subtotal price on the checkout page is displayed as "$20.00".	**Actual Result:** By clicking the "Subscribe" button, it opens the expected URL in a new window, and the subtotal price on the checkout page is displayed as "$20.00".	**Status:** PASS

Notes:

```
@Test
@Order(10)
public void testOrderSubtotal() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));

    Actions actions = new Actions(webDriver);

    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Subscribe']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement checkoutButton = webDriver.findElement(By.xpath("//button[@aria-
label='Checkout with $20.00 selection']//span[@class='checkoutBtnTxt'][normalize-
space()='CHECKOUT']"));
    checkoutButton.click();
    Thread.sleep(3000);
    WebElement subtotal =
webDriver.findElement(By.xpath("//*[@id=\"storeCheckoutForm\"]/div[1]/div[2]/div[3]
/span[2]"));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", subtotal);
    Thread.sleep(3000);
    String subtotalText = subtotal.getText();
    assertEquals("$20.00", subtotalText, "Total price is not equal.");
}
```

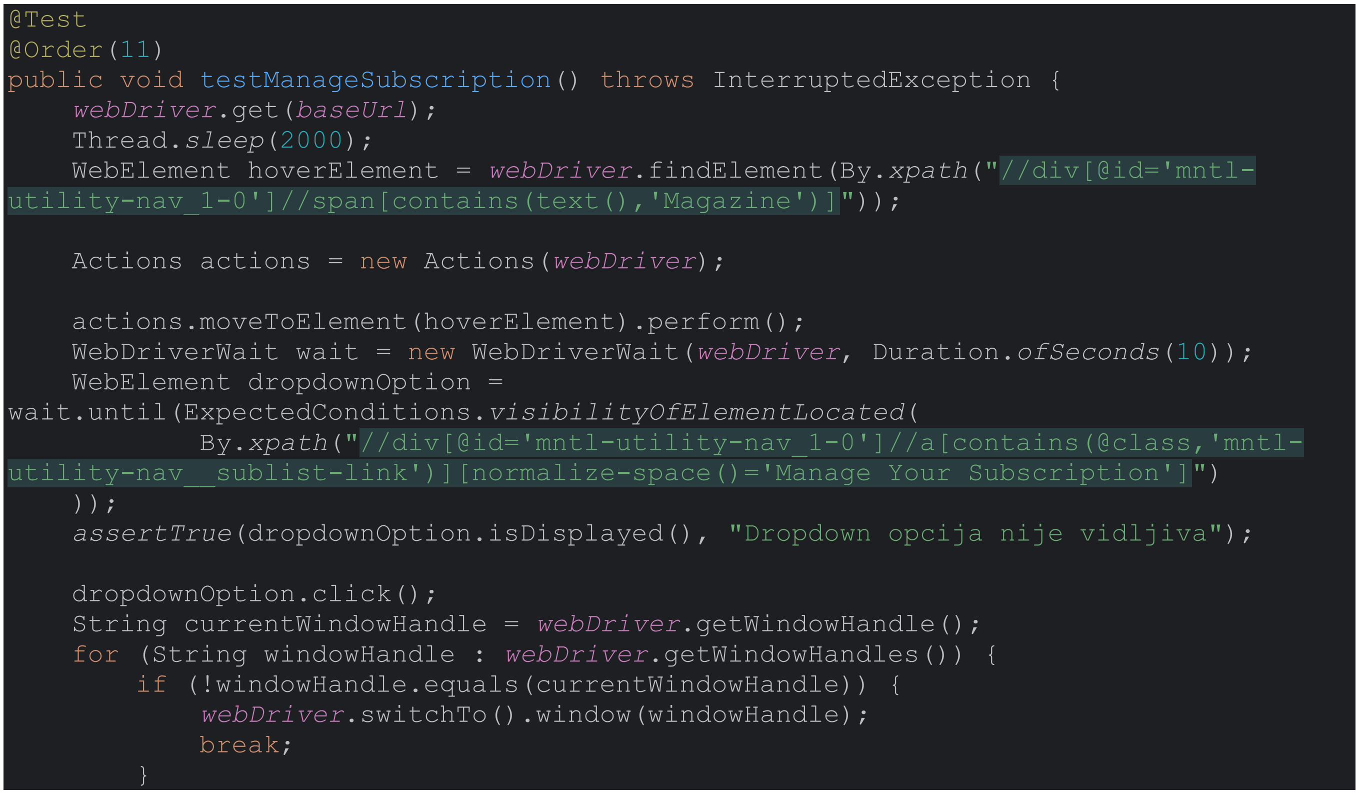
Test Name: Test manage subscription

Description: Check if the "Manage Your Subscription" page opens in a new tab and displays the correct headings for login options.

Pre-condition(s):

Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
1. Wait for the page to load. 2. Hover over the “Magazine” menu.		On clicking the "Manage Your	On clicking the "Manage Your Subscription"	PASS

3. Wait for the "Manage Your Subscription" dropdown option to appear. 4. Verify the dropdown option is visible and click the it. 5. Switch to the newly opened tab. 6. Verify the Login Using Your Account Number and ZIP code heading is displayed. 7. Verify the Login Using Your Name and Address heading is displayed.		Subscription" option, a new tab opens displaying the correct headings for login options.	option, a new tab opens displaying the correct headings for login options.	
Notes:				



```
@Test
@Order(11)
public void testManageSubscription() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));

    Actions actions = new Actions(webDriver);

    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Manage Your Subscription']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");

    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement h1 = webDriver.findElement(By.xpath("//h1[normalize-space()='Login 
Using Your Account Number and ZIP Code']"));
    Thread.sleep(3000);
    WebElement h2 = webDriver.findElement(By.xpath("//h1[normalize-space()='Login 
Using Your Name and Address']"));
    assertTrue(h1.isDisplayed(), "Heading nije prisutan na stranici");
    assertTrue(h2.isDisplayed(), "Heading nije prisutan na stranici");
}
```

Test Name: Test give a gift- link on the manage subscription page				
Description: Check if the "Give a Gift" link on the "Manage Your Subscription" page redirects to the correct URL.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Hover over the “Magazine” menu. 3. Wait for the “Manage Your Subscription” option to appear and click it. 4. Switch to the new tab. 5. Locate and click the “Give a Gift” link. 6. Verify the redirected URL matches the expected URL.	**Test Data:**	**Expected Result:** By clicking the "Give a Gift" link, the user is redirected to the expected URL.	**Actual Result:** By clicking the "Give a Gift" link, the user is redirected to the expected URL.	**Status:** PASS
Notes:				

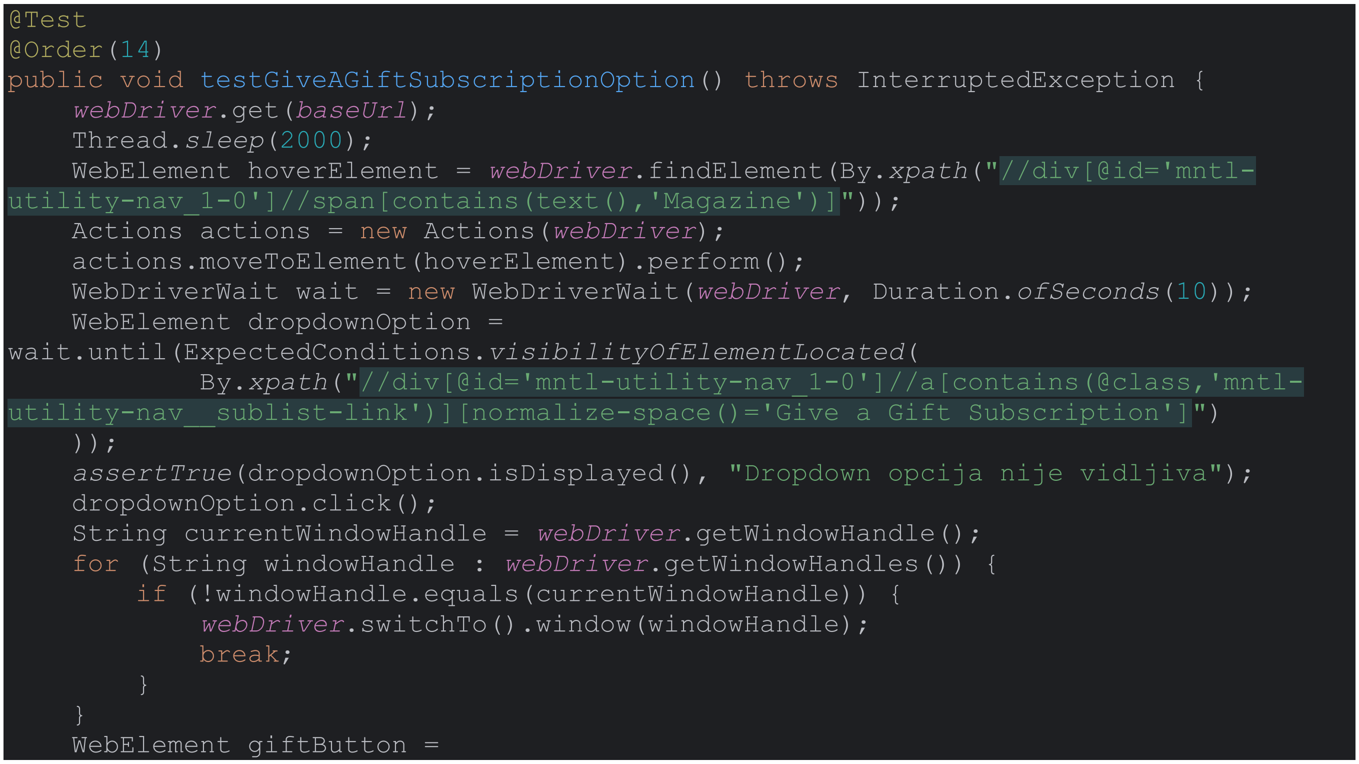
```
@Test
@Order(12)
public void testGiveAgiftLinkOnManageSubscriptionPage() throws InterruptedException
{
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption = 
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Manage Your Subscription']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement linkGiveAgift = webDriver.findElement(By.xpath("//a[normalize-
space()='Give a Gift']"));
    linkGiveAgift.click();
    Thread.sleep(3000);
    String expectedUrl = "https://www.magazines.com/eatingwell-
magazine.html?utm_source=engage&utm_medium=internal&utm_campaign=etq_care_crumb";
    assertEquals(expectedUrl, webDriver.getCurrentUrl(), "Thats not the expected
url.");
}
```

Test Name: Test give a git link redirects to gift page				
Description: Check if the "Give a Gift" link on the "Manage Your Subscription" page navigates to the gift page and displays the correct message.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Hover over the “Magazine” menu. 3. Wait for the “Manage Your Subscription” option to appear and click it. 4. Switch to the new tab. 5. Locate and click the “Give a Gift” link. 6. Wait for the gift page to load. 7. Verify that the text “Gift Options Available” is displayed on the gift page.	**Test Data:**	**Expected Result:** By clicking the "Give a Gift" link, the user is redirected to the gift page, and the text "Gift Options Available" is displayed.	**Actual Result:** By clicking the "Give a Gift" link, the user is redirected to the gift page, and the text "Gift Options Available" is displayed.	**Status:** PASS
Notes:				

```
@Test
@Order(13)
public void testGiveAgiftLinkOnManageSubscriptionPage1() throws
InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);

    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Manage Your Subscription']")
    ));

    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement linkGiveAgift = webDriver.findElement(By.xpath("//a[normalize-
space()='Give a Gift']"));
    linkGiveAgift.click();
    Thread.sleep(3000);
    WebElement giftText =
webDriver.findElement(By.xpath("//span[@class='gift__notice--text']"));
    Thread.sleep(2000);
    assertEquals("Gift Options Available", giftText.getText(), "Text doesn't match,
this is not gift page.");
}
```

Test Name: Test give a gift subscription option				
Description: Check if the "Give a Gift Subscription" link in the "Magazine" dropdown menu navigates to the gift subscription page and displays the "Give as Gift" button.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

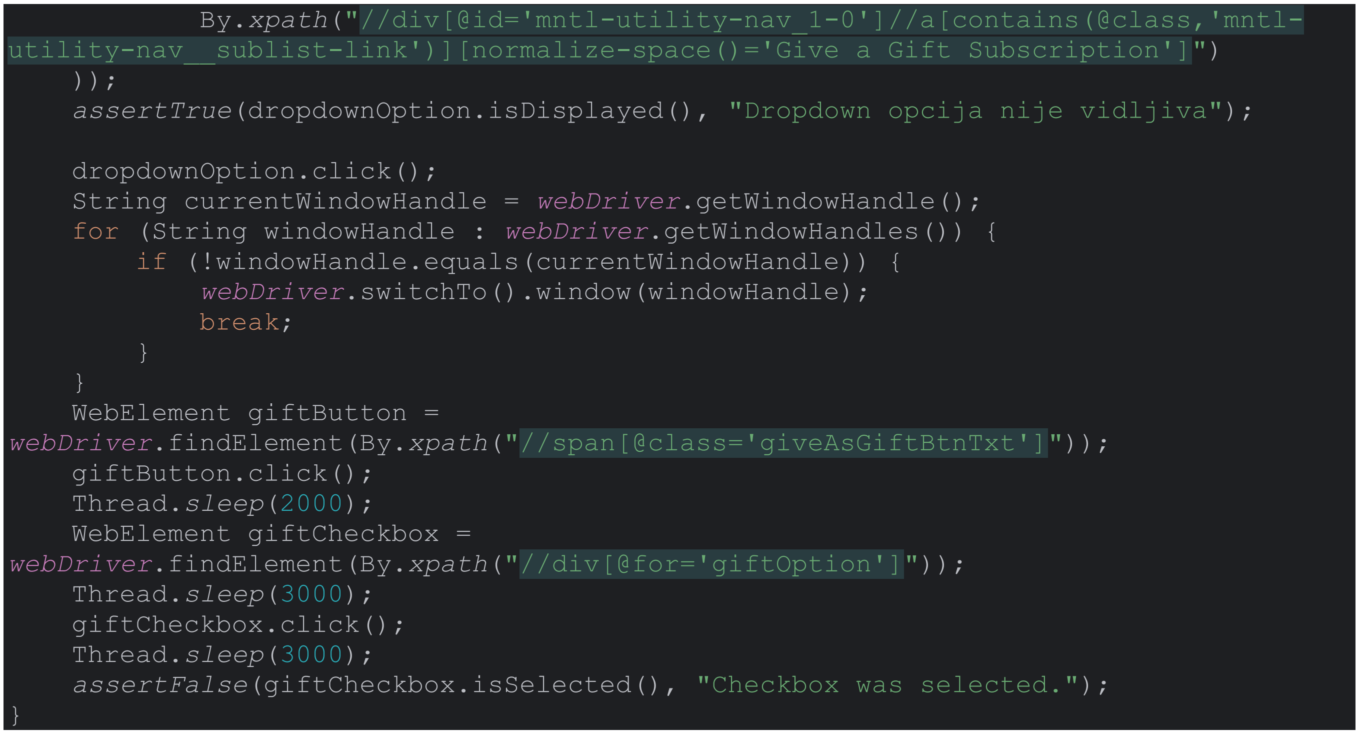
1. Wait for the page to load. 2. Hover over the “Magazine” menu. 3. Wait until the “Give a Gift Subscription” link is visible in the dropdown menu. 4. Verify that the “Give a Gift Subscription” link is displayed. 5. Click on the “Give a Gift Subscription” link. 6. Switch to the newly opened browser tab. 7. Locate the “Give as Gift” button on the gift subscription page. 8. Verify that the “Give as Gift” button is displayed on the page.		By clicking on the "Give a Gift Subscription" link, the user should be navigated to the gift subscription page where the "Give as Gift" button is displayed.	By clicking on the "Give a Gift Subscription" link, the user was navigated to the gift subscription page where the "Give as Gift" button was displayed.	PASS
Notes:				



```
@Test
@Order(14)
public void testGiveAGiftSubscriptionOption() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Give a Gift Subscription']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement giftButton =
webDriver.findElement(By.xpath("//span[@class='giveAsGiftBtnTxt']"));
    Thread.sleep(2000);
    assertTrue(giftButton.isDisplayed(), "There is no such button, so this is not
gift page.");
}
```

Test Name: Test give a gift checkbox				
Description: Check if the "Gift Option" checkbox on the gift subscription page can be clicked and selected correctly.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Hover over the “Magazine” menu. 3. Wait for and verify the “Give a Gift Subscription” link to be visible. 4. Click the “Give a Gift Subscription” link. 5. Switch to the new tab. 6. Click the “Give as Gift” button. 7. Wait for the gift options to load. 8. Click the “Gift Option” checkbox. 9. Verify the checkbox is selected.	**Test Data:**	**Expected Result:** By clicking on the "Gift Option" checkbox, the option should be selected, indicating the gift option is enabled for the subscription.	**Actual Result:** By clicking on the "Gift Option" checkbox, the option was selected, indicating the gift option is enabled for the subscription.	**Status:** PASS
Notes:				

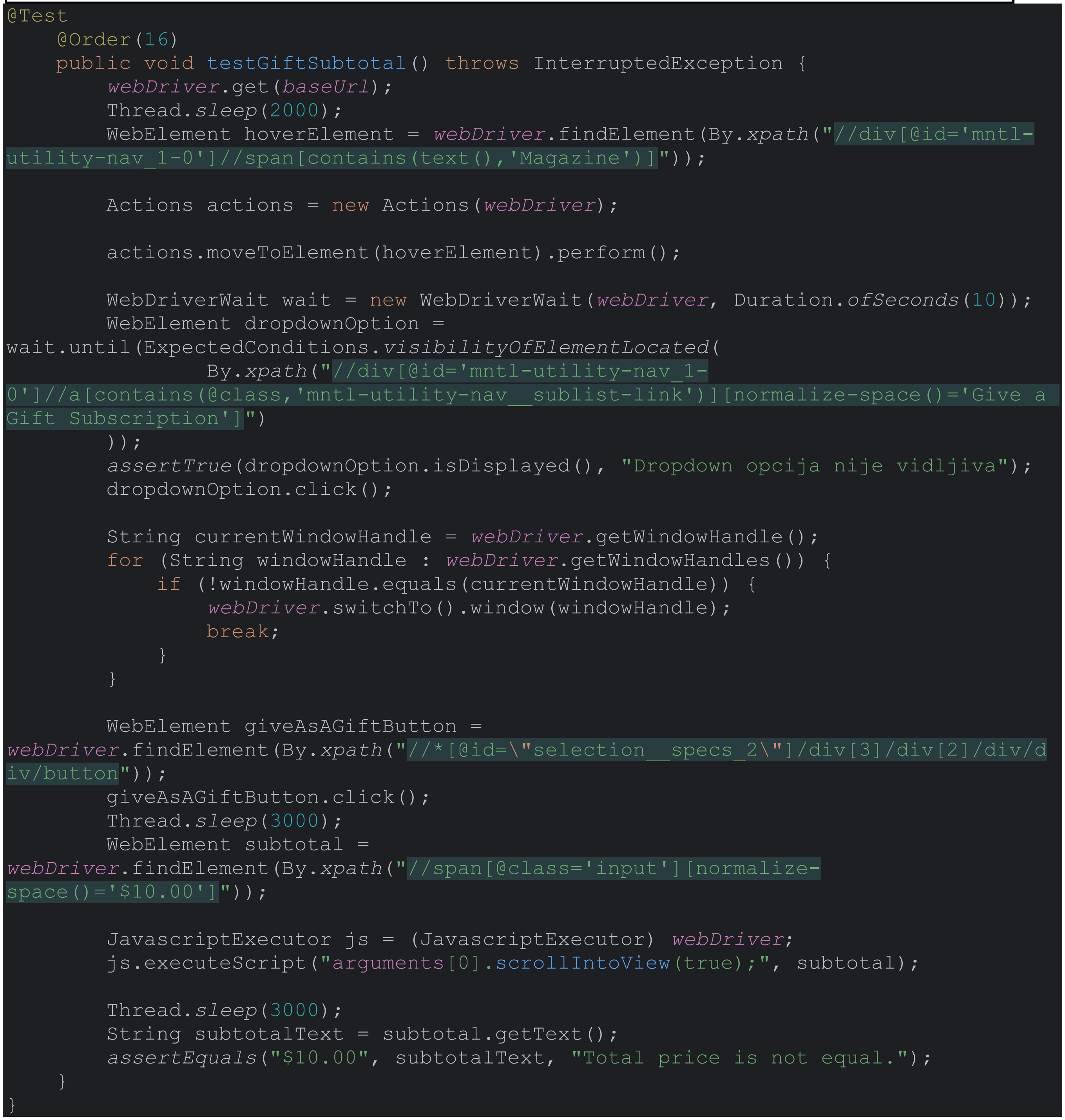
```
@Test
@Order(15)
public void testGiveAGiftCheckbox() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
```



```
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[contains(@class,'mntl-
utility-nav__sublist-link')][normalize-space()='Give a Gift Subscription']")
    ));
    assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");

    dropdownOption.click();
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {
            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement giftButton =
webDriver.findElement(By.xpath("//span[@class='giveAsGiftBtnTxt']"));
    giftButton.click();
    Thread.sleep(2000);
    WebElement giftCheckbox =
webDriver.findElement(By.xpath("//div[@for='giftOption']"));
    Thread.sleep(3000);
    giftCheckbox.click();
    Thread.sleep(3000);
    assertFalse(giftCheckbox.isSelected(), "Checkbox was selected.");
}
```

Test Name: Test gift subtotal				
Description: Check if the subtotal price is correctly displayed after clicking the "Give as Gift" button on the "Give a Gift Subscription" page.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Hover over the "Magazine" menu. 3. Wait for and verify the "Give a Gift Subscription" link to be visible. 4. Click the "Give a Gift Subscription" link. 5. Switch to the new tab. 6. Click the "Give as Gift" button. 7. Wait for the gift options to load. 8. Scroll down to the	**Test Data:**	**Expected Result:** By clicking the "Give as Gift" button, the subtotal price displayed was "$10.00".	**Actual Result:** By clicking the "Give as Gift" button, the subtotal price displayed was "$10.00".	**Status:** PASS

subtotal element. 9. Verify that the subtotal price is “$10.00”.				
Notes:				

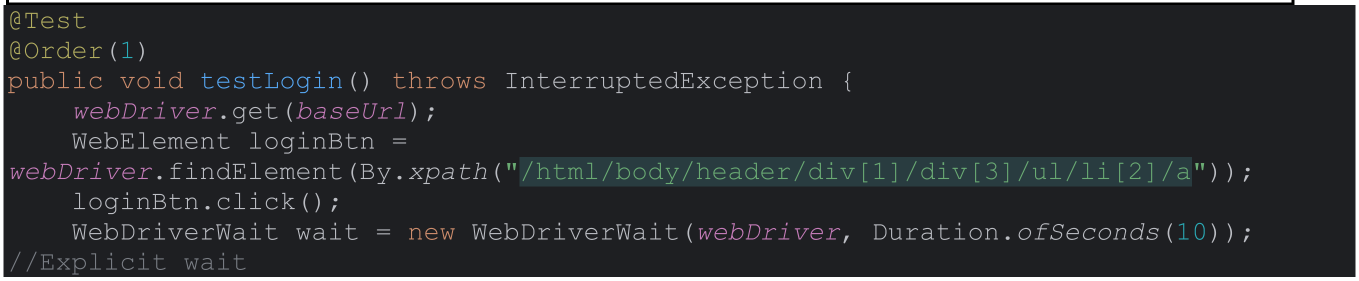


```
@Test
    @Order(16)
    public void testGiftSubtotal() throws InterruptedException {
        webDriver.get(baseUrl);
        Thread.sleep(2000);
        WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[contains(text(),'Magazine')]"));

        Actions actions = new Actions(webDriver);

        actions.moveToElement(hoverElement).perform();

        WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
        WebElement dropdownOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
                By.xpath("//div[@id='mntl-utility-nav_1-
0']//a[contains(@class,'mntl-utility-nav__sublist-link')][normalize-space()='Give a
Gift Subscription']")
        ));
        assertTrue(dropdownOption.isDisplayed(), "Dropdown opcija nije vidljiva");
        dropdownOption.click();

        String currentWindowHandle = webDriver.getWindowHandle();
        for (String windowHandle : webDriver.getWindowHandles()) {
            if (!windowHandle.equals(currentWindowHandle)) {
                webDriver.switchTo().window(windowHandle);
                break;
            }
        }

        WebElement giveAsAGiftButton =
webDriver.findElement(By.xpath("//*[@id=\"selection__specs_2\"]/div[3]/div[2]/div/d
iv/button"));
        giveAsAGiftButton.click();
        Thread.sleep(3000);
        WebElement subtotal =
webDriver.findElement(By.xpath("//span[@class='input'][normalize-
space()='$10.00']"));

        JavascriptExecutor js = (JavascriptExecutor) webDriver;
        js.executeScript("arguments[0].scrollIntoView(true);", subtotal);

        Thread.sleep(3000);
        String subtotalText = subtotal.getText();
        assertEquals("$10.00", subtotalText, "Total price is not equal.");
    }
}
```

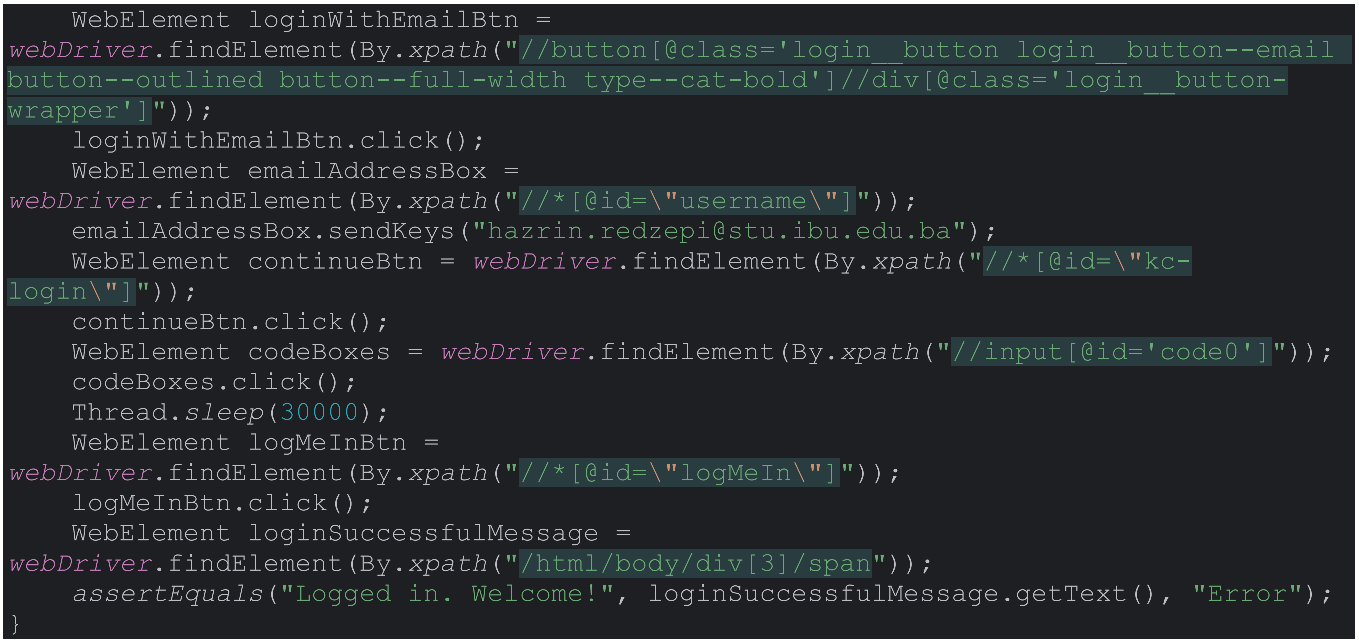
3.4. Test Scenario Name: Login test

This test scenario is about testing login and it's main functionalities, log out and all actions you can do when you are logged in and when you are logged out.

Test Name: Test login				
Description: Check if the user can successfully log in using their email address and verification code.				
Pre-condition(s): The user must be registered to access the login functionality.				
Test Steps: 1. Wait for the page to load. 2. Locate and click the Login button in the header. 3. Wait for the login options to load. 4. Select the Login with Email option. 5. Enter a valid email address in the email input field and click the Continue button. 6. Wait for the verification code field to load. 7. Enter the verification code in the provided fields and click the Log Me In button. 8. Verify the success message "Logged In. Welcome!" is displayed.	**Test Data:** The following specific input data is used for this test case: Email Address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test (manually entered).	**Expected Result:** By completing the login process, the message "Logged in. Welcome!" should appear, confirming a successful login.	**Actual Result:** By completing the login process, the message "Logged in. Welcome!" was displayed, confirming a successful login.	**Status:** PASS
Notes:				



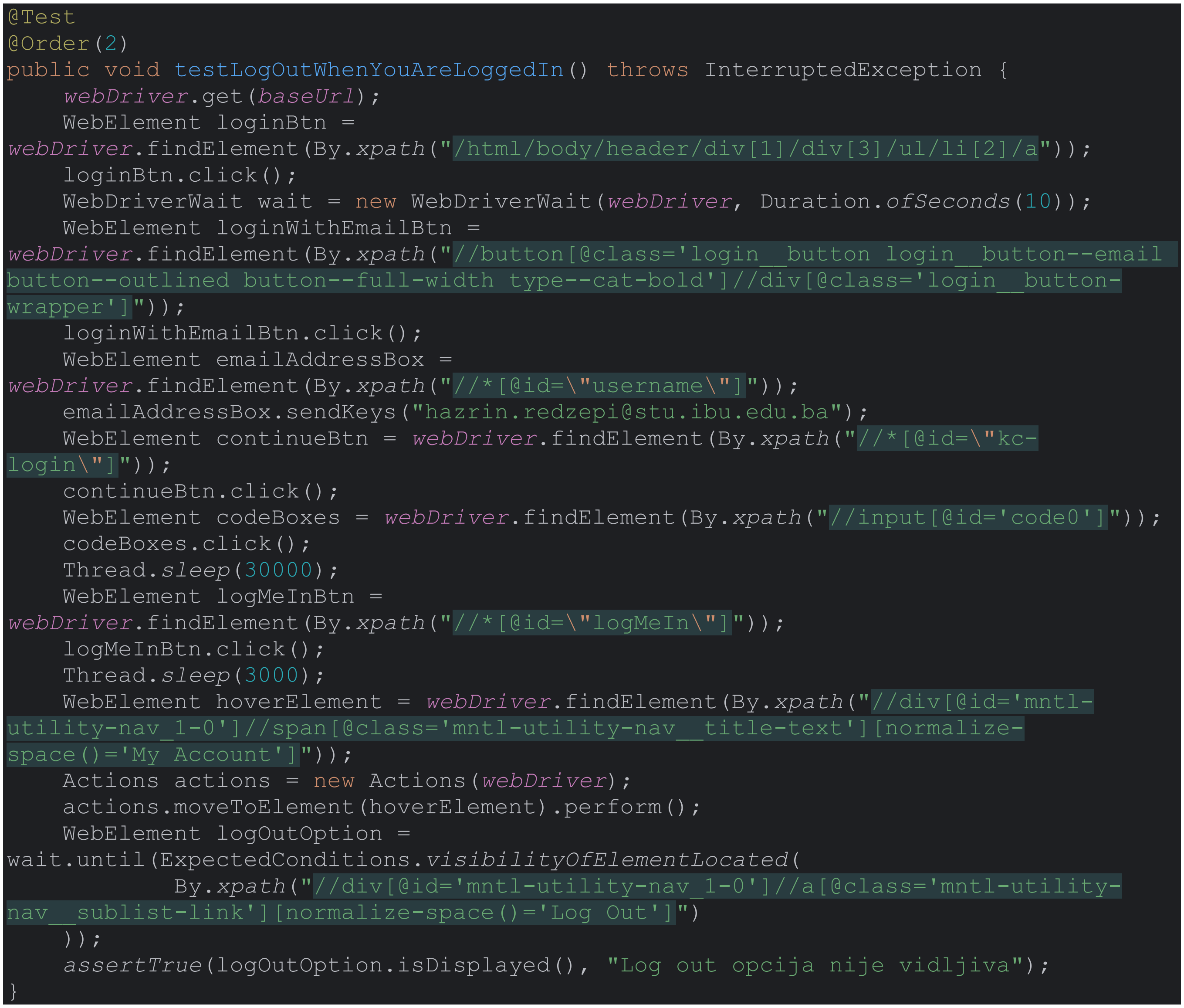
```
@Test
@Order(1)
public void testLogin() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn =
webDriver.findElement(By.xpath("/html/body/header/div[1]/div[3]/ul/li[2]/a"));
    loginBtn.click();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
//Explicit wait
```



```
    WebElement loginWithEmailBtn =
webDriver.findElement(By.xpath("//button[@class='login__button login__button--email
button--outlined button--full-width type--cat-bold']//div[@class='login__button-
wrapper']"));
    loginWithEmailBtn.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    WebElement loginSuccessfulMessage =
webDriver.findElement(By.xpath("/html/body/div[3]/span"));
    assertEquals("Logged in. Welcome!", loginSuccessfulMessage.getText(), "Error");
}
```

Test Name: Test logout when you are logged in				
Description: Check if the "Log Out" option appears in the "My Account" dropdown after logging in.				
Pre-condition(s): The user must be logged in first.				
Test Steps: 1. Wait for the page to load. 2. Click the “Login” button. 3. Wait for and click the “Login with Email” button. 4. Enter the valid email in the email address field and click the “Continue” button. 5. Wait for the code input field and click on it. 6. Wait for and click the “Log Me In” button. 7. Wait for the page to load and hover over the “My Account”	**Test Data:** The following specific input data is used for this test case: Email Address: hazrin.redzepi@stu.ibu.edu.ba	**Expected Result:** By logging in, the "Log Out" option will appear in the "My Account" dropdown.	**Actual Result:** By logging in, the "Log Out" option is visible in the "My Account" dropdown.	**Status:** PASS

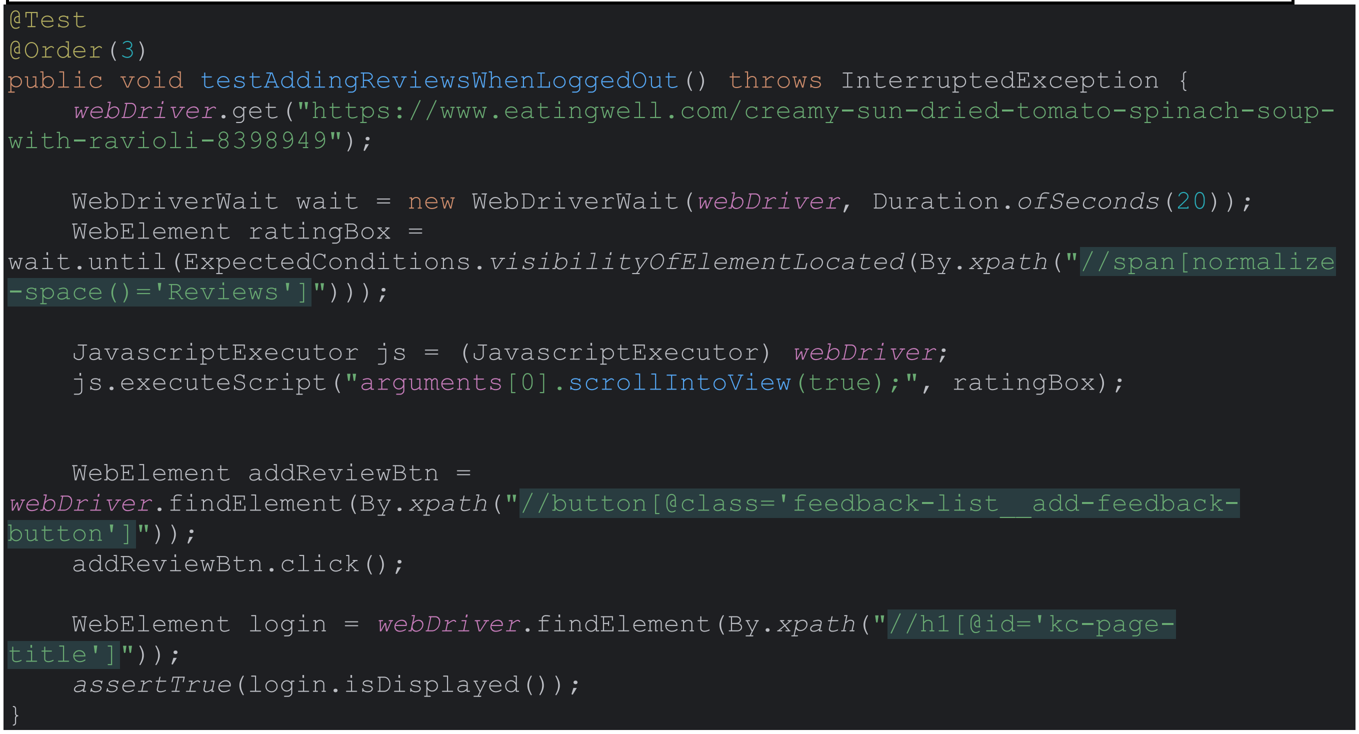
menu. 8. Verify the “Log Out” option to be visible.				
Notes:				



```
@Test
@Order(2)
public void testLogOutWhenYouAreLoggedIn() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn =
webDriver.findElement(By.xpath("/html/body/header/div[1]/div[3]/ul/li[2]/a"));
    loginBtn.click();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement loginWithEmailBtn =
webDriver.findElement(By.xpath("//button[@class='login__button login__button--email
button--outlined button--full-width type--cat-bold']//div[@class='login__button-
wrapper']"));
    loginWithEmailBtn.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(3000);
    WebElement hoverElement = webDriver.findElement(By.xpath("//div[@id='mntl-
utility-nav_1-0']//span[@class='mntl-utility-nav__title-text'][normalize-
space()='My Account']"));
    Actions actions = new Actions(webDriver);
    actions.moveToElement(hoverElement).perform();
    WebElement logOutOption =
wait.until(ExpectedConditions.visibilityOfElementLocated(
            By.xpath("//div[@id='mntl-utility-nav_1-0']//a[@class='mntl-utility-
nav__sublist-link'][normalize-space()='Log Out']")
    ));
    assertTrue(logOutOption.isDisplayed(), "Log out opcija nije vidljiva");
}
```

Test Name: Test adding reviews when logged out
Description: Check if the user is prompted to log in when trying to add a review while logged out.

Pre-condition(s): The user must be logged in before the test begins, and then logged out in the test.				
Test Steps: 1. Wait for the page to load. 2. Wait for the "Reviews" section to be visible. 3. Scroll to the "Reviews" section. 4. Click the "Add Review" button. 5. Verify that the login page is displayed.	**Test Data:**	**Expected Result:** By clicking the "Add Review" button, the user will be redirected to the login page, and the login prompt will be visible.	**Actual Result:** By clicking the "Add Review" button, the user is redirected to the login page, and the login prompt is displayed.	**Status:** PASS
Notes:				



```
@Test
@Order(3)
public void testAddingReviewsWhenLoggedOut() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-with-ravioli-8398949");

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement ratingBox =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize-space()='Reviews']")));

    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);


    WebElement addReviewBtn =
webDriver.findElement(By.xpath("//button[@class='feedback-list__add-feedback-button']"));
    addReviewBtn.click();

    WebElement login = webDriver.findElement(By.xpath("//h1[@id='kc-page-title']"));
    assertTrue(login.isDisplayed());
}
```

Test Name: Test adding hearts when logged out
Description: Check if the user is prompted to log in when attempting to add a heart (like) to a recipe after logging out.
Pre-condition(s): The user must be logged out before performing the test.

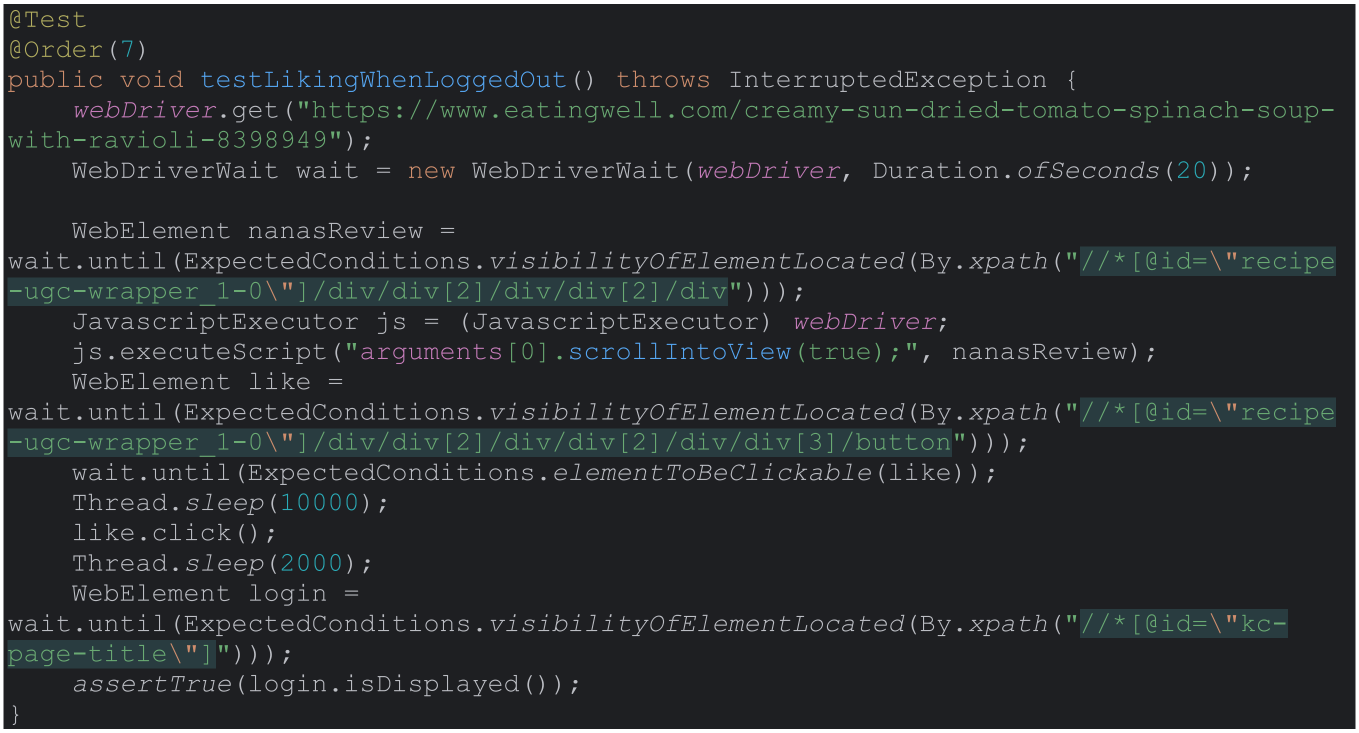
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Scroll to the “Articles” section. 3. Wait for and locate the heart icon for liking the recipe. 4. Wait for the heart icon to become clickable and click the heart icon. 5. Wait for the login button to be displayed. 6. Verify the login button is visible.		After clicking the heart icon, the user should be redirected to the login page, where the login button is displayed.	After clicking the heart icon, the user is redirected to the login page, and the login button is displayed.	PASS

Notes:

```
@Test
@Order(6)
public void testAddingHeartsWhenLoggedOut() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-with-ravioli-8398949");

    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement articles = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//h2[@id='mntl-recirc-section__header_1-0']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", articles);
    WebElement heart = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"card__favorite_5-0\"]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(heart));
    Thread.sleep(10000);
    heart.click();
    Thread.sleep(2000);
    WebElement loginButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"mm-myrecipes-interstitial__content_1-0\"]/div[1]/div[2]/a")));
    assertTrue(loginButton.isDisplayed());
}
```

Test Name: Test liking when logged out
Description: Check if the user is prompted to log in when attempting to like a review after being logged out.

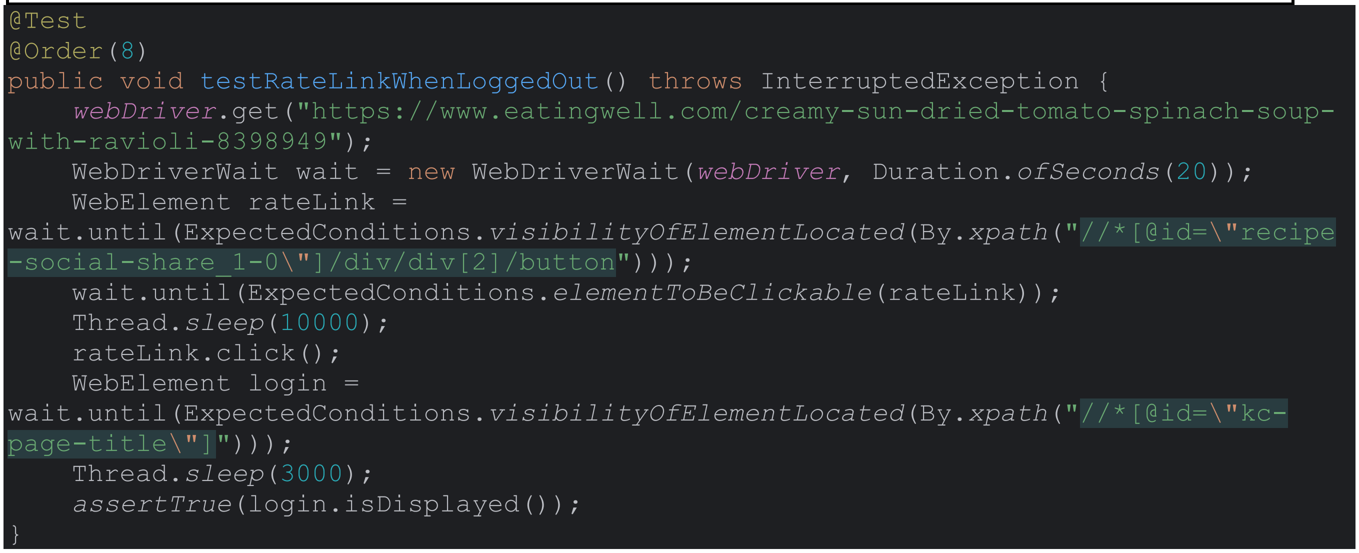
Pre-condition(s): The user must be logged out before performing the test.				
Test Steps: 1. Wait for the page to load. 2. Scroll to the “Nana’s Review” section. 3. Wait for and locate the “like” button. 4. Wait for the “like” button to become clickable and click on it. 5. Wait for the page to load. 6. Verify the login page is visible with the login prompt.	**Test Data:**	**Expected Result:** By clicking the "like" button on the review, the user should be prompted to log in, and the login page should be displayed.	**Actual Result:** After clicking the like button, the user is redirected to the login page, and the login prompt is displayed.	**Status:** PASS
Notes:				



```
@Test
@Order(7)
public void testLikingWhenLoggedOut() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));

    WebElement nanasReview =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe
-ugc-wrapper_1-0\"]/div/div[2]/div/div[2]/div")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", nanasReview);
    WebElement like =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe
-ugc-wrapper_1-0\"]/div/div[2]/div/div[2]/div/div[3]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(like));
    Thread.sleep(10000);
    like.click();
    Thread.sleep(2000);
    WebElement login =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"kc-
page-title\"]")));
    assertTrue(login.isDisplayed());
}
```

Test Name: Test rate the recipe when logged out
Description: Check if the user is prompted to log in when attempting to rate the recipe after being logged out.

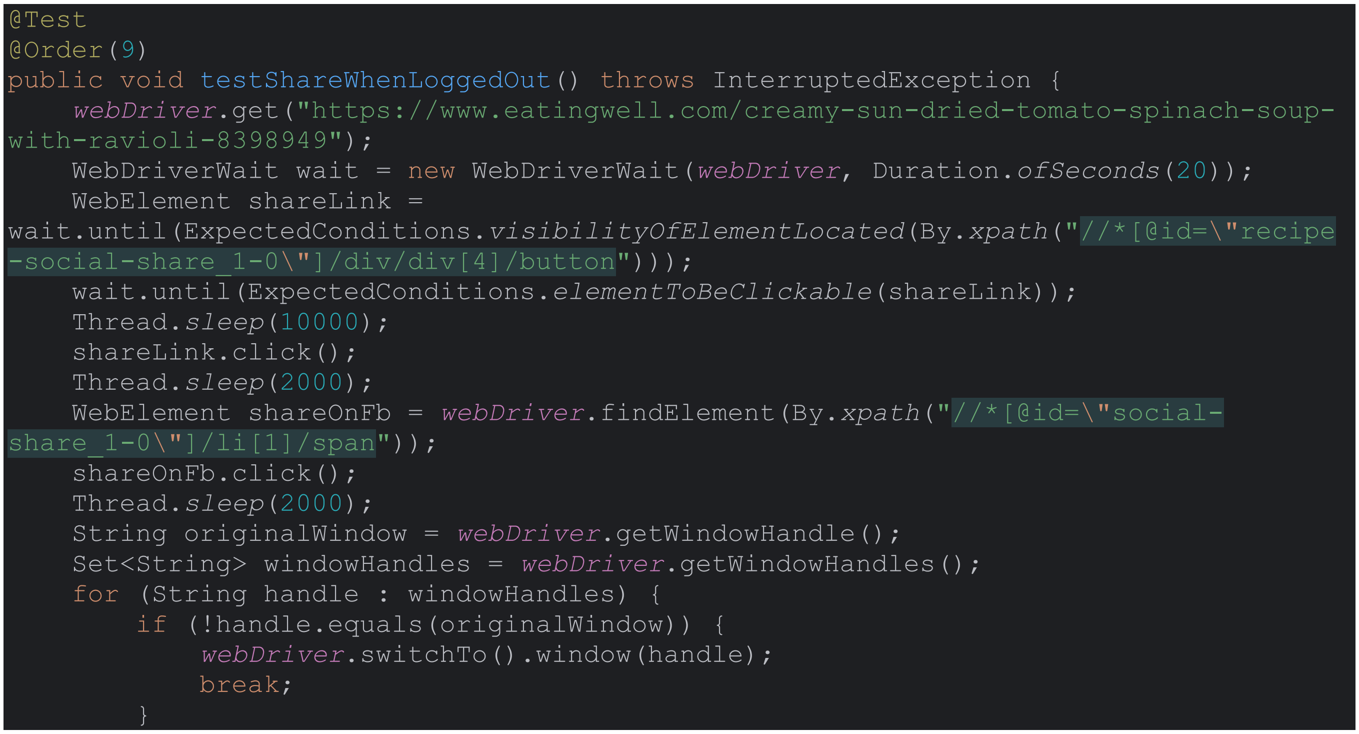
Pre-condition(s): The user must be logged out before performing the test.				
Test Steps: 1. Wait for the page to load. 2. Wait for the “Rate” link to become visible. 3. Wait for the “Rate” link to become clickable and click on it. 4. Wait for the login page to load. 5. Verify that the login page is displayed with the login prompt.	**Test Data:**	**Expected Result:** By clicking the "Rate" button, the user should be prompted to log in, and the login page should be displayed.	**Actual Result:** After clicking the "Rate" link, the user is redirected to the login page, and the login prompt is displayed.	**Status:** PASS
Notes:				



```
@Test
@Order(8)
public void testRateLinkWhenLoggedOut() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement rateLink =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe
-social-share_1-0\"]/div/div[2]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(rateLink));
    Thread.sleep(10000);
    rateLink.click();
    WebElement login =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"kc-
page-title\"]")));
    Thread.sleep(3000);
    assertTrue(login.isDisplayed());
}
```

Test Name: Test share the recipe when logged out				
Description: Check if the user is prompted to log in when trying to share the recipe on Facebook after being logged out.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load. 2. Wait for the “Share” button to become visible and then wait for the “Share” button to become clickable and then click on it. 3. Wait for the Facebook share button to be visible and clickable. 4. Switch to the new window opened by the Facebook share link. 5. Verify the URL is correct and verify the message “Not Logged In” is displayed. 6. Switch back to the original window.		After clicking the "Share" button and choosing to share on Facebook, the user should be redirected to the Facebook login page with the message "Not Logged In".	After clicking the "Share" button and selecting Facebook, the user is redirected to the Facebook login page, and the message "Not Logged In" is displayed.	PASS
Notes:				

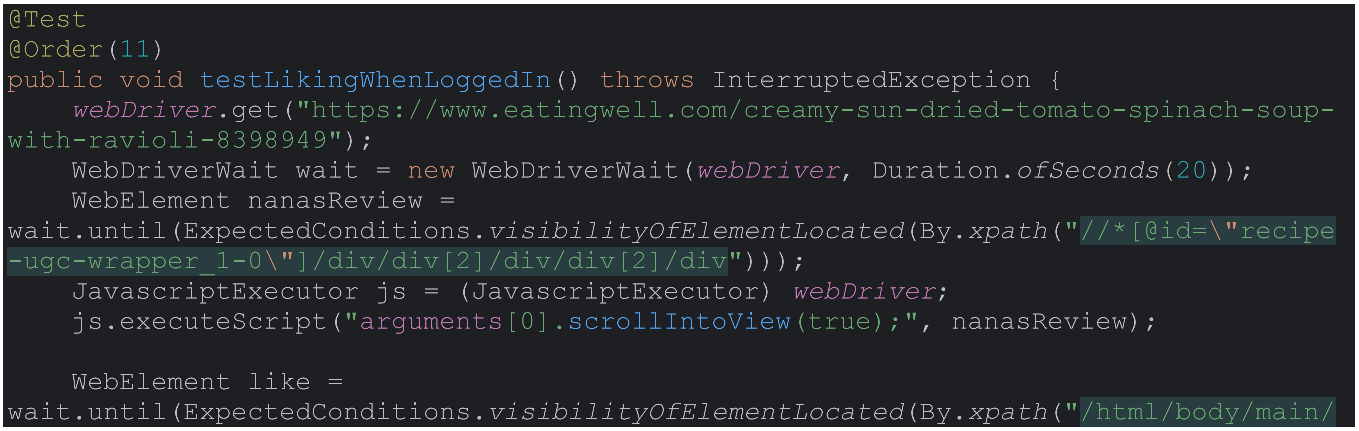


```
@Test
@Order(9)
public void testShareWhenLoggedOut() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement shareLink =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe
-social-share_1-0\"]/div/div[4]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(shareLink));
    Thread.sleep(10000);
    shareLink.click();
    Thread.sleep(2000);
    WebElement shareOnFb = webDriver.findElement(By.xpath("//*[@id=\"social-
share_1-0\"]/li[1]/span"));
    shareOnFb.click();
    Thread.sleep(2000);
    String originalWindow = webDriver.getWindowHandle();
    Set<String> windowHandles = webDriver.getWindowHandles();
    for (String handle : windowHandles) {
        if (!handle.equals(originalWindow)) {
            webDriver.switchTo().window(handle);
            break;
        }
    }
    WebElement message =
webDriver.findElement(By.xpath("//h2[@class='uiHeaderTitle']"));
    String expectedUrl =
"https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.eatingwell.com%2Fcr
eamy-sun-dried-tomato-spinach-soup-with-ravioli-
8398949%3Futm_source%3Dfacebook.com%26utm_medium%3Dsocial%26utm_campaign%3Dsocial-
share-article";
    assertEquals(webDriver.getCurrentUrl(), expectedUrl, "Didn't navigate to the
correct page in the new window");
    assertTrue(message.getText().equals("Not Logged In"), "You are logged in.");
    webDriver.switchTo().window(originalWindow);
}
```

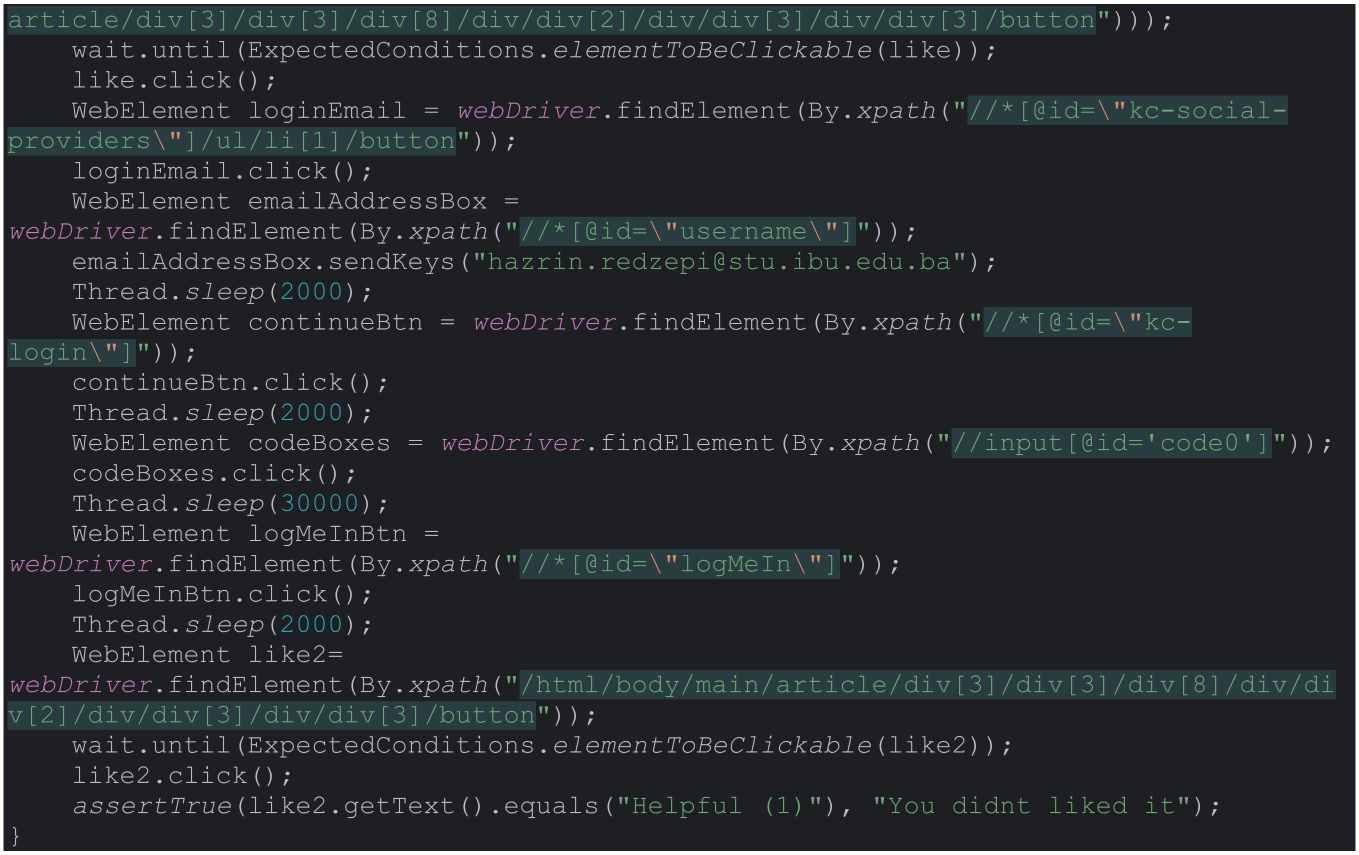
Test Name: Test login when logged out				
Description: Verify if the login page is displayed when the user clicks the "Log In" link while logged out.				
Pre-condition(s): The user must be logged out before performing the test.				
Test Steps: 1. Wait for the page to load. 2. Click the “Log In” link in the utility navigation. 3. Wait for the login page to load. 4. Verify that the login page is visible by checking the title of the page.	**Test Data:**	**Expected Result:** After clicking the "Log In" link, the user should be redirected to the login page, and the login form should be visible.	**Actual Result:** After clicking the "Log In" link, the login page is displayed, and the login form is visible.	**Status:** PASS
Notes:				

```
@Test
@Order(10)
public void testLoginWhenLoggedOut() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement login = webDriver.findElement(By.xpath("//div[@id='mntl-utility-
nav_1-0']//span[@class='mntl-utility-nav__sublist-link-text'][normalize-
space()='Log In']"));
    login.click();
    Thread.sleep(2000);
    WebElement loginWindow = webDriver.findElement(By.xpath("//h1[@id='kc-page-
title']"));
    assertTrue(loginWindow.isDisplayed());
}
```

Test Name: Test liking when logged in				
Description: Check if a logged-in user can like a review and see the "Helpful (1)" status after clicking the like button.				
Pre-condition(s): User must be logged out before the test and should have an existing account to be able to log in.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Click on the “Log In” button. 3. Wait for the login page to appear. 4. Enter valid credentials. 5. Wait for the code input field to appear and enter the authentication code. 6. Click on the login button. 7. Wait for the page to load after successful login. 8. Scroll down to the review section and find the like button and click on it. 9. Verify that the like button text changes to “Helpful (1).	The following specific input data is used for this test case: Email Address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test (manually entered).	After clicking the "like" button when logged in, the text on the button should change to "Helpful (1)" to reflect the user's action.	After the user logs in and clicks the "like" button, the text changes to "Helpful (1)", indicating the action was successful.	PASS
Notes:				



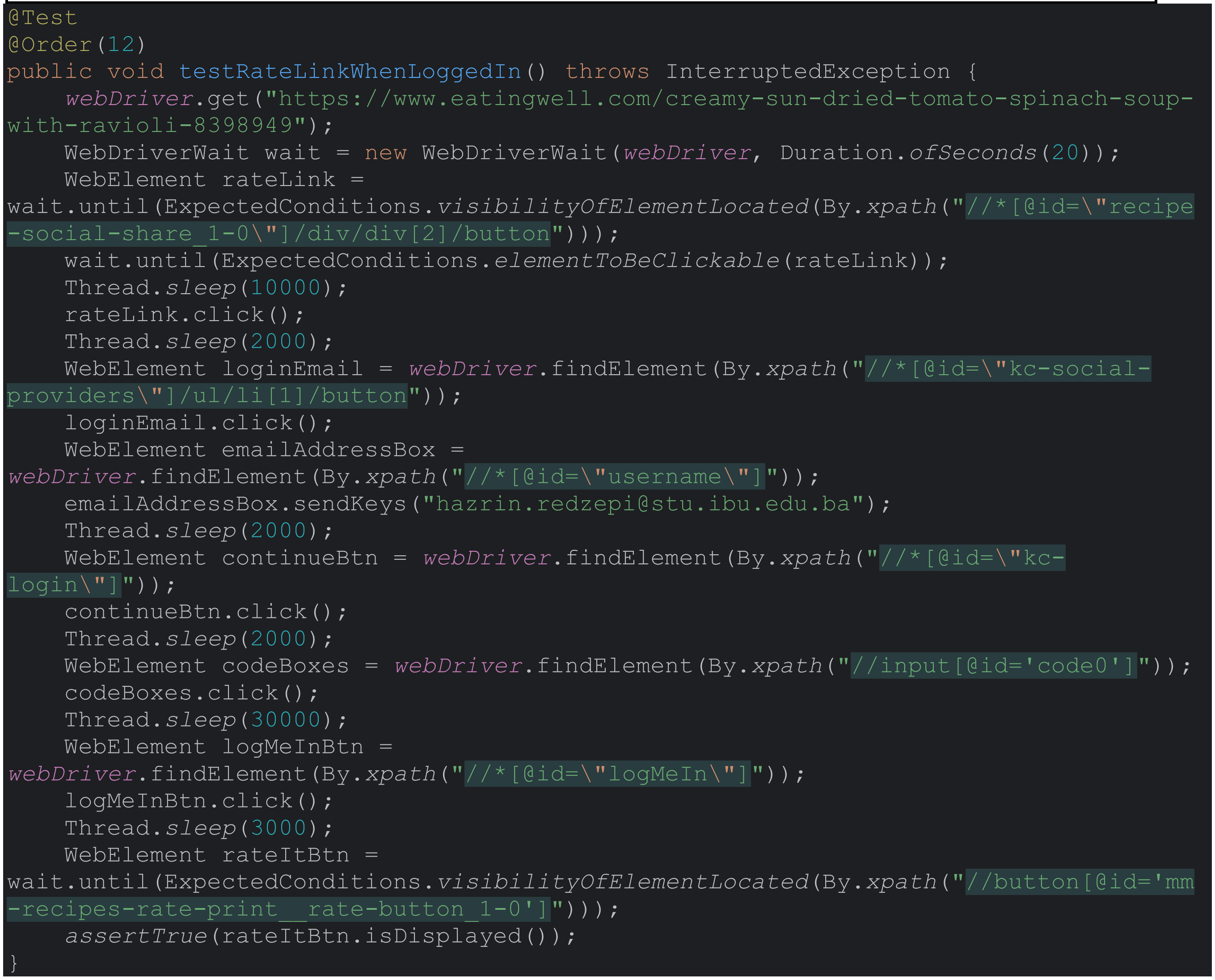
```
@Test
@Order(11)
public void testLikingWhenLoggedIn() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement nanasReview = 
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe-ugc-wrapper_1-0\"]/div/div[2]/div/div[2]/div")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", nanasReview);

    WebElement like = 
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("/html/body/main/
```



```
article/div[3]/div[3]/div[8]/div/div[2]/div/div[3]/div/div[3]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(like));
    like.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement like2=
webDriver.findElement(By.xpath("/html/body/main/article/div[3]/div[3]/div[8]/div/di
v[2]/div/div[3]/div/div[3]/button"));
    wait.until(ExpectedConditions.elementToBeClickable(like2));
    like2.click();
    assertTrue(like2.getText().equals("Helpful (1)"), "You didnt liked it");
}
```

Test Name: Test rate link when logged in				
Description: Check if the "Rate It" button is displayed after the user logs in successfully and clicks on the "Rate" link.				
Pre-condition(s): User must be logged out before the test and should have an existing account to log in.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Find the "rate" link and wait until it's clickable and click on it. 3. Wait for the login page to appear and click on the "Log In with Email" button. 4. Enter valid credentials. 5. Wait for the code input field to appear	The following specific input data is used for this test case: Email Address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test (manually entered).	The "Rate It" button should be visible after successful login.	The "Rate It" button was successfully displayed after login.	PASS

and enter the authentication code. 6. Click on the login button. 7. Wait for the page to load after successful login. 8. Verify the “Rate It” button visibility by checking if it is displayed.				
Notes:				



```
@Test
@Order(12)
public void testRateLinkWhenLoggedIn() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement rateLink =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//*[@id=\"recipe
-social-share_1-0\"]/div/div[2]/button")));
    wait.until(ExpectedConditions.elementToBeClickable(rateLink));
    Thread.sleep(10000);
    rateLink.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(3000);
    WebElement rateItBtn =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//button[@id='mm
-recipes-rate-print__rate-button_1-0']")));
    assertTrue(rateItBtn.isDisplayed());
}
```

3.5. Test Scenario Name: Registration test

This test scenario is about registration.

<table>
<tr><td colspan="5"><b>Test Name: Test register</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if the user can successfully register an account by completing the registration process and receiving a success message.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b> User must not be logged in before the test and should have a valid email address for registration.</td></tr>
<tr><td><b>Test Steps:</b>
1. Wait for the page to load.
2. Find and click the “Login” button.
3. Wait for the “Join Now” button to be clickable and click it.
4. Click on the “Sign Up with Email” button.
5. Wait for the email input field and enter a valid email address.
6. Wait for the “Join Now” button to be clickable and click it.
7. Wait for the code input field to appear and click on it.
8. Wait for the authentication process and click the “Log Me In” button.
9. Wait for the page to load after successful registration.
10. Verify that the success message “Account created. Welcome!” is displayed.</td><td><b>Test Data:</b>
The test data includes an email address (skerim3112@gmail.com), a valid verification code received via email.</td><td><b>Expected Result:</b>
A success message with the text "Account created. Welcome!" should appear after successful registration.</td><td><b>Actual Result:</b>
The success message "Account created. Welcome!" was displayed as expected after the registration process.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b></td></tr>
</table>

```
@Test
@Order(1)
public void testRegister () throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn =
webDriver.findElement(By.xpath("/html/body/header/div[1]/div[3]/ul/li[2]/a"));
    loginBtn.click();
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));
    WebElement joinNowBtn =
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("/html/body/div[1]/div[
2]/div[2]/div/div/div/div[3]/div/div/div/span/a")));
    joinNowBtn.click();
    WebElement SignUpWithEmailBtn =
webDriver.findElement(By.xpath("/html/body/div[1]/div[2]/div[2]/div/div/div/div[1]/
ul/li[1]/button"));
    SignUpWithEmailBtn.click();
    Thread.sleep(2000);
    WebElement emailAddressBtn =
webDriver.findElement(By.xpath("//*[@id=\"email\"]"));
    emailAddressBtn.sendKeys("skerim3112@gmail.com");
    Thread.sleep(2000);
    WebElement joinNowBtn1 = webDriver.findElement(By.xpath("//input[@id='kc-
register']"));
    joinNowBtn1.click();
    Thread.sleep(3000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//*[@id=\"code0\"]"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(3000);
    WebElement loginSuccessfulMessage =
webDriver.findElement(By.xpath("/html/body/div[3]/span"));
    assertEquals("Account created. Welcome!", loginSuccessfulMessage.getText(),
"Error");

}
```

<table>
<tr><td colspan="5"><b>Test Name: Test register with invalid email address</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if an invalid email address triggers an error message during the registration process.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b></td></tr>
<tr><td><b>Test Steps:</b></td><td><b>Test Data:</b></td><td><b>Expected Result:</b></td><td><b>Actual Result:</b></td><td><b>Status:</b></td></tr>
</table>

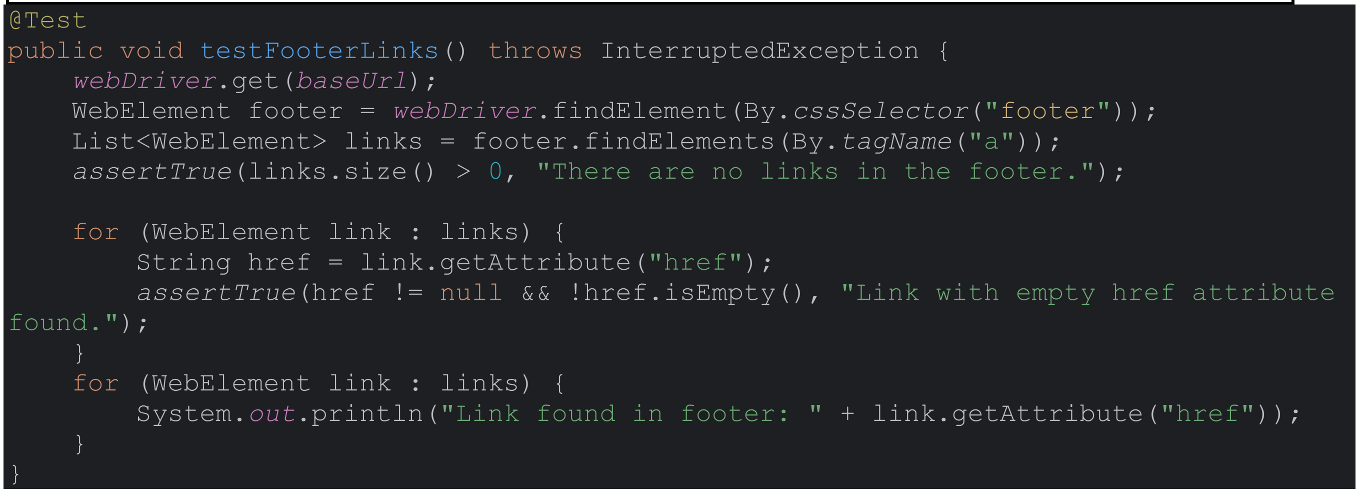
1. Wait for the page to load. 2. Find and click the “Login” button. 3. Wait for the “Join Now” button to be clickable and click it. 4. Click on the “Sign Up with Email” button. 5. Wait for the email input field and enter an invalid email address. 6. Wait for the “Join Now” button to be clickable and click it. 7. Wait for the error message to appear. 8. Verify that the error message “Invalid email address” is displayed.	Email address: lamija@gmail.com	The error message "Invalid email address" should be displayed when an invalid email address is entered.	The error message "Invalid email address" is displayed.	PASS
Notes:				

```
@Test
@Order(3)
public void testInvalidEmailAddress() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn =
webDriver.findElement(By.xpath("/html/body/header/div[1]/div[3]/ul/li[2]/a"));
    loginBtn.click();
    WebElement joinNowBtn =
webDriver.findElement(By.xpath("/html/body/div[1]/div[2]/div[2]/div/div/div/div[3]/
div/div/div/span/a"));
    joinNowBtn.click();
    WebElement SignUpWithEmailBtn =
webDriver.findElement(By.xpath("/html/body/div[1]/div[2]/div[2]/div/div/div/div[1]/
ul/li[1]/button"));
    SignUpWithEmailBtn.click();
    WebElement emailAddressBtn =
webDriver.findElement(By.xpath("//*[@id=\"email\"]"));
    emailAddressBtn.sendKeys("lamija@gmail.com");
    WebElement joinNowBtn1 = webDriver.findElement(By.xpath("//input[@id='kc-
register']"));
    joinNowBtn1.click();
    WebElement invalidAddress = webDriver.findElement(By.xpath("//*[@id=\"input-
error\"]/span"));
    assertEquals("Invalid email address", invalidAddress.getText(), "Error text");

}
```

3.6. Test Scenario Name: Footer test

This test scenario is about footer of our page. It has a critical functionality in footer so we have to test that (newsletters). Also we tested all links in footer.

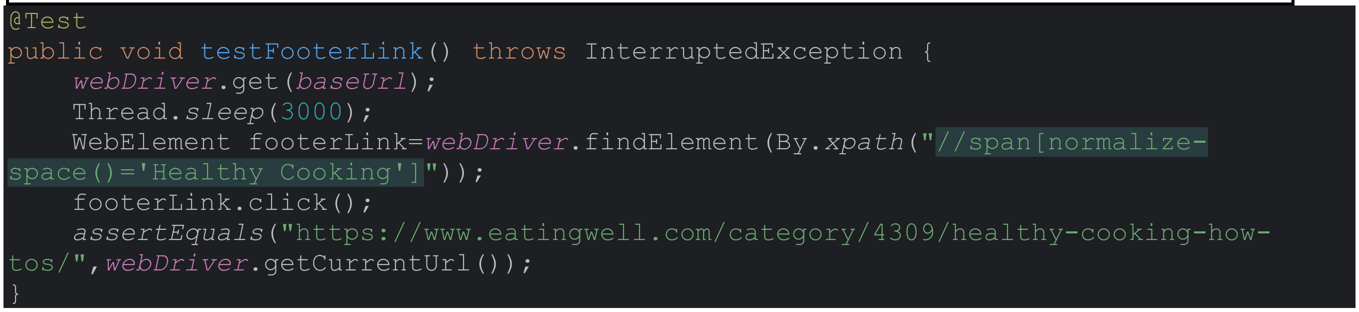
<table>
<tr><td colspan="5"><b>Test Name: Test footer links</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if there are valid links in the footer of the page and ensure that all links have a non-empty href attribute.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b></td></tr>
<tr><td><b>Test Steps:</b>
1. Wait for the page to load.
2. Find the footer element.
3. Find all the links inside the footer using the <a> tag.
4. Verify that there are links in the footer.
5. Verify each link to ensure it has a non-empty href attribute.
6. Print each link’s href attribute.</td><td><b>Test Data:</b></td><td><b>Expected Result:</b>
All links in the footer should be valid, with a non-empty href attribute, and the href values should be printed in the console.</td><td><b>Actual Result:</b>
Links are found in the footer, each with a non-empty href attribute, and the href values are printed in the console.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b></td></tr>
</table>



```
@Test
public void testFooterLinks() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement footer = webDriver.findElement(By.cssSelector("footer"));
    List<WebElement> links = footer.findElements(By.tagName("a"));
    assertTrue(links.size() > 0, "There are no links in the footer.");

    for (WebElement link : links) {
        String href = link.getAttribute("href");
        assertTrue(href != null && !href.isEmpty(), "Link with empty href attribute
found.");
    }
    for (WebElement link : links) {
        System.out.println("Link found in footer: " + link.getAttribute("href"));
    }
}
```

Test Name: Test footer specific link
Description: Check if clicking the "Healthy Cooking" footer link navigates to the expected

URL.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the “Healthy Cooking” link in the footer. 3. Click on the “Healthy Cooking” link. 4. Verify that the page navigates to the correct URL.	**Test Data:**	**Expected Result:** The page should navigate to the URL: https://www.eatingwell.com/category/4309/healthy-cooking-how-tos/.	**Actual Result:** The page navigates to the correct URL: https://www.eatingwell.com/category/4309/healthy-cooking-how-tos/.	**Status:** PASS
Notes:				



```
@Test
public void testFooterLink() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement footerLink=webDriver.findElement(By.xpath("//span[normalize-space()='Healthy Cooking']"));
    footerLink.click();
    assertEquals("https://www.eatingwell.com/category/4309/healthy-cooking-how-tos/",webDriver.getCurrentUrl());
}
```

Test Name: Test footer newsletters button				
Description: Check if clicking the "Healthy Cooking" footer link navigates to the expected URL.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find and click the “Healthy Cooking” link in the footer. 3. Verify if the page navigates to the correct URL.	**Test Data:**	**Expected Result:** The page should navigate to the URL: https://www.eatingwell.com/category/4309/healthy-cooking-how-tos/.	**Actual Result:** The page navigates to the correct URL: https://www.eatingwell.com/category/4309/healthy-cooking-how-tos/.	**Status:** PASS
Notes:				

```java
@Test
public void testFooterNewslettersBtn() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement newslettersBtn=webDriver.findElement(By.xpath("//a[@id='mntl-newsletter-dialog--footer-link_1-0']//span[@class='link__wrapper'][normalize-space()='Newsletters']"));
    newslettersBtn.click();
    Thread.sleep(2000);
    WebElement newslettersSignUp=
webDriver.findElement(By.xpath("//*[@id=\"newsletter-dialog-footer_1-0-title\"]"));
    assertTrue(newslettersSignUp.getText().equals("Newsletter Sign Up"));
}
```

3.7. Test Scenario Name: Homepage test

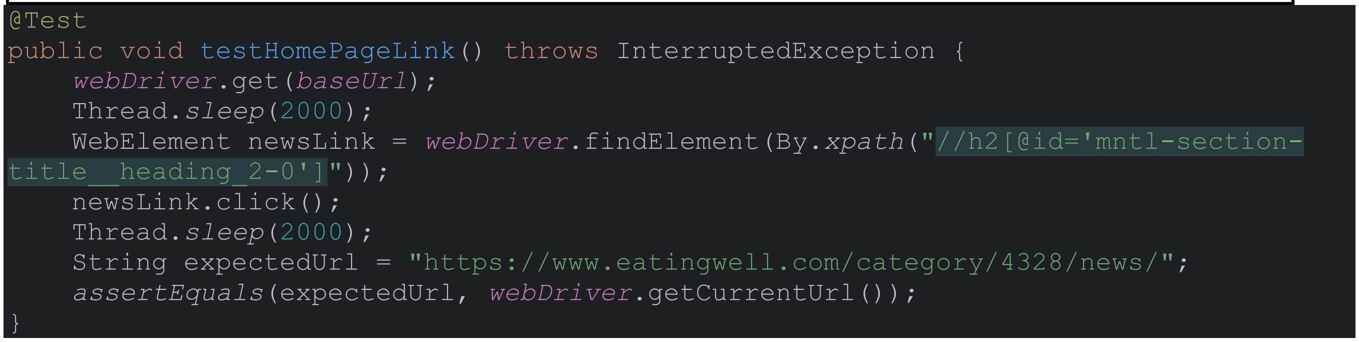
This test scenario is about homepage of our page. We tested everything related to it including: advertisements, alt texts for images for people with visual impairments, logo, navbar links and other.

Test Name: Test homepage advertisement				
Description: Check if the advertisement on the homepage is displayed correctly.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the advertisement element on the homepage. 3. Verify if the advertisement is displayed.	**Test Data:**	**Expected Result:** The advertisement should be displayed on the homepage.	**Actual Result:** The advertisement was displayed on the homepage.	**Status:** PASS
Notes:				

```java
@Test
public void testHomePageAdvertisement() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement advertisement =
webDriver.findElement(By.xpath("//*[@id=\"google_ads_iframe_3865/ddm.eatingwell.com/tier1/taxonomy/homepage_0__container__\"]"));
    assertTrue(advertisement.isDisplayed(), "There is no advertisement.");
}
```

Test Name: Test homepage link

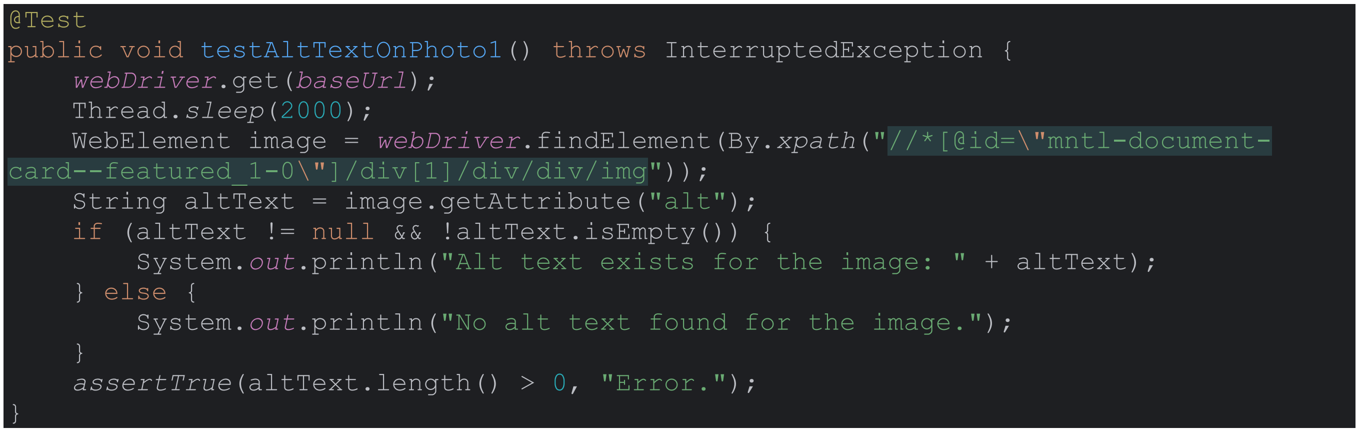
Description: Check if clicking the "News" link on the homepage redirects to the correct page.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the “News” link on the homepage. 3. Click on the “News” link. 4. Wait for the new page to load. 5. Verify if the current URL is correct.	**Test Data:**	**Expected Result:** The page should redirect to the correct news category URL.	**Actual Result:** The page redirected to the correct news category URL.	**Status:** PASS
Notes:				



```
@Test
public void testHomePageLink() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement newsLink = webDriver.findElement(By.xpath("//h2[@id='mntl-section-title__heading_2-0']"));
    newsLink.click();
    Thread.sleep(2000);
    String expectedUrl = "https://www.eatingwell.com/category/4328/news/";
    assertEquals(expectedUrl, webDriver.getCurrentUrl());
}
```

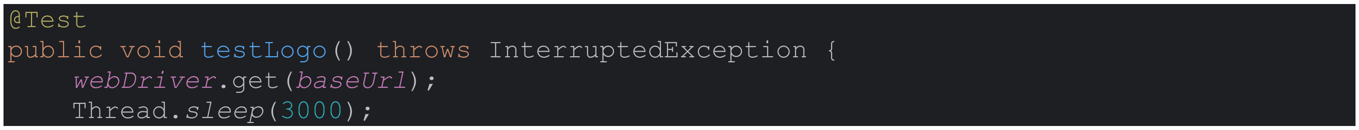
Test Name: Test the alt text on the photo				
Description: Check if the image on the page has an alt text attribute.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the first image. 3. Retrieve the alt attribute of the image. 4. Check if the alt text exists. 5. Verify if the alt text is	**Test Data:**	**Expected Result:** The image should have a non-empty alt text attribute.	**Actual Result:** The image has an alt text attribute.	**Status:** PASS

not empty.				
Notes: There are multiple similar tests in the project that perform the same actions, just on different photos.				



```
@Test
public void testAltTextOnPhoto1() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement image = webDriver.findElement(By.xpath("//*[@id=\"mntl-document-card--featured_1-0\"]/div[1]/div/div/img"));
    String altText = image.getAttribute("alt");
    if (altText != null && !altText.isEmpty()) {
        System.out.println("Alt text exists for the image: " + altText);
    } else {
        System.out.println("No alt text found for the image.");
    }
    assertTrue(altText.length() > 0, "Error.");
}
```

Test Name: Test logo				
Description: Check if clicking the logo on the homepage redirects to the correct URL.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the logo and click on it. 3. Wait for the page to load after the click. 4. Verify if the current URL is the homepage URL.	**Test Data:**	**Expected Result:** The user should be redirected to the homepage URL.	**Actual Result:** The page redirects to the correct homepage URL.	**Status:** PASS
Notes:				



```
@Test
public void testLogo() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement logo = webDriver.findElement(By.xpath("//a[@id='header-logo_1-
0']//*[name()='svg']"));
    logo.click();
    Thread.sleep(2000);
    assertEquals("https://www.eatingwell.com/", webDriver.getCurrentUrl());
```

Test Name: Test the alt text on all photos				
Description: Check if all images on the page have alt text and log any missing alt text.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Wait for all photos to load on the page. 3. Retrieve all photos on the page. 4. Loop through each photo and check if alt text exists. 5. Verify that each photo has non-empty alt text. 6. Wait for all photos to load again after scrolling.		All images on the page should have non-empty alt text.	All images on the page have non-empty alt text.	PASS
Notes:				

```
@Test
public void testAltTextOnAllPhotos() throws InterruptedException {
    webDriver.get(baseUrl);
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(10));

wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(By.tagName("img")));
    List<WebElement> images = webDriver.findElements(By.tagName("img"));
    for (WebElement image : images) {
        String altText = image.getAttribute("alt");
        if (altText != null && !altText.isEmpty()) {
            System.out.println("Alt text exists for the image: " + altText);
        } else {
            System.out.println("No alt text found for the image: " +
image.getAttribute("src"));
        }
        assertTrue(altText != null && !altText.isEmpty(),
                "Error: Alt text is missing or empty for image: " +
image.getAttribute("src"));
    }
    ((JavascriptExecutor) webDriver).executeScript("window.scrollTo(0,
document.body.scrollHeight);");
```

```
    Thread.sleep(2000);

wait.until(ExpectedConditions.presenceOfAllElementsLocatedBy(By.tagName("img")));

}
```

Test Name: Test navbar links				
Description: Check if there are links in the navigation bar and if all links have a valid `href` attribute (non-empty and non-null).				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the navigation bar and wait for the links to be present. 3. Verify if there are any links and then verify if each link has a valid href attribute. 4. Print each link's href attribute.	**Test Data:**	**Expected Result:** The navbar should contains links, and all links have valid, non-empty, non-null href attributes.	**Actual Result:** The navbar contains links, and all links have valid, non-empty, non-null href attributes.	**Status:** PASS
Notes:				

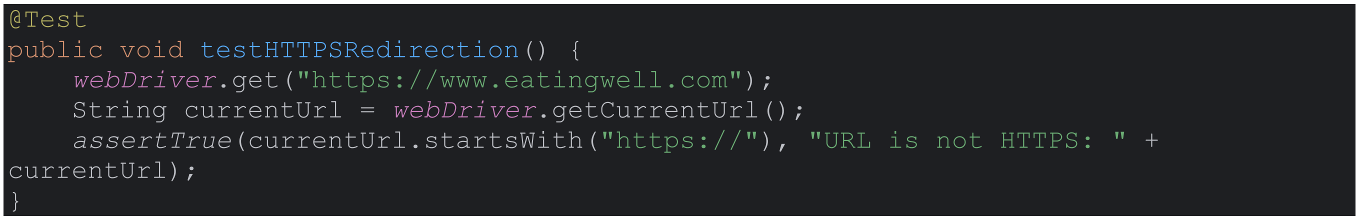
```
@Test
public void testNavbarLinks() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement navbar = webDriver.findElement(By.cssSelector("nav"));
    List<WebElement> links = navbar.findElements(By.tagName("a"));
    assertTrue(links.size() > 0, "There are no links in the navbar.");
    for (WebElement link : links) {
        String href = link.getAttribute("href");
        assertTrue(href != null && !href.isEmpty(), "Link with empty href attribute
found.");
    }
    for (WebElement link : links) {
        System.out.println("Link found: " + link.getAttribute("href"));
    }}
```

3.8. Test Scenario Name: HTTPS test

This test scenario is about enforcement of HTTPS.

Test Name: Test HTTPS redirection
Description: Check if the website redirects to a secure HTTPS connection when

accessed.				
Pre-condition(s):				
Test Steps: 1. Open the browser, navigate to https://www.eatingwell.com and wait for the page to load completely. 2. Retrieve the current URL of the page after the load process is complete. 3. Verify if the URL starts with https://, indicating a secure connection.	**Test Data:**	**Expected Result:** The URL starts with https://, confirming that the website uses a secure connection.	**Actual Result:** The URL https://www.eatingwell.com starts with https://.	**Status:** PASS
Notes:				



```
@Test
public void testHTTPSRedirection() {
    webDriver.get("https://www.eatingwell.com");
    String currentUrl = webDriver.getCurrentUrl();
    assertTrue(currentUrl.startsWith("https://"), "URL is not HTTPS: " +
currentUrl);
}
```

Test Name: Test HTTPS redirects to HTTPS				
Description: Check if navigating to the HTTP version of the website redirects the user to the HTTPS version.				
Pre-condition(s):				
Test Steps: 1. Open the browser. 2. Navigate to the HTTP version of the website. 3. Retrieve the current URL of the page after the redirection. 4. Verify if the current	**Test Data:**	**Expected Result:** The page should redirect from http:// to https://.	**Actual Result:** The page is redirected from http:// to https://.	**Status:** PASS

URL starts with https://, confirming the redirection to a secure HTTPS connection.				
Notes:				

```
@Test
public void testHTTPRedirectsToHTTPS() {
    webDriver.get("http://www.eatingwell.com");
    String currentUrl = webDriver.getCurrentUrl();
    assertTrue(currentUrl.startsWith("https://"), "HTTP did not redirect to HTTPS:
" + currentUrl);
}
```

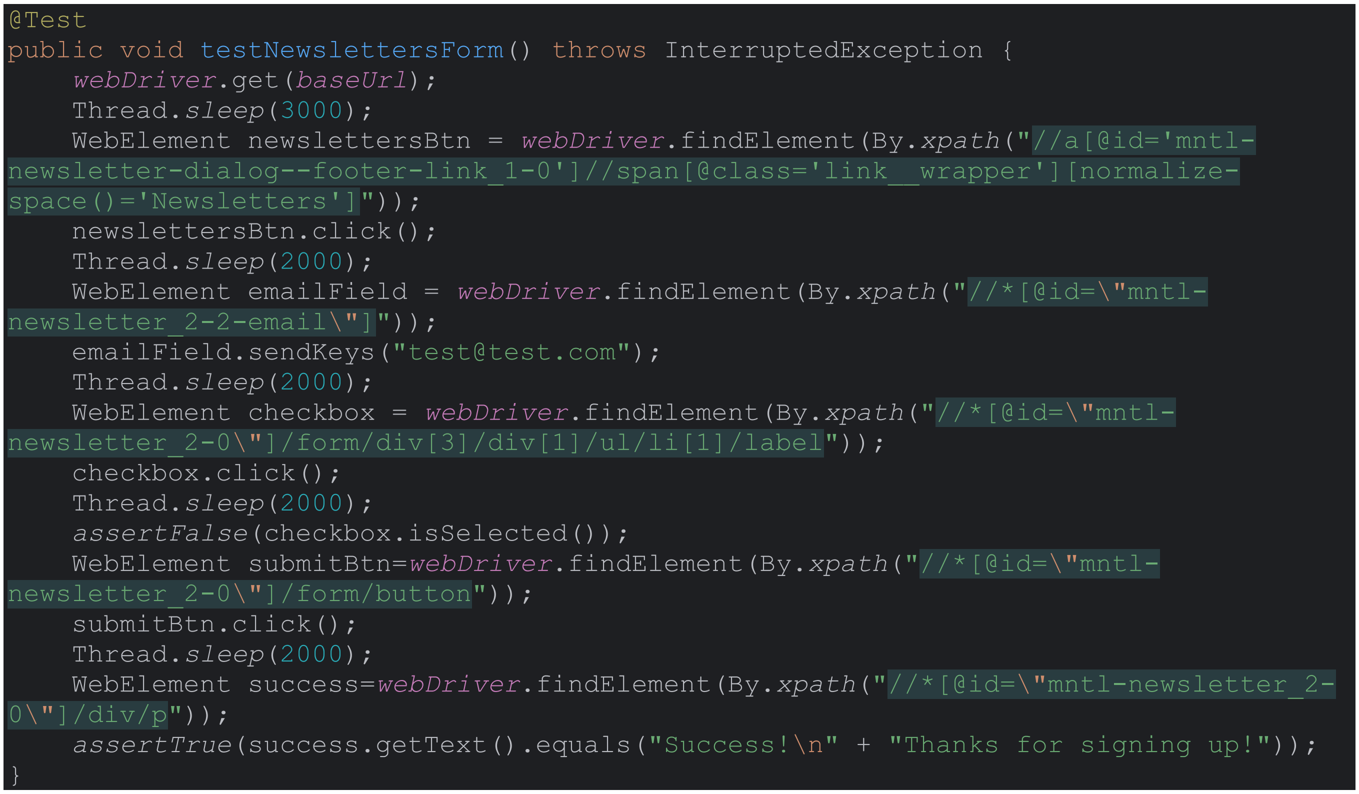
3.9. Test Scenario Name: Newsletters test

This test scenario is about Newsletters pop up because it is a part of critical functionality of this page.

Test Name: Test newsletters button				
Description: Check if the "Newsletters" button in the footer opens the newsletter subscription form and if the form displays the correct heading.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the “Newsletters” button and click it. 3. Wait for the newsletter subscription form to appear. 4. Verify if the heading of the newsletter subscription form is “Newsletter Subscriptions”.	**Test Data:**	**Expected Result:** The page should display the "Newsletter Subscriptions" heading in the subscription form.	**Actual Result:** The page displays the "Newsletter Subscriptions" heading in the subscription form.	**Status:** PASS
Notes:				

```
@Test
public void testNewslettersBtn() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement newslettersBtn = webDriver.findElement(By.xpath("//a[@id='mntl-
newsletter-dialog--footer-link_1-0']//span[@class='link__wrapper'][normalize-
space()='Newsletters']"));
    newslettersBtn.click();
    Thread.sleep(2000);
    WebElement newslettersSubscription =
webDriver.findElement(By.xpath("//*[@id=\"mntl-newsletter_2-
0\"]/form/div[3]/div[1]/div"));
    assertTrue(newslettersSubscription.getText().equals("Newsletter
Subscriptions"));
}
```

Test Name: Test newsletters form				
Description: Check if the newsletter form correctly handles filling out an email, unchecking the checkbox, submitting the form, and displaying the success message.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Find the “Newsletters” button and click it. 3. Wait for the newsletter form to appear. 4. Enter a valid email address and wait for the checkbox to be visible and then click it. 5. Verify if the checkbox is unchecked after clicking it. 6. Click the submit button and wait for the success message to appear. 7. Verify that the success message is “Success! Thanks for signing up!”	**Test Data:** Email address: test@test.com Checkbox status	**Expected Result:** The checkbox should remain unchecked after clicking it, and the success message should display "Success! Thanks for signing up!".	**Actual Result:** The checkbox remains unchecked after clicking it, and the success message displays "Success! Thanks for signing up!".	**Status:** PASS
Notes:				



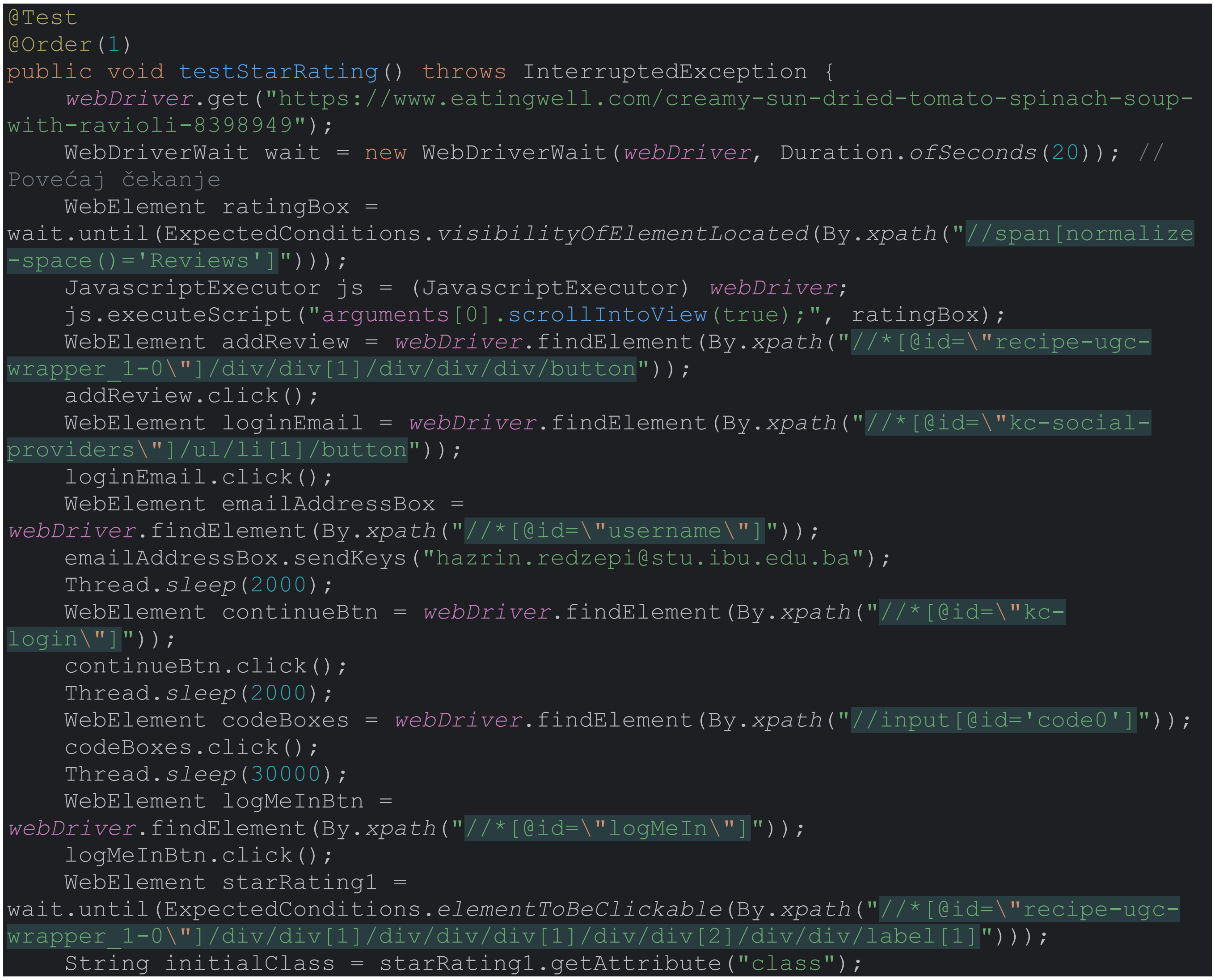
```
@Test
public void testNewslettersForm() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(3000);
    WebElement newslettersBtn = webDriver.findElement(By.xpath("//a[@id='mntl-newsletter-dialog--footer-link_1-0']//span[@class='link__wrapper'][normalize-space()='Newsletters']"));
    newslettersBtn.click();
    Thread.sleep(2000);
    WebElement emailField = webDriver.findElement(By.xpath("//*[@id=\"mntl-newsletter_2-2-email\"]"));
    emailField.sendKeys("test@test.com");
    Thread.sleep(2000);
    WebElement checkbox = webDriver.findElement(By.xpath("//*[@id=\"mntl-newsletter_2-0\"]/form/div[3]/div[1]/ul/li[1]/label"));
    checkbox.click();
    Thread.sleep(2000);
    assertFalse(checkbox.isSelected());
    WebElement submitBtn=webDriver.findElement(By.xpath("//*[@id=\"mntl-newsletter_2-0\"]/form/button"));
    submitBtn.click();
    Thread.sleep(2000);
    WebElement success=webDriver.findElement(By.xpath("//*[@id=\"mntl-newsletter_2-0\"]/div/p"));
    assertTrue(success.getText().equals("Success!\n" + "Thanks for signing up!"));
}
```

3.10. Test Scenario Name: Review test

This test scenario is about Reviews. Adding them, rating, editing and other actions related to it.

Test Name: Test star rating for a recipe				
Description: Check if the user can successfully log in and submit a star rating for a recipe.				
Pre-condition(s): User is registered but not logged in for this test.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Wait for the "Reviews" section to appear. 3. Scroll to the "Reviews" section and click the "add Review" button. 4. Click the email login button and enter the	Email address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The star rating system should update the clicked star after the user selects a star.	The star rating system updates the clicked star confirming that the rating functionality works.	PASS

email then click “Continue”. 5. Wait for the code input field and click on it, then wait for the code to load and click “Log me In”. 6. Click on the first star to submit a rating. 7. Wait for the rating to appear. 8. Verify the clicked star has class “star-rating__star—active”.				
Notes:				



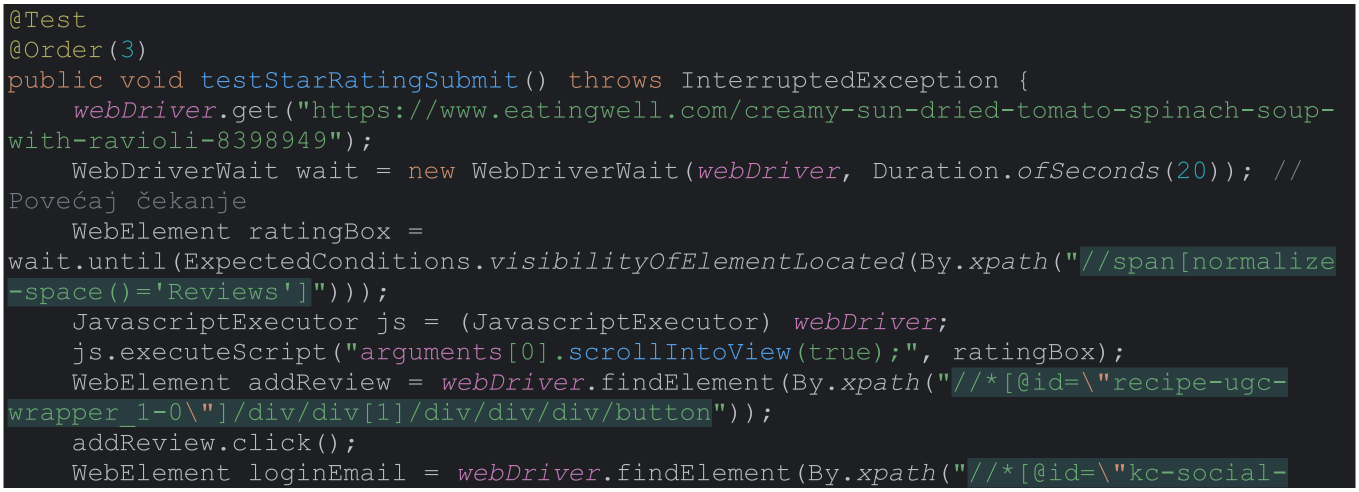
```
@Test
@Order(1)
public void testStarRating() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20)); // Povećaj čekanje
    WebElement ratingBox = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize-space()='Reviews']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);
    WebElement addReview = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-wrapper_1-0\"]/div/div[1]/div/div/div/button"));
    addReview.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox = webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn = webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    WebElement starRating1 = wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id=\"recipe-ugc-wrapper_1-0\"]/div/div[1]/div/div/div[1]/div/div[2]/div/div/label[1]")));
    String initialClass = starRating1.getAttribute("class");
    starRating1.click();
    Thread.sleep(3000);
    assertTrue(starRating1.getAttribute("class").contains("star-rating__star--
active"));
}
```

Test Name: Test star rating text				
Description: Check if the rating text displayed after submitting a review matches the expected value.				
Pre-condition(s): User is registered but not logged in for this test.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Wait for the “Reviews” section to become visible and scroll to the rating box. 3. Click on the “Add Review” button, then click the email login button. 4. Enter email address and click “Continue”. 5. Wait for the verification code field to load and enter the verification, then click “Log Me In”. 6. Wait for the rating stars to be clickable and click the first star to submit a review. 7. Wait for the rating text to be displayed. 8. Verify if the rating text is “Couldn’t eat it”.	Email address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The rating text below the star rating should be displayed as "Couldn't eat it."	The rating text displayed is: "Couldn't eat it."	PASS
Notes:				

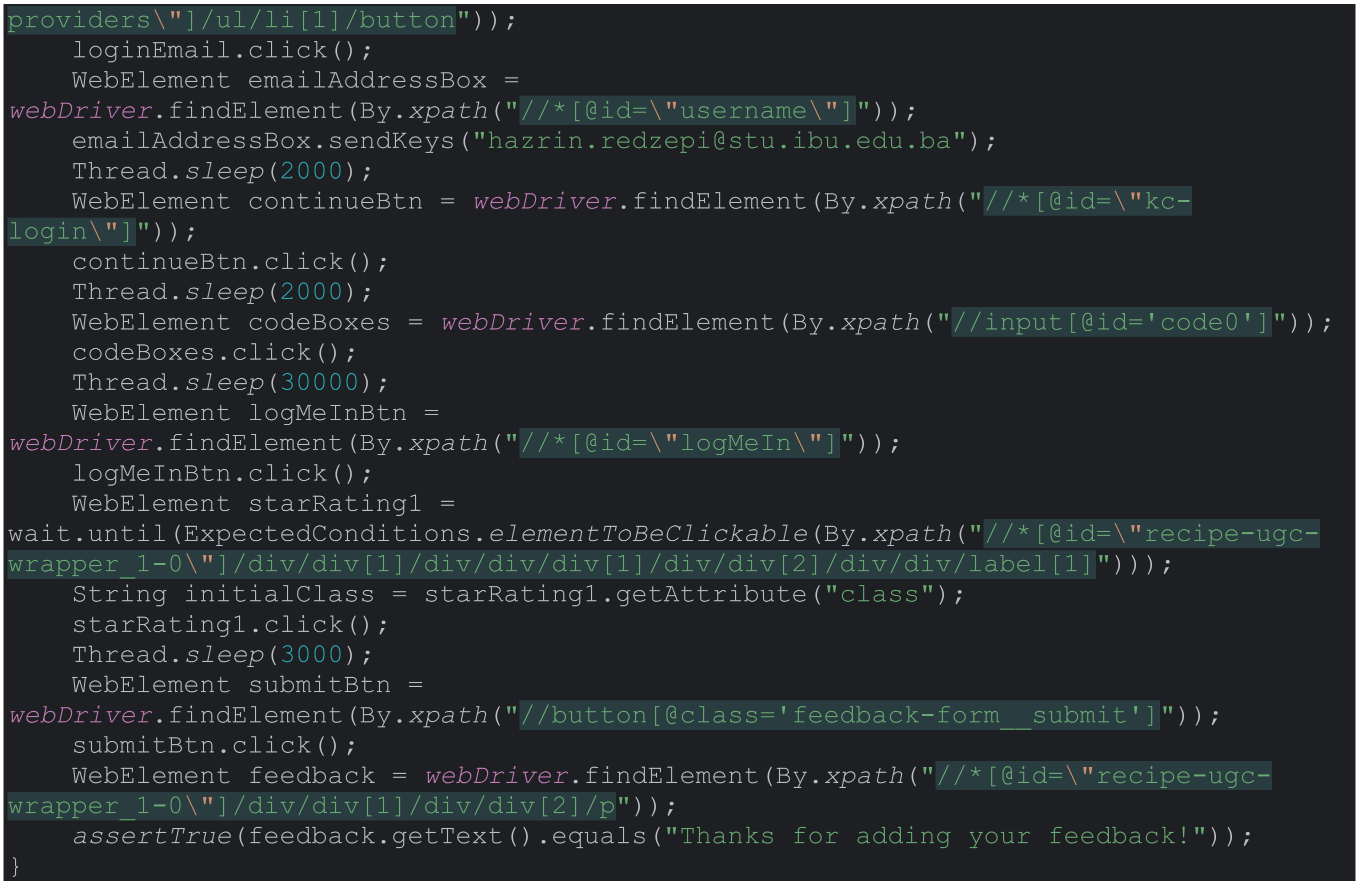
```
@Test
@Order(2)
public void testStarRatingText() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20));
    WebElement ratingBox =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize
-space()='Reviews']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);
    WebElement addReview = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div/div/button"));
    addReview.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(20000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    WebElement starRating1 =
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div[1]/div/div[2]/div/div/label[1]")));
    String initialClass = starRating1.getAttribute("class");
    starRating1.click();
    WebElement ratingText = webDriver.findElement(By.xpath("//span[@class='star-
rating__text']"));
    assertTrue(ratingText.getText().equals("Couldn't eat it"), "Thats not the
actual text message");
}
```

Test Name: Test star rating submit				
Description: Check if the user is able to submit a rating and see the success feedback message.				
Pre-condition(s): User is registered but not logged in.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load. 2. Wait for the “Reviews” section to become visible and scroll to the rating box. 3. Click on the “Add Review” button, then click the email login button. 4. Enter email address and click “Continue”. 5. Wait for the verification code and “Log Me In”. 6. Wait for the rating stars to be clickable and click the first star to submit a review. 7. Wait for the “Submit” button to appear and click it. 8. Wait for the success feedback message to appear. 9. Verify if the feedback message is “Thanks for adding your feedback!”.	Email address: hazrin.redzepi@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The feedback message displayed should be "Thanks for adding your feedback!".	The feedback message displayed is: "Thanks for adding your feedback!".	PASS
Notes:				



```
@Test
@Order(3)
public void testStarRatingSubmit() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20)); //
Povećaj čekanje
    WebElement ratingBox =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize
-space()='Reviews']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);
    WebElement addReview = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div/div/button"));
    addReview.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
```



```
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    WebElement starRating1 =
wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div/div[1]/div/div[2]/div/div/label[1]")));
    String initialClass = starRating1.getAttribute("class");
    starRating1.click();
    Thread.sleep(3000);
    WebElement submitBtn =
webDriver.findElement(By.xpath("//button[@class='feedback-form__submit']"));
    submitBtn.click();
    WebElement feedback = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div[2]/p"));
    assertTrue(feedback.getText().equals("Thanks for adding your feedback!"));
}
```

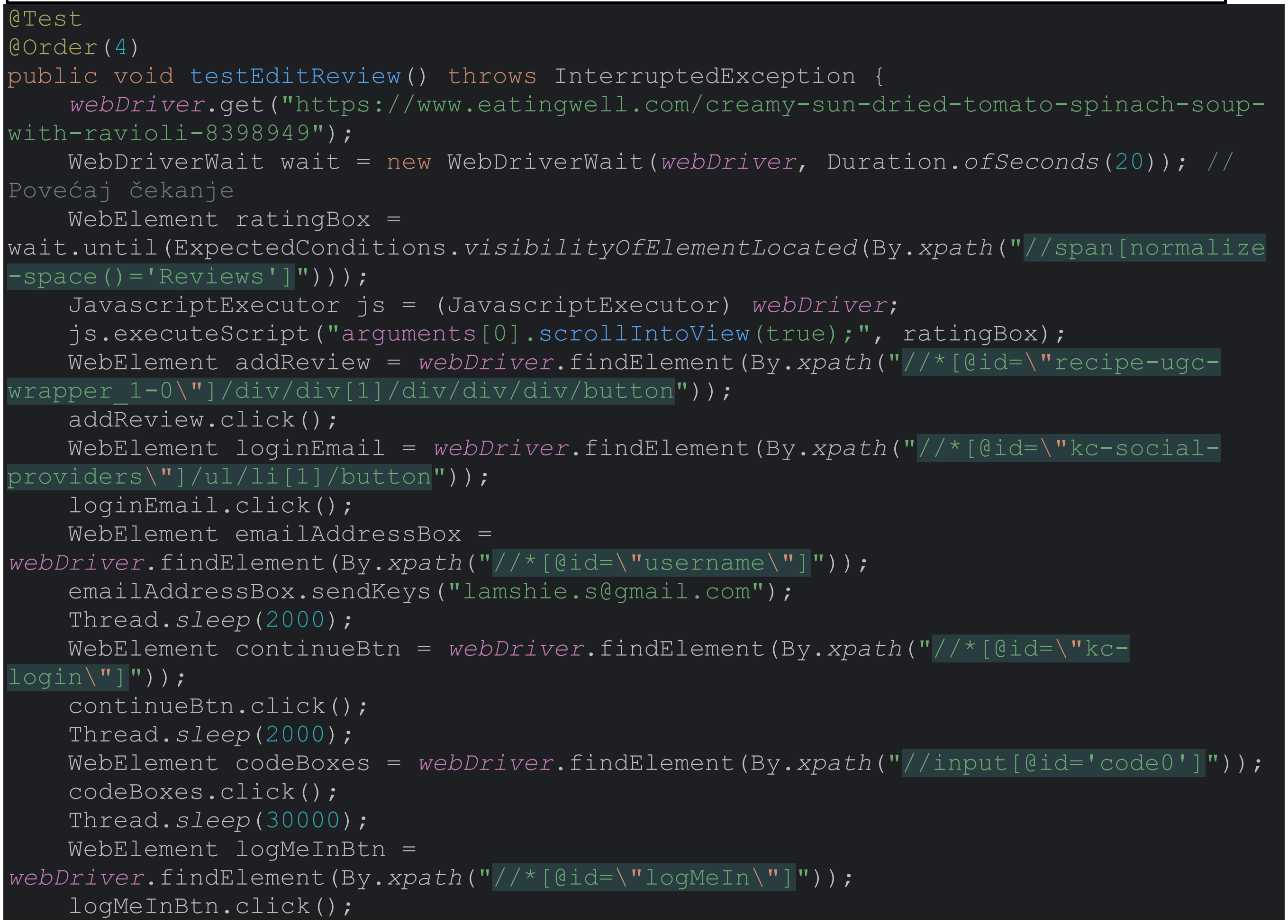
Test Name: Test edit review

Description: Check if the user can edit their review and submit it successfully, verifying the confirmation toast message.

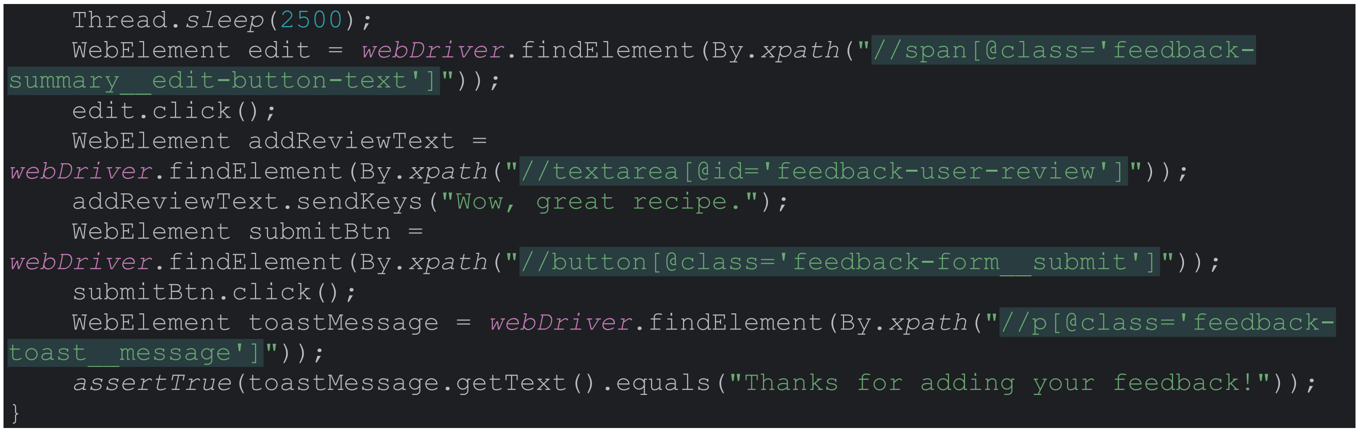
Pre-condition(s): User is registered but not logged in, and a previous review exists for the recipe.

Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:
1. Wait for the page to load. 2. Scroll to the “Reviews” section and wait for it to become visible and click on the “Add Review” button. 3. Click the email login button and enter email address, then click “Continue”. 4. Wait for the	Email address: lamshie.s@gmail.com Verification Code: A valid code provided during the test	The user should be able to successfully edit their review, and a toast message "Thanks for adding your feedback!" should appear after submission.	The user successfully edited their review, and the toast message "Thanks for adding your feedback!" appeared as expected.	PASS

verification code and click “Log Me In”. 5. Wait for the page to reload and display the review options. 6. Locate and click the “Edit Review” button. 7. Enter the updated review text and click the “Submit” button to save the updated review. 8. Wait for the confirmation toast message to appear. 9. Verify the toast message text is “Thanks for adding your feedback!”.				
Notes:				



```
@Test
@Order(4)
public void testEditReview() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/creamy-sun-dried-tomato-spinach-soup-
with-ravioli-8398949");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20)); //
Povećaj čekanje
    WebElement ratingBox =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize
-space()='Reviews']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);
    WebElement addReview = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div/div/button"));
    addReview.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("lamshie.s@gmail.com");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
```



```
    Thread.sleep(2500);
    WebElement edit = webDriver.findElement(By.xpath("//span[@class='feedback-summary__edit-button-text']"));
    edit.click();
    WebElement addReviewText =
webDriver.findElement(By.xpath("//textarea[@id='feedback-user-review']"));
    addReviewText.sendKeys("Wow, great recipe.");
    WebElement submitBtn =
webDriver.findElement(By.xpath("//button[@class='feedback-form__submit']"));
    submitBtn.click();
    WebElement toastMessage = webDriver.findElement(By.xpath("//p[@class='feedback-toast__message']"));
    assertTrue(toastMessage.getText().equals("Thanks for adding your feedback!"));
}
```

<table>
<tr><td colspan="5"><b>Test Name: Test adding review without stars</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if the user is unable to submit a review without selecting a star rating.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b> User is registered but not logged in.</td></tr>
<tr><td><b>Test Steps:</b>
1. Wait for the page to load.
2. Scroll to the “Reviews” section and click the “Add Review” button.
3. Click the email login button and enter email address, then click “Continue”.
4. Wait for the verification code and click “Log Me In”.
5. Wait for the page to reload and display the review options.
6. Enter review text without selecting a star rating.
7. Verify the “Submit” button is disabled and no review is submitted.</td><td><b>Test Data:</b>
Email address: hazrin.redzepi@stu.ibu.edu.ba
Verification Code: A valid code provided during the test
Review text: “This is so yummy!”</td><td><b>Expected Result:</b>
The user should not be able to submit a review without selecting a star rating. The submit button should be disabled, preventing submission.</td><td><b>Actual Result:</b>
The submit button is disabled, and the user cannot submit the review without selecting a star rating, as expected.</td><td><b>Status:</b>
PASS</td></tr>
<tr><td colspan="5"><b>Notes:</b></td></tr>
</table>

```
@Test
@Order(5)
public void testAddingReviewWithoutStars() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com/baked-feta-tomato-spaghetti-squash-
8407035");
    WebDriverWait wait = new WebDriverWait(webDriver, Duration.ofSeconds(20)); //
Povećaj čekanje
    WebElement ratingBox =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//span[normalize
-space()='Reviews']")));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", ratingBox);
    WebElement addReview = webDriver.findElement(By.xpath("//*[@id=\"recipe-ugc-
wrapper_1-0\"]/div/div[1]/div/div/div/button"));
    addReview.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
    emailAddressBox.sendKeys("hazrin.redzepi@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2500);
    WebElement addReviewText =
webDriver.findElement(By.xpath("//textarea[@id='feedback-user-review']"));
    addReviewText.sendKeys("This is so yummy!");
    WebElement submitBtn =
webDriver.findElement(By.xpath("//button[@class='feedback-form__submit']"));
    submitBtn.click();
    Thread.sleep(2000);
    assertTrue(submitBtn.getAttribute("disabled") != null, "Button is not
disabled");
    assertFalse(submitBtn.isEnabled(),"Button is clickable when it shouldn't be");

}
```

3.11. Test Scenario Name: myRecipes Registration test

This test scenario is about our second part of the project and that is a registration test for myRecipes page.

Test Name: Test registration
Description: Check if the success message is displayed, confirming the successful

registration.				
Pre-condition(s):				
Test Steps: 1. Wait for the page to load. 2. Click on the login modal icon and click “Join Now” to start registration. 3. Click “Sign up with Email” button and enter the email address, then click “Join Now” button. 4. Wait for the verification code and then click the code. 5. Click “Log me in” and then click “My Recipes”. 6. Verify the “Account created. Welcome!” message.	**Test Data:** Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	**Expected Result:** The user should successfully register and be greeted with the message "Account created. Welcome!".	**Actual Result:** The user successfully registered and was greeted with the message "Account created. Welcome!".	**Status:** PASS
Notes:				

```
@Test
public void testRegistration() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com");
    Thread.sleep(2000);
    WebElement myRecipes = webDriver.findElement(By.xpath("//div[@id='mntl-utility-
nav_1-0']//a[@aria-label='Trigger interstitial modal to log in to
MyRecipes']//*[name()='svg']"));
    myRecipes.click();
    Thread.sleep(2000);
    WebElement joinNowBtn = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-
interstitial__content_1-0\"]/div[1]/a"));
    joinNowBtn.click();
    Thread.sleep(2000);
    WebElement SignUpWithEmailBtn =
webDriver.findElement(By.xpath("//span[normalize-space()='Sign up with Email']"));
    SignUpWithEmailBtn.click();
    Thread.sleep(2000);
    WebElement emailAddressBtn =
webDriver.findElement(By.xpath("//*[@id=\"email\"]"));
    emailAddressBtn.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement joinNowBtn1 = webDriver.findElement(By.xpath("//input[@id='kc-
register']"));
```

```
    joinNowBtn1.click();
    Thread.sleep(3000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//*[@id=\"code0\"]"));
    codeBoxes.click();
    Thread.sleep(20000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(3000);
    WebElement myRecipes2 = webDriver.findElement(By.xpath("//*[@id=\"mntl-utility-
nav_1-0\"]/ul/li[8]/button"));
    myRecipes2.click();
    Thread.sleep(2000);
    WebElement registrationSuccessfulMessage =
webDriver.findElement(By.xpath("/html/body/div[3]/span"));
    assertEquals("Account created. Welcome!",
registrationSuccessfulMessage.getText(), "Error");
}
```

Test Name: Test register with email which already exists				
Description: Check if an error message is displayed when trying to register with an email that already exists in the system.				
Pre-condition(s):				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Click on the “Join for Free” button, then click on “Sign up with Email” button and enter email address. 3. Click “Join Now”. 4. Wait for the error message to appear. 5. Verify that the error message contains “Email already exists.”	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The user should see an error message that says "Email already exists."	The user saw an error message that said "Email already exists."	PASS
Notes:				

```
@Test
public void testEmailAlreadyExists() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement joinForFreeBtn =
webDriver.findElement(By.xpath("//*[@id=\"homepage_1-0\"]/div[1]/a"));
    joinForFreeBtn.click();
    Thread.sleep(2000);
    WebElement signUpEmail=webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    signUpEmail.click();
    Thread.sleep(2000);
    WebElement emailBox=webDriver.findElement(By.xpath("//*[@id=\"email\"]"));
    emailBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement joinNowBtn=webDriver.findElement(By.xpath("//*[@id=\"kc-
register\"]"));
    joinNowBtn.click();
    Thread.sleep(2000);
    WebElement errorMessage=webDriver.findElement(By.xpath("//*[@id=\"input-
error\"]/span"));
    assertTrue(errorMessage.getText().contains("Email already exists."));
}
```

3.12. Test Scenario Name: myRecipes Login test

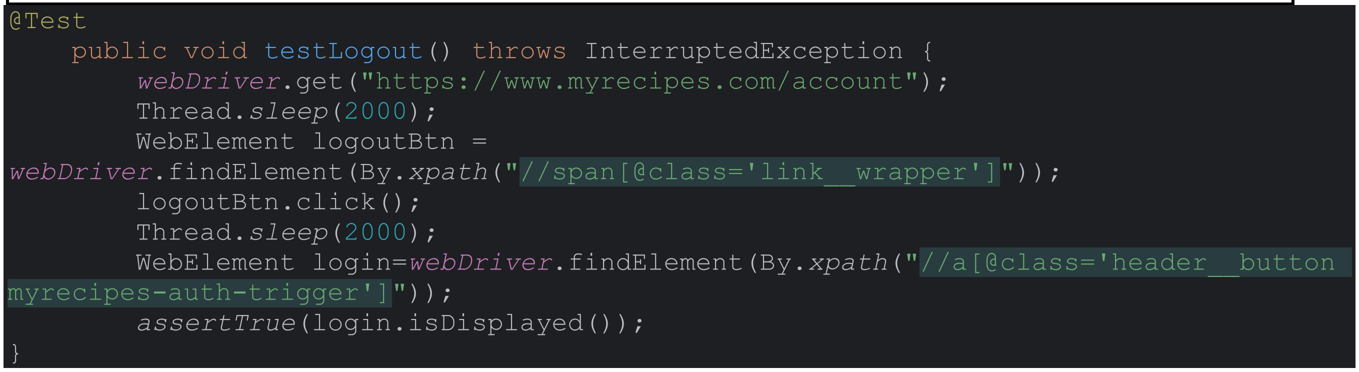
This test scenario is about login and it’s funtionalities.

<table>
<tr><td colspan="5"><b>Test Name: Test login</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if a user can successfully log in and navigate to the favorites page.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b> The user must have a registered account</td></tr>
<tr><td><b>Test Steps:</b>
1. Wait for the page to load.
2. Locate and click the login icon to open the login modal, then select the “Login with Email” option.
3. Enter a email address and click the “Continue” button.
4. Wait for the verification code field to appear.
5. Enter the verification code in the provided fields and click the “Log Me In” button.</td><td><b>Test Data:</b></td><td><b>Expected Result:</b>
The user should successfully log in and be redirected to the favorites page</td><td><b>Actual Result:</b>
The user successfully logged in and was redirected to the favorites page</td><td><b>Status:</b>
PASS</td></tr>
</table>

6. Verify that the success message “Logged in. Welcome!” is displayed.				
Notes:				

```
@Test
public void testLogin() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com");
    WebElement myRecipes = webDriver.findElement(By.xpath("//div[@id='mntl-utility-nav_1-0']//a[@aria-label='Trigger interstitial modal to log in to MyRecipes']//*[name()='svg']"));
    myRecipes.click();
    Thread.sleep(2000);
    WebElement loginBtn = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-interstitial__content_1-0\"]/div[1]/div[2]/a"));
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement loginSuccessfulMessage =
webDriver.findElement(By.xpath("/html/body/div[3]/span"));
    assertEquals("Logged in. Welcome!", loginSuccessfulMessage.getText(), "Error");
    WebElement myRecipes2 = webDriver.findElement(By.xpath("//*[@id=\"mntl-utility-nav_1-0\"]/ul/li[8]/button"));
    myRecipes2.click();
    Thread.sleep(2000);
    WebElement gotofavoritesBtn=webDriver.findElement(By.xpath("//*[@id=\"mntl-myr-nav-menu_1-0\"]/div/div/a"));
    gotofavoritesBtn.click();
    Thread.sleep(3000);
    assertEquals("https://www.myrecipes.com/", webDriver.getCurrentUrl(), "Error");
}
```

Test Name: Test logout				
Description: Check if the user is able to log out successfully and is redirected to the login page, where the "Login" button is visible.				
Pre-condition(s): The user must be logged in first.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Click the “Logout” button. 3. Wait for the page to fully load. 4. Verify that the “Login” button is visible, indicating the user has been logged out successfully.		After logging out, the "Login" button should be visible on the page.	The "Login" button is visible after the user successfully logs out.	PASS
Notes:				



```
@Test
    public void testLogout() throws InterruptedException {
        webDriver.get("https://www.myrecipes.com/account");
        Thread.sleep(2000);
        WebElement logoutBtn =
webDriver.findElement(By.xpath("//span[@class='link__wrapper']"));
        logoutBtn.click();
        Thread.sleep(2000);
        WebElement login=webDriver.findElement(By.xpath("//a[@class='header__button
myrecipes-auth-trigger']"));
        assertTrue(login.isDisplayed());
}
```

Test Name: Test login with invalid email				
Description: Check if the system correctly handles invalid email input during login by displaying an appropriate error message.				
Pre-condition(s): The user must have a registered account.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load. 2. Click the “login” button to open the login modal, then select the “Log in with Email” option. 3. Enter an invalid email address and click the “Continue” button. 4. Wait for the error message to load. 5. Verify that the error message “Invalid email. Please try again.” is displayed.	Email address: “invalid email”	The user should see the error message "Invalid email. Please try again." when entering an invalid email.	The error message "Invalid email. Please try again." is displayed when an invalid email is entered.	PASS
Notes:				

```
@Test
    public void testInvalidEmail() throws InterruptedException {
        webDriver.get(baseUrl);
        Thread.sleep(2000);
        WebElement login=webDriver.findElement(By.xpath("//a[@class='header__button
myrecipes-auth-trigger']"));
        login.click();
        Thread.sleep(2000);
        WebElement loginEmail = webDriver.findElement(By.xpath("//span[normalize-
space()='Log in with Email']"));
        loginEmail.click();
        WebElement emailBox =
webDriver.findElement(By.xpath("//*[@id=\"username\"]"));
        emailBox.sendKeys("invalid email");
        Thread.sleep(2000);
        WebElement continueBtn = webDriver.findElement(By.xpath("//*[@id=\"kc-
login\"]"));
        continueBtn.click();
        Thread.sleep(2000);
        WebElement errorMessage=webDriver.findElement(By.xpath("//*[@id=\"input-
error\"]/span"));
        assertTrue(errorMessage.getText().contains("Invalid email. Please try
again."));
}
```

3.13. Test Scenario Name: myRecipes HTTPS Test

This test scenario is about enforcement of HTTPS.

Test Name: Test HTTPS redirection
Description: Check if the website redirects to a secure HTTPS connection when

accessed.				
Pre-condition(s):				
Test Steps: 1. Open the browser, navigate to page and wait for the page to load completely. 2. Retrieve the current URL of the page after the load process is complete. 3. Verify if the URL starts with https://, indicating a secure connection.	**Test Data:**	**Expected Result:** The URL starts with https://, confirming that the website uses a secure connection.	**Actual Result:** The URL https://www.eatingwell.com starts with https://.	**Status:** PASS
Notes:				

```
@Test
public void testHTTPSRedirection() {
    webDriver.get("https://www.myrecipes.com/");
    String currentUrl = webDriver.getCurrentUrl();
    assertTrue(currentUrl.startsWith("https://"), "URL is not HTTPS: " +
currentUrl);
}
```

Test Name: Test HTTP redirects to HTTPS				
Description: Check if navigating to the HTTP version of the website redirects the user to the HTTPS version.				
Pre-condition(s):				
Test Steps: 1. Open the browser. 2. Navigate to the HTTP version of the website. 3. Retrieve the current URL of the page after the redirection. 4. Verify if the current URL starts with	**Test Data:**	**Expected Result:** The page should redirect from http:// to https://.	**Actual Result:** The page is redirected from http:// to https://.	**Status:** PASS

https://, confirming the redirection to a secure HTTPS connection.				
Notes:				

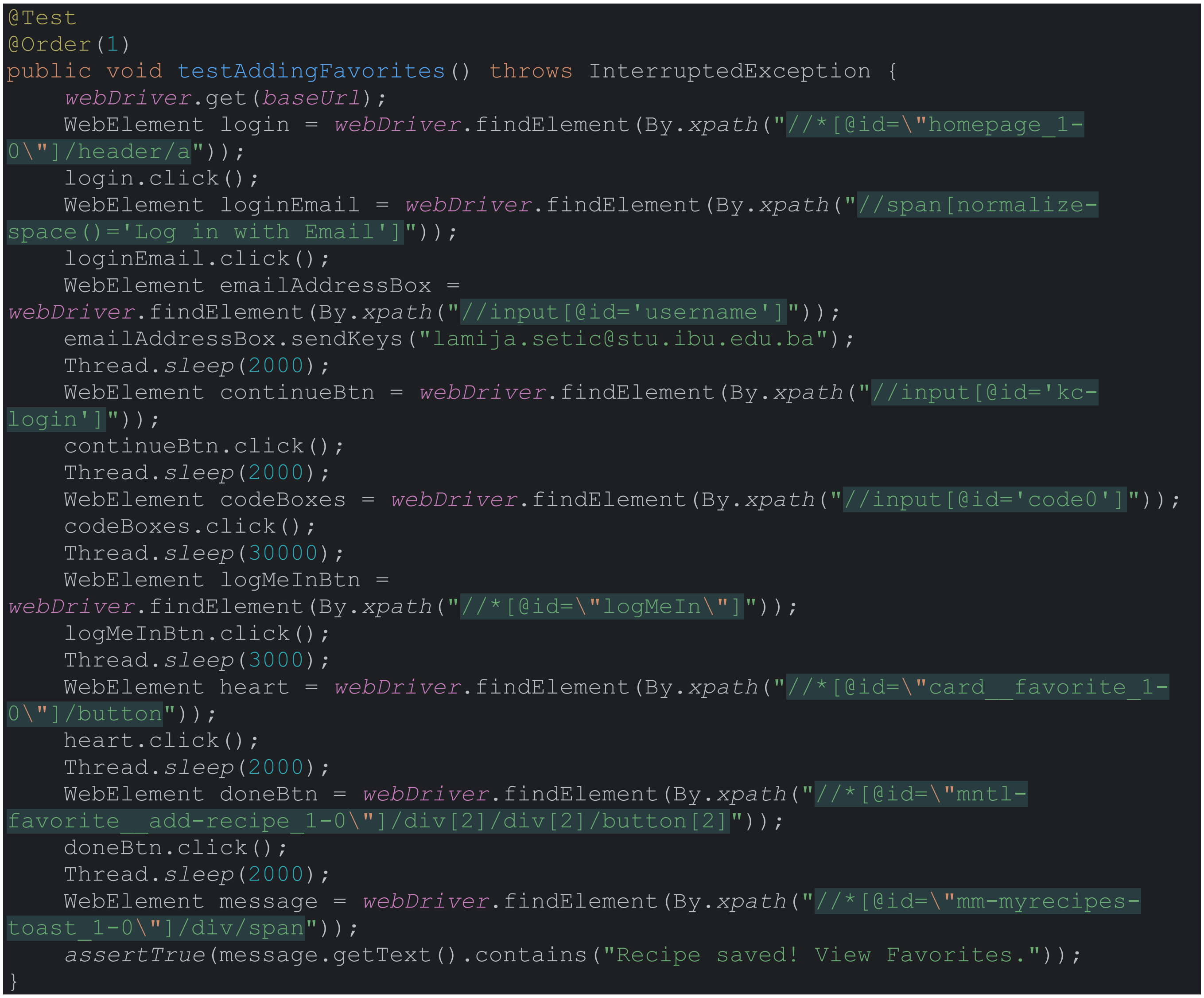
```
@Test
public void testHTTPRedirectsToHTTPS() {
    webDriver.get("https://www.myrecipes.com/");
    String currentUrl = webDriver.getCurrentUrl();
    assertTrue(currentUrl.startsWith("https://"), "HTTP did not redirect to HTTPS:
" + currentUrl);
}
```

3.14. Test Scenario Name: myRecipes Favorites test

This test scenario is about testing favorites on myRecipes page. That’s the most important functionality of this page and it has many actions to test (add, edit, delete, remove..).

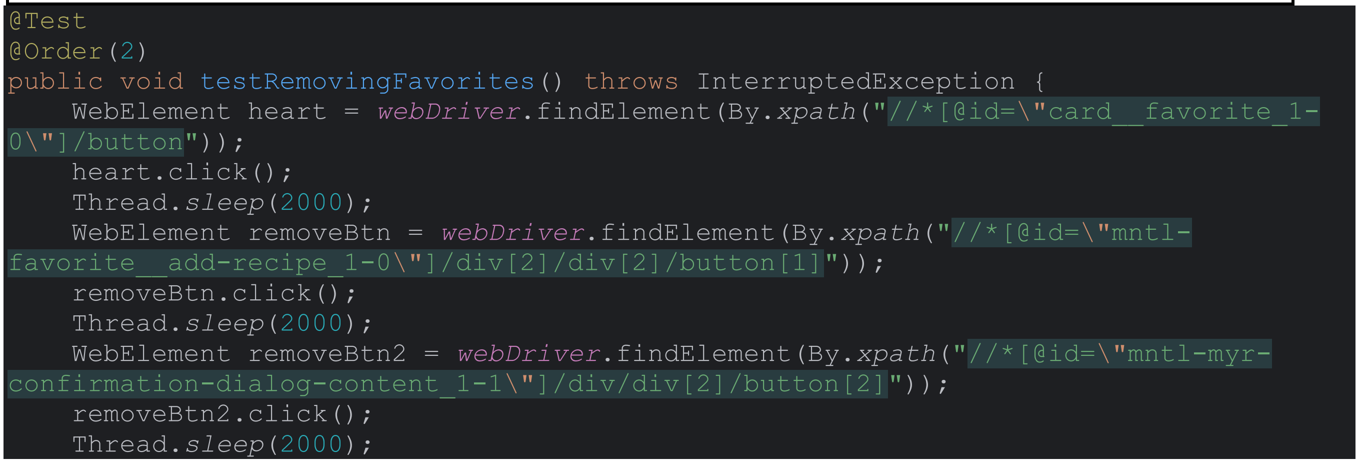
Test Name: Test adding favorites				
Description: Check if the user can successfully log in and add a recipe to their favorites, confirming the success message is displayed.				
Pre-condition(s): The user must have a registered account.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Wait for the page to load. 2. Click the “Login” button to open the login modal. 3. Wait for the login options to load and select the “Log in with Email” option. 4. Enter the valid email address and click the “Continue” button. 5. Enter the verification code and click the “Login Me In” button. 6. Wait for the page to	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The success message "Recipe saved! View Favorites." should appear after adding the recipe to favorites.	The success message "Recipe saved! View Favorites." is displayed after the recipe is successfully added to favorites.	PASS

load and locate the heart icon of a recipe. 7. Click the heart icon to add the recipe to favorites. 8. Click the “Done” button on the success popup. 9. Verify the success message “Recipe saved! View Favorites.” Is displayed.				
Notes:				



```
@Test
@Order(1)
public void testAddingFavorites() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement login = webDriver.findElement(By.xpath("//*[@id=\"homepage_1-0\"]/header/a"));
    login.click();
    WebElement loginEmail = webDriver.findElement(By.xpath("//span[normalize-space()='Log in with Email']"));
    loginEmail.click();
    WebElement emailAddressBox = webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn = webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(3000);
    WebElement heart = webDriver.findElement(By.xpath("//*[@id=\"card__favorite_1-0\"]/button"));
    heart.click();
    Thread.sleep(2000);
    WebElement doneBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-favorite__add-recipe_1-0\"]/div[2]/div[2]/button[2]"));
    doneBtn.click();
    Thread.sleep(2000);
    WebElement message = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-toast_1-0\"]/div/span"));
    assertTrue(message.getText().contains("Recipe saved! View Favorites."));
}
```

Test Name: Test removing favorites				
Description: Check if the user can successfully remove a recipe from their favorites, and verify the recipe is no longer displayed in the favorites list.				
Pre-condition(s): User is logged in				
Test Steps: 1. Wait for the page to load. 2. Locate the heart icon of the recipe and click it to remove the recipe from favorites. 3. Wait for the confirmation popup and click the "Remove" button. 4. Confirm the removal by clicking the "Remove" button in the second confirmation dialog. 5. Wait for the page to load and navigate to the "Favorites" section. 6. Verify the recipe is no longer present in the favorites list.	**Test Data:**	**Expected Result:** The recipe should be removed from the favorites list and should not appear in the favorites section.	**Actual Result:** The recipe "Slow-Cooker Kentucky Burgoo" is no longer displayed in the favorites list after being removed.	**Status:** PASS
Notes:				

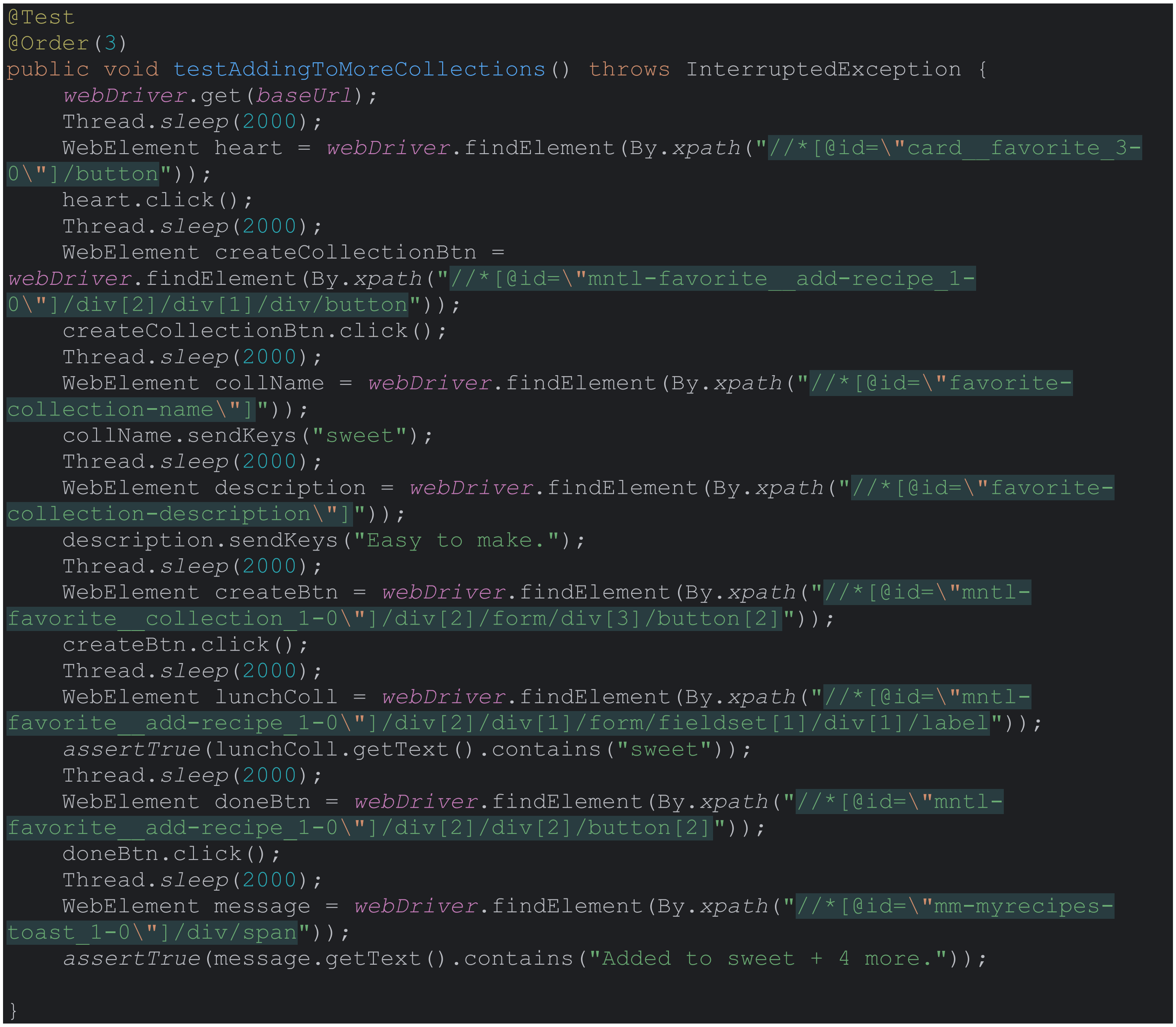


```java
@Test
@Order(2)
public void testRemovingFavorites() throws InterruptedException {
    WebElement heart = webDriver.findElement(By.xpath("//*[@id=\"card__favorite_1-0\"]/button"));
    heart.click();
    Thread.sleep(2000);
    WebElement removeBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-favorite__add-recipe_1-0\"]/div[2]/div[2]/button[1]"));
    removeBtn.click();
    Thread.sleep(2000);
    WebElement removeBtn2 = webDriver.findElement(By.xpath("//*[@id=\"mntl-myr-confirmation-dialog-content_1-1\"]/div/div[2]/button[2]"));
    removeBtn2.click();
    Thread.sleep(2000);
    WebElement favs = webDriver.findElement(By.xpath("//*[@id=\"navigation_1-
0\"]/ul/li[2]/a"));
    favs.click();
    Thread.sleep(2000);
    assertFalse(webDriver.getPageSource().contains("Slow-Cooker Kentucky Burgoo"));
}
```

Test Name: Test adding to more collections				
Description: Check if the recipe is added to a new collection and if the success message appears.				
Pre-condition(s): User is logged in				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load.
2. Click the heart button to add to favorites.
3. Wait for collection options to load.
4. Click "Create Collection".
5. Wait for the collection name field.
6. Enter the name "sweet" and description "Easy to make".
7. Click the "Create" button.
8. Wait for the collection list to update.
9. Verify the new collection name contains "sweet".
10. Click the "Done" button.
11. Wait for the success message.
12 . Verify the success message contains "Added to sweet + 4 more".
	collection=”sweets”
	description=”Easy to make”
		The recipe should be successfully added to a new collection named "sweet", and the success message should display "Added to sweet + 4 more".	The recipe is successfully added to the "sweet" collection, and the success message displays "Added to sweet + 4 more".	PASS

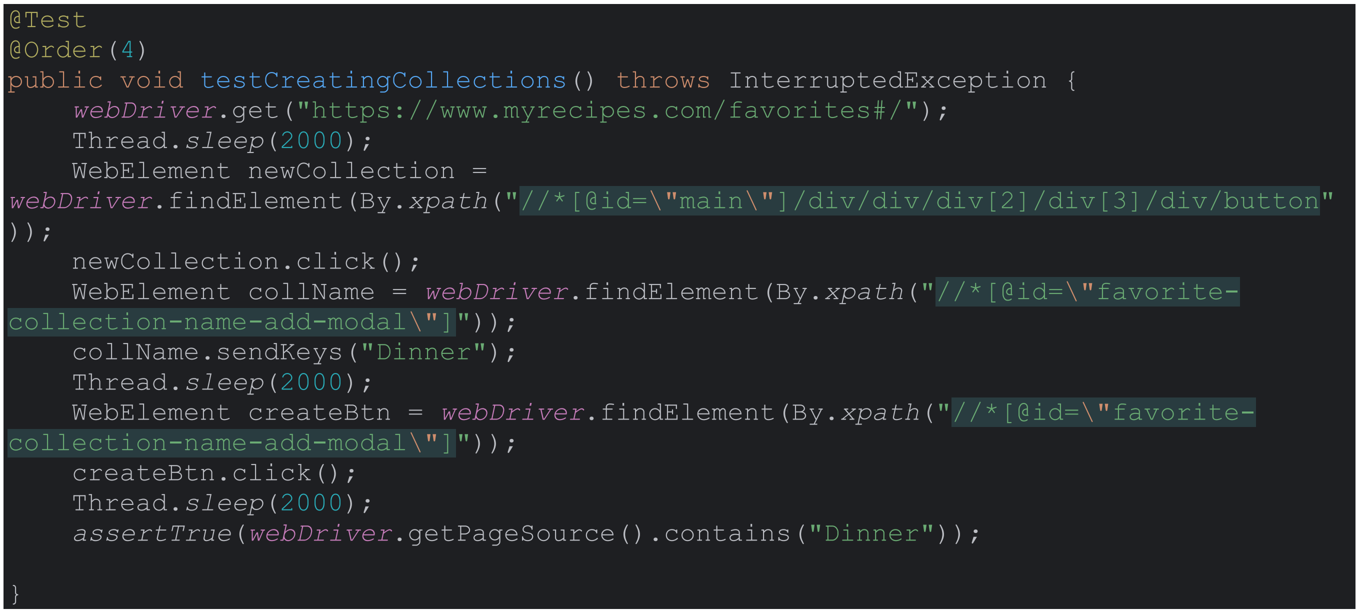
Notes:				



```
@Test
@Order(3)
public void testAddingToMoreCollections() throws InterruptedException {
    webDriver.get(baseUrl);
    Thread.sleep(2000);
    WebElement heart = webDriver.findElement(By.xpath("//*[@id=\"card__favorite_3-
0\"]/button"));
    heart.click();
    Thread.sleep(2000);
    WebElement createCollectionBtn =
webDriver.findElement(By.xpath("//*[@id=\"mntl-favorite__add-recipe_1-
0\"]/div[2]/div[1]/div/button"));
    createCollectionBtn.click();
    Thread.sleep(2000);
    WebElement collName = webDriver.findElement(By.xpath("//*[@id=\"favorite-
collection-name\"]"));
    collName.sendKeys("sweet");
    Thread.sleep(2000);
    WebElement description = webDriver.findElement(By.xpath("//*[@id=\"favorite-
collection-description\"]"));
    description.sendKeys("Easy to make.");
    Thread.sleep(2000);
    WebElement createBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-
favorite__collection_1-0\"]/div[2]/form/div[3]/button[2]"));
    createBtn.click();
    Thread.sleep(2000);
    WebElement lunchColl = webDriver.findElement(By.xpath("//*[@id=\"mntl-
favorite__add-recipe_1-0\"]/div[2]/div[1]/form/fieldset[1]/div[1]/label"));
    assertTrue(lunchColl.getText().contains("sweet"));
    Thread.sleep(2000);
    WebElement doneBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-
favorite__add-recipe_1-0\"]/div[2]/div[2]/button[2]"));
    doneBtn.click();
    Thread.sleep(2000);
    WebElement message = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-
toast_1-0\"]/div/span"));
    assertTrue(message.getText().contains("Added to sweet + 4 more."));

}
```

Test Name: Test creating collections				
Description: Check if a new collection can be created successfully.				
Pre-condition(s): User is logged in				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load. 2. Click the button to create a new collection. 3. Enter "Dinner" as the collection name. 4. Click the button to save the new collection. 5. Wait for the page to update and verify if the collection "Dinner" is visible on the page.	Collection=”Dinner”	The new collection named "Dinner" should be visible on the page.	The new collection named "Dinner" is visible on the page.	PASS
Notes:				



```
@Test
@Order(4)
public void testCreatingCollections() throws InterruptedException {
    webDriver.get("https://www.myrecipes.com/favorites#/");
    Thread.sleep(2000);
    WebElement newCollection =
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[2]/div[3]/div/button"
));
    newCollection.click();
    WebElement collName = webDriver.findElement(By.xpath("//*[@id=\"favorite-
collection-name-add-modal\"]"));
    collName.sendKeys("Dinner");
    Thread.sleep(2000);
    WebElement createBtn = webDriver.findElement(By.xpath("//*[@id=\"favorite-
collection-name-add-modal\"]"));
    createBtn.click();
    Thread.sleep(2000);
    assertTrue(webDriver.getPageSource().contains("Dinner"));

}
```

Test Name: Test recipe link				
Description: Check if clicking on the first recipe in the favorites list redirects to the correct recipe page.				
Pre-condition(s): User is logged in				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Wait for the page to load. 2. Click the first recipe in the list. 3. Wait for the recipe detail dialog to open. 4. Click on the "View Recipe" link. 5. Wait for the page to load. 6. Verify that the current URL matches the expected recipe URL.		The browser should navigate to the URL.	The browser navigated to the URL.	PASS

Notes:

```java
@Test
@Order(5)
public void testRecipeLink() throws InterruptedException {
    webDriver.get("https://www.myrecipes.com/favorites#/");
    Thread.sleep(2000);
    WebElement firstRecipe =
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[2]/div[1]/ul/li[1]/div[2]"));
    firstRecipe.click();
    Thread.sleep(2000);
    WebElement viewRecipe = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-dialog-content_1-0\"]/div/div[1]/div[2]/a"));
    viewRecipe.click();
    Thread.sleep(2000);
    assertEquals("https://www.foodandwine.com/mujadara-8724605",
webDriver.getCurrentUrl());

}
```

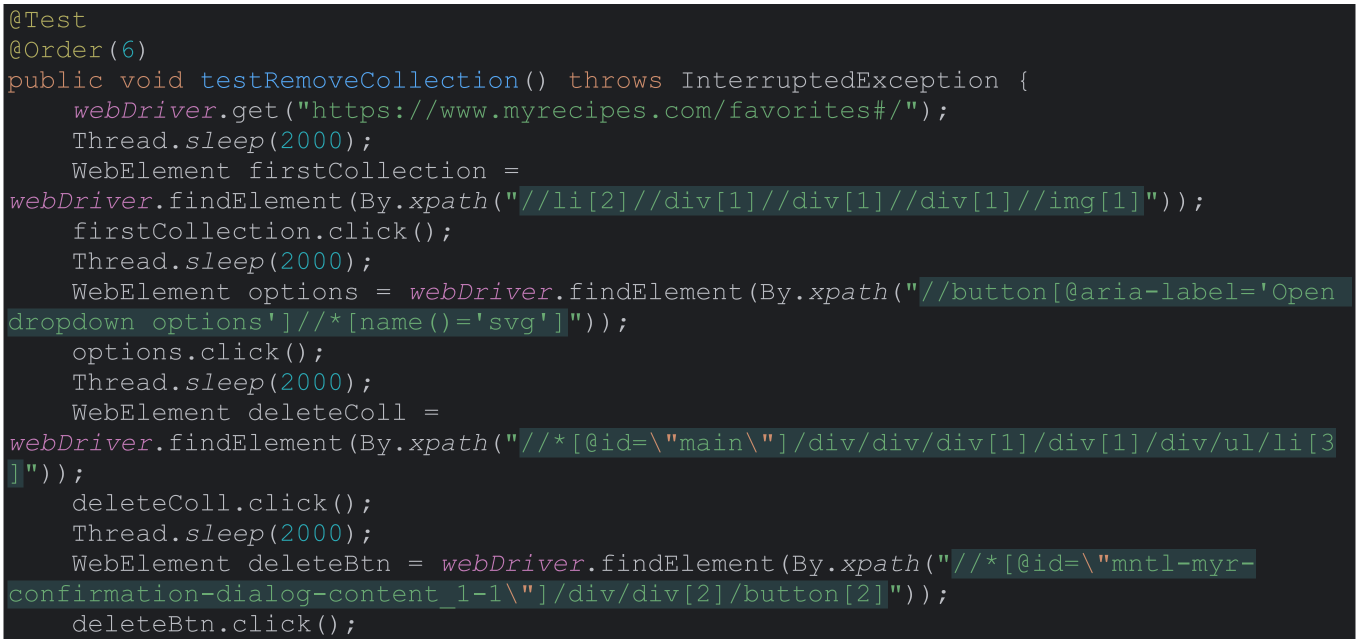
Test Name: Test remove collection

Description: Check if the collection is removed successfully after clicking on the delete option.

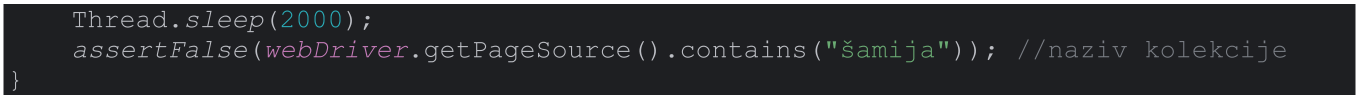
Pre-condition(s): User is logged in and has at least one collection to delete.

Test Steps:	Test Data:	Expected Result:	Actual Result:	Status:

1. Wait for the page to load. 2. Click on the second collection in the list. 3. Wait for the collection details to load. 4. Click on the options button (three dots) for the selected collection. 5. Wait for the options menu to open. 6. Select the "Delete Collection" option. 7. Confirm deletion by clicking the "Delete" button. 8. Wait for the page to load and verify that the collection name is no longer present.		The collection should be removed, and the collection name should no longer appear on the page.	The collection named "šamija" was successfully removed from the favorites list.	PASS
Notes:				



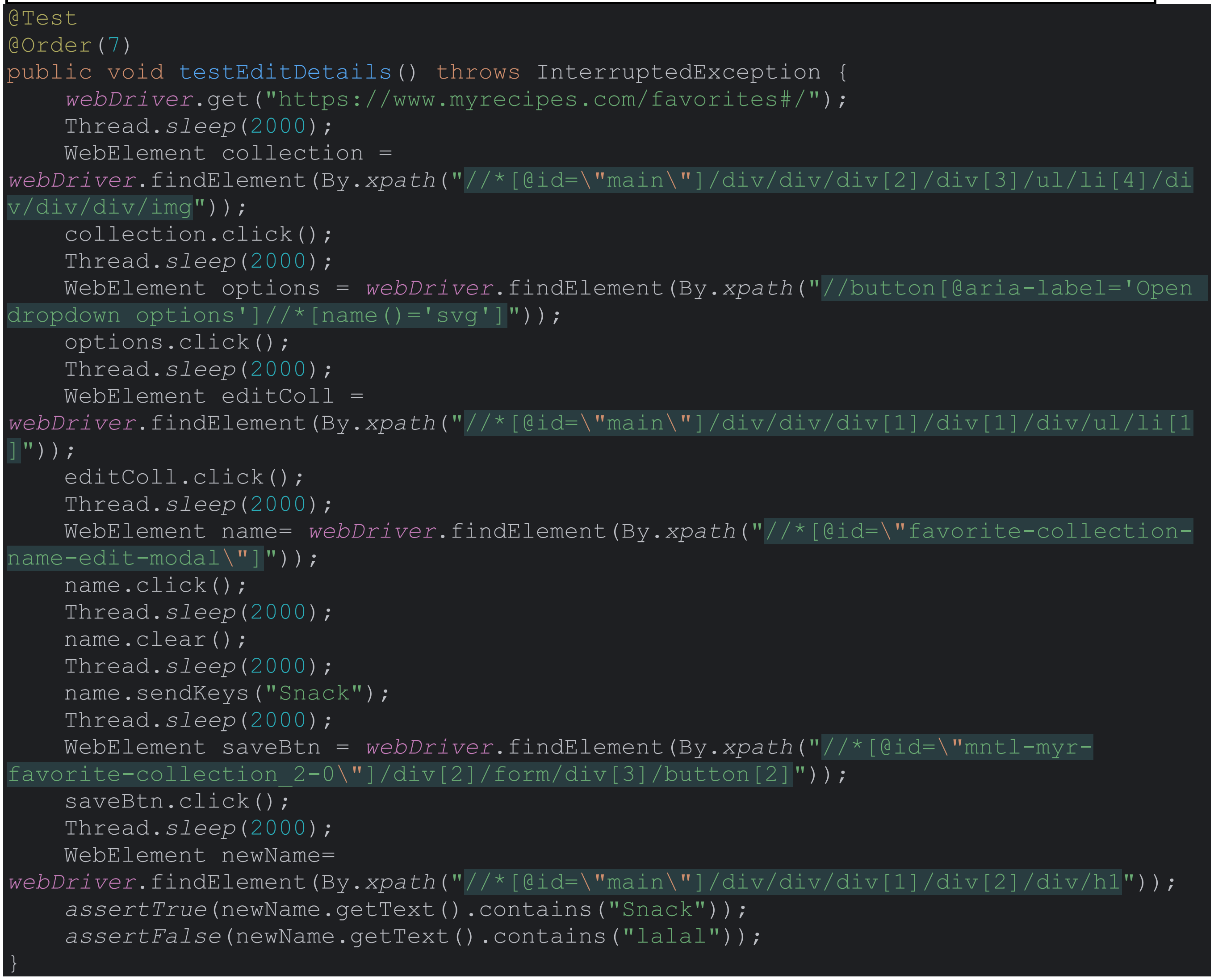
```
@Test
@Order(6)
public void testRemoveCollection() throws InterruptedException {
    webDriver.get("https://www.myrecipes.com/favorites#/");
    Thread.sleep(2000);
    WebElement firstCollection =
webDriver.findElement(By.xpath("//li[2]//div[1]//div[1]//div[1]//img[1]"));
    firstCollection.click();
    Thread.sleep(2000);
    WebElement options = webDriver.findElement(By.xpath("//button[@aria-label='Open
dropdown options']//*[name()='svg']"));
    options.click();
    Thread.sleep(2000);
    WebElement deleteColl =
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[1]/div[1]/div/ul/li[3
]"));
    deleteColl.click();
    Thread.sleep(2000);
    WebElement deleteBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-myr-
confirmation-dialog-content_1-1\"]/div/div[2]/button[2]"));
    deleteBtn.click();
```



```
    Thread.sleep(2000);
    assertFalse(webDriver.getPageSource().contains("šamija")); //naziv kolekcije
}
```

<table>
<tr><td colspan="5"><b>Test Name: Test edit collection details</b></td></tr>
<tr><td colspan="5"><b>Description:</b> Check if the collection details (name) are updated correctly after editing.</td></tr>
<tr><td colspan="5"><b>Pre-condition(s):</b> User is logged in and has at least one collection to edit.</td></tr>
<tr><td><b>Test Steps:</b>
1. Wait for the page to load.
2. Click on the fourth collection in the list.
3. Wait for the collection details to load.
4. Click on the options button (three dots) for the selected collection.
5. Wait for the options menu to open.
6. Select the "Edit Collection" option.
7. Wait for the edit modal to appear.
8. Click on the collection name field and clear the existing name.
9. Enter the new collection name ("Snack").
10. Click on the "Save" button to save the changes.
11. Wait for the page to load and verify that the collection</td><td><b>Test Data:</b>
Collection=”Snack”
Collection “lalal”</td><td><b>Expected Result:</b>
The collection name should be successfully updated to "Snack" and the old name ("lalal") should no longer appear.</td><td><b>Actual Result:</b>
The collection name was updated to "Snack" and "lalal" was removed.</td><td><b>Status:</b>
PASS</td></tr>
</table>

name has been updated to "Snack". 12. Verify that the old collection name ("lalal") no longer appears.				
Notes:				



```
@Test
@Order(7)
public void testEditDetails() throws InterruptedException {
    webDriver.get("https://www.myrecipes.com/favorites#/");
    Thread.sleep(2000);
    WebElement collection =
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[2]/div[3]/ul/li[4]/div/div/div/img"));
    collection.click();
    Thread.sleep(2000);
    WebElement options = webDriver.findElement(By.xpath("//button[@aria-label='Open dropdown options']//*[name()='svg']"));
    options.click();
    Thread.sleep(2000);
    WebElement editColl =
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[1]/div[1]/div/ul/li[1]"));
    editColl.click();
    Thread.sleep(2000);
    WebElement name= webDriver.findElement(By.xpath("//*[@id=\"favorite-collection-name-edit-modal\"]"));
    name.click();
    Thread.sleep(2000);
    name.clear();
    Thread.sleep(2000);
    name.sendKeys("Snack");
    Thread.sleep(2000);
    WebElement saveBtn = webDriver.findElement(By.xpath("//*[@id=\"mntl-myr-favorite-collection_2-0\"]/div[2]/form/div[3]/button[2]"));
    saveBtn.click();
    Thread.sleep(2000);
    WebElement newName=
webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[1]/div[2]/div/h1"));
    assertTrue(newName.getText().contains("Snack"));
    assertFalse(newName.getText().contains("lalal"));
}
```

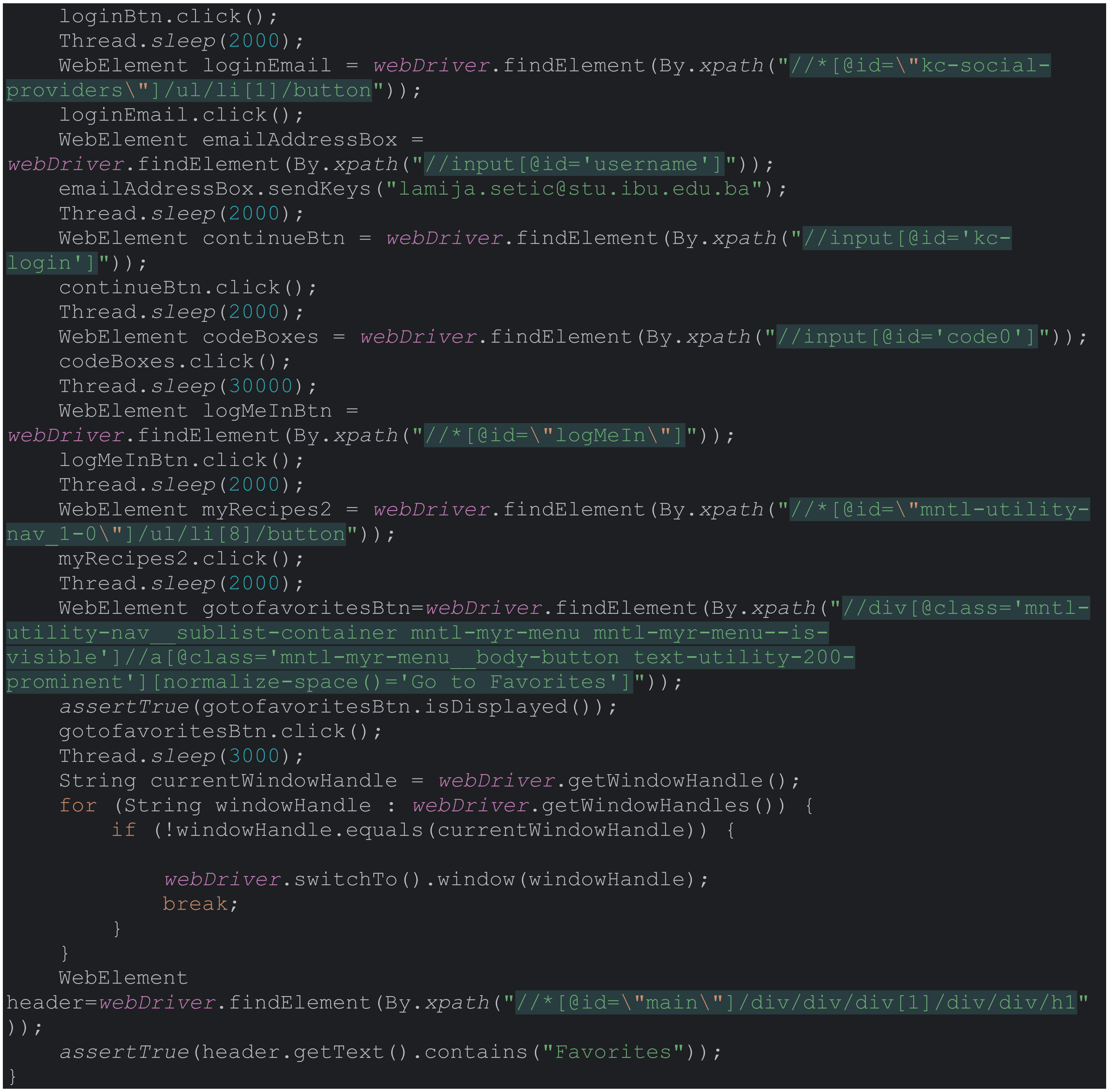
3.15. Test Scenario Name: myRecipes Homepage test

This test scenario is about testing homepage of this page (cards, carousels, navbar links).

Test Name: Test favorites link

Description: Check if the user is able to successfully log in and navigate to the "Favorites" page.				
Pre-condition(s): User is registered.				
Test Steps	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Navigate to the "EatingWell" website. 2. Click the "MyRecipes" login button. 3. Select "Login with Email". 4. Enter the email address and click "Continue". 5. Enter the verification code and click "Log me in". 6. Click the "My Recipes" dropdown. 7. Verify the "Go to Favorites" button is visible. 8. Click the "Go to Favorites" button. 9. Switch to the new window. 10. Verify the page header contains "Favorites".	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The "Go to Favorites" button should be visible, and the user should successfully navigate to the "Favorites" page with the header containing the word "Favorites".	The "Go to Favorites" button was visible, and the user successfully navigated to the "Favorites" page where the header contained the word "Favorites".	PASS
Notes:				

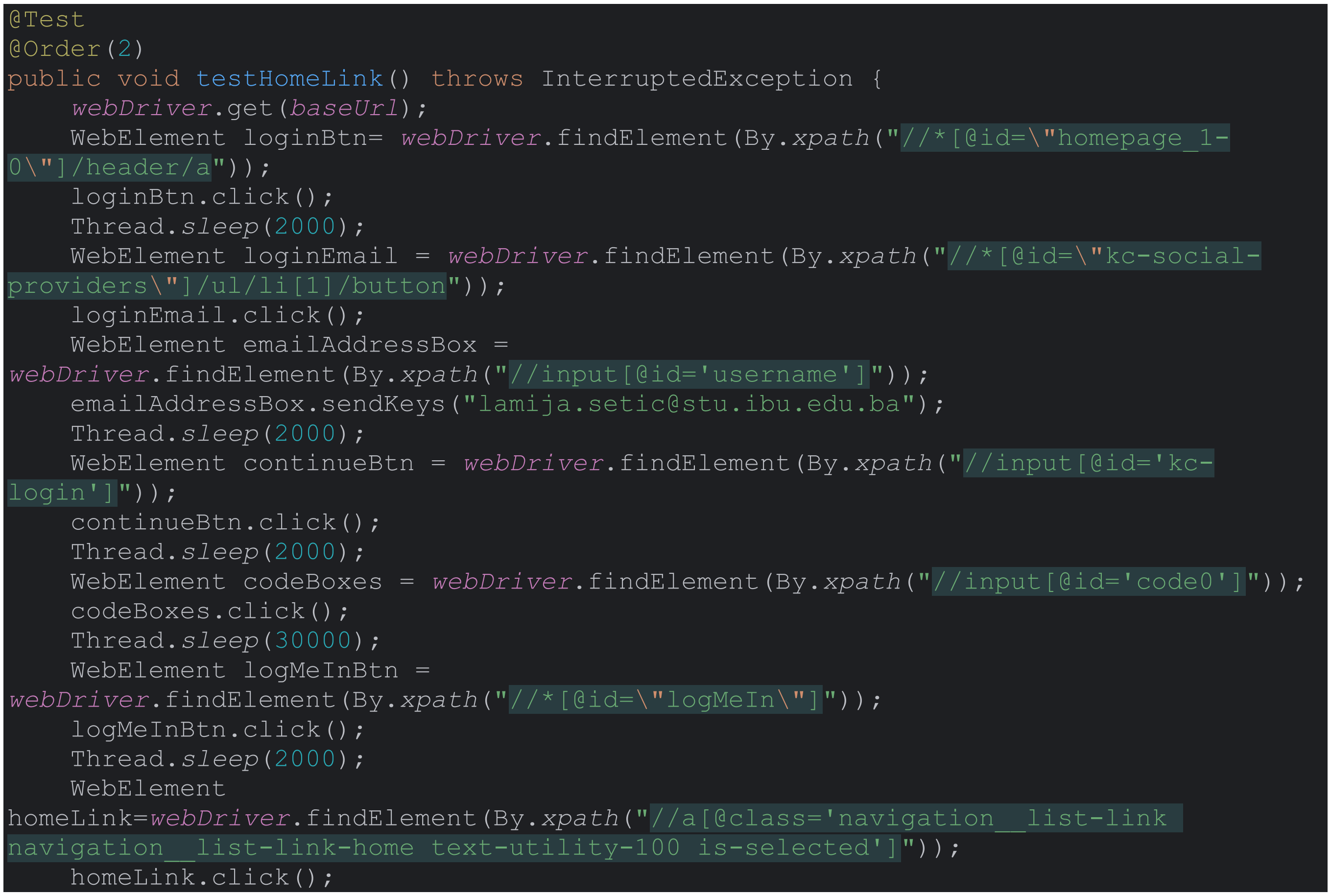
```
@Test
@Order(1)
public void testFavoritesLink() throws InterruptedException {
    webDriver.get("https://www.eatingwell.com");
    WebElement myRecipes = webDriver.findElement(By.xpath("//div[@id='mntl-utility-
nav_1-0']//a[@aria-label='Trigger interstitial modal to log in to 
MyRecipes']//*[name()='svg']"));
    myRecipes.click();
    Thread.sleep(2000);
    WebElement loginBtn = webDriver.findElement(By.xpath("//*[@id=\"mm-myrecipes-
interstitial__content_1-0\"]/div[1]/div[2]/a"));
```



```
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-
login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement myRecipes2 = webDriver.findElement(By.xpath("//*[@id=\"mntl-utility-
nav_1-0\"]/ul/li[8]/button"));
    myRecipes2.click();
    Thread.sleep(2000);
    WebElement gotofavoritesBtn=webDriver.findElement(By.xpath("//div[@class='mntl-
utility-nav__sublist-container mntl-myr-menu mntl-myr-menu--is-
visible']//a[@class='mntl-myr-menu__body-button text-utility-200-
prominent'][normalize-space()='Go to Favorites']"));
    assertTrue(gotofavoritesBtn.isDisplayed());
    gotofavoritesBtn.click();
    Thread.sleep(3000);
    String currentWindowHandle = webDriver.getWindowHandle();
    for (String windowHandle : webDriver.getWindowHandles()) {
        if (!windowHandle.equals(currentWindowHandle)) {

            webDriver.switchTo().window(windowHandle);
            break;
        }
    }
    WebElement
header=webDriver.findElement(By.xpath("//*[@id=\"main\"]/div/div/div[1]/div/div/h1"
));
    assertTrue(header.getText().contains("Favorites"));
}
```

Test Name: Test homepage link				
Description: Check if the user can successfully log in and navigate to the "Home" page where the header displays "A Home for Your Favorite Recipes."				
Pre-condition(s): User is registered.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**

1. Navigate to the base URL. 2. Click the "Login" button. 3. Select "Login with Email" and enter the email address. 4. Click "Continue". 5. Enter the verification code and click "Log me in". 6. Click the "Home" link in the navigation menu. 7. Verify the header contains "A Home for Your Favorite Recipes".	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The header should contain the text "A Home for Your Favorite Recipes."	The header contains the text "A Home for Your Favorite Recipes."	PASS
Notes:				



```
@Test
@Order(2)
public void testHomeLink() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn= webDriver.findElement(By.xpath("//*[@id=\"homepage_1-
0\"]/header/a"));
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-
login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement
homeLink=webDriver.findElement(By.xpath("//a[@class='navigation__list-link
navigation__list-link-home text-utility-100 is-selected']"));
    homeLink.click();
    Thread.sleep(2000);
    WebElement header=webDriver.findElement(By.xpath("//h2[@id='home-
loggedin__title_1-0']"));
    assertTrue(header.getText().contains("A Home for Your Favorite Recipes"));

}
```

Test Name: Test account link				
Description: Check if a logged-in user can successfully navigate to their account page and see the correct email address displayed.				
Pre-condition(s): User is registered.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Open the browser and navigate to the base URL. 2. Click the "Login" button in the header. 3. Click the email login option. 4. Enter a valid email address ("lamija.setic@stu.ibu.edu.ba") in the email input box. 5. Click the "Continue" button to proceed. 6. Click the code box to enter the authentication code. 7. Wait for the code input to become active (simulate a 30-second delay). 8. Click the "Log me in" button. 9. Click the "Account" link in the navigation menu.	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The correct email address ("lamija.setic@stu.ibu.edu.ba") should be displayed on the account page.	The email address ("lamija.setic@stu.ibu.edu.ba") is correctly displayed on the account page.	PASS

10. Wait for the account page to load. 11. Verify that the correct email address is displayed on the account page.				

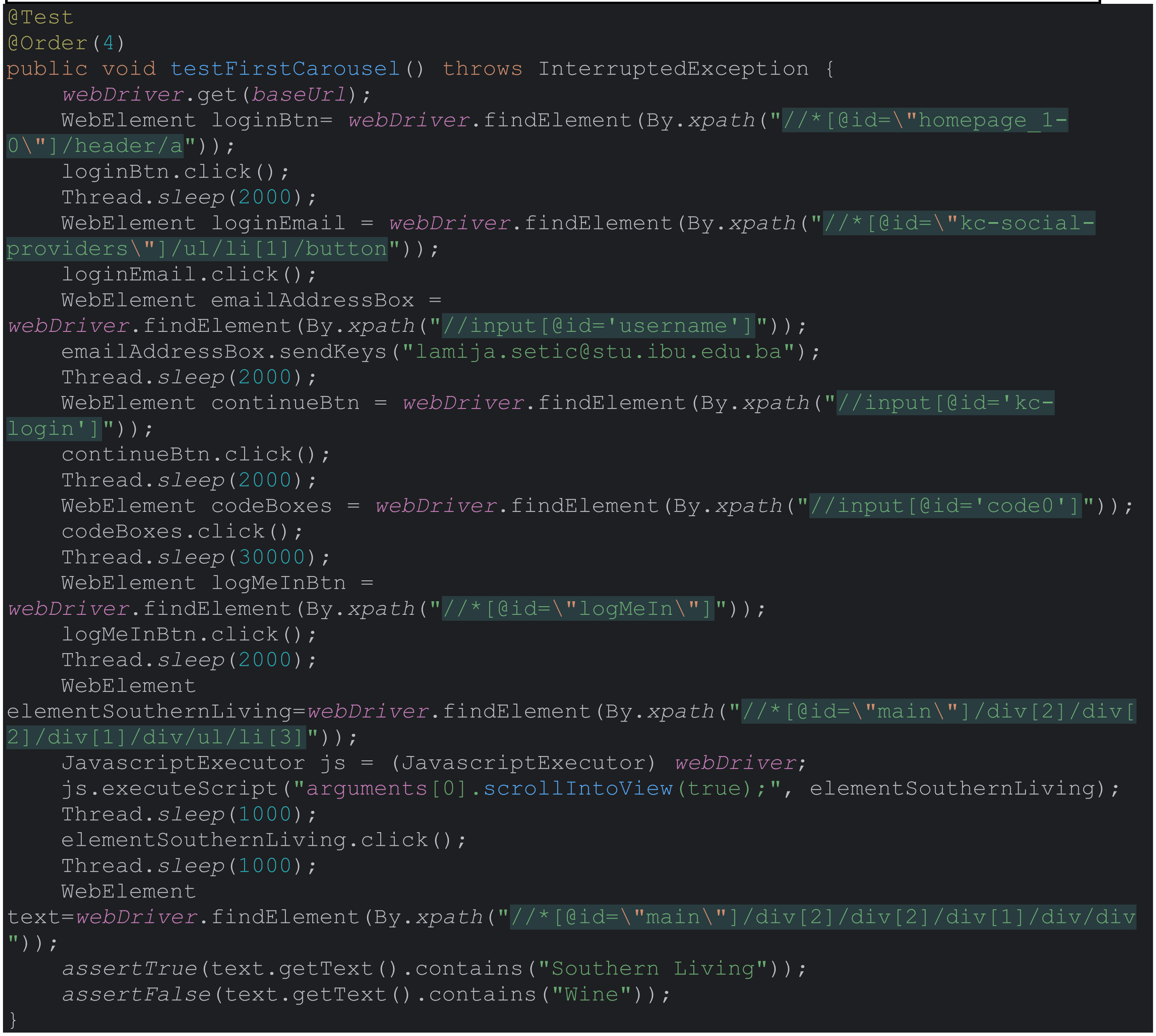
Notes:

```
@Test
@Order(3)
public void testAccountLink() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn= webDriver.findElement(By.xpath("//*[@id=\"homepage_1-
0\"]/header/a"));
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-
login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement
accountLink=webDriver.findElement(By.xpath("//a[@class='navigation__list-link 
navigation__list-link-account text-utility-100 ']"));
    accountLink.click();
    Thread.sleep(3000);
    WebElement email=webDriver.findElement(By.xpath("//div[@class='account__email-
display']"));
    assertTrue(email.getText().contains("lamija.setic@stu.ibu.edu.ba"));

}
```

Test Name: Test first carousel
Description: Check if the user can successfully click on the first item in the carousel and verify that the corresponding content is displayed correctly.

Pre-condition(s): User is registered.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Open the browser and navigate to the base URL. 2. Click the "Login" button in the header. 3. Click the email login option. 4. Enter a valid email address ("lamija.setic@stu.ibu.edu.ba") in the email input box. 5. Click the "Continue" button to proceed. 6. Click the code box to enter the authentication code. 7. Wait for the code input to become active (simulate a 30-second delay). 8. Click the "Log me in" button. 9. Wait for the page to load. 10. Locate the first item in the carousel, "Southern Living." 11. Scroll to the carousel element to bring it into view. 12. Click the "Southern Living" item in the carousel. 13. Wait for the	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	The content page should display text containing "Southern Living" and not contain "Wine."	The content displayed contains "Southern Living" and does not contain "Wine."	PASS

content page to load. 14. Verify that the content displayed contains "Southern Living" and does not contain "Wine."				
Notes:				



```
@Test
@Order(4)
public void testFirstCarousel() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn= webDriver.findElement(By.xpath("//*[@id=\"homepage_1-
0\"]/header/a"));
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-
providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox =
webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-
login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn =
webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement
elementSouthernLiving=webDriver.findElement(By.xpath("//*[@id=\"main\"]/div[2]/div[
2]/div[1]/div/ul/li[3]"));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", elementSouthernLiving);
    Thread.sleep(1000);
    elementSouthernLiving.click();
    Thread.sleep(1000);
    WebElement
text=webDriver.findElement(By.xpath("//*[@id=\"main\"]/div[2]/div[2]/div[1]/div/div
"));
    assertTrue(text.getText().contains("Southern Living"));
    assertFalse(text.getText().contains("Wine"));
}
```

Test Name: Test card

Description: Check if clicking on a card does not navigate to the incorrect URL.				
Pre-condition(s): User is registered.				
Test Steps:	**Test Data:**	**Expected Result:**	**Actual Result:**	**Status:**
1. Open the browser and navigate to the base URL. 2. Click the "Login" button in the header. 3. Click the email login option. 4. Enter a valid email address ("lamija.setic@stu.ibu.edu.ba") in the email input box. 5. Click the "Continue" button to proceed. 6. Click the code box to enter the authentication code. 7. Wait for the code input to become active. 8. Click the "Log me in" button. 9. Wait for the page to load. 10. Locate the card element on the page. 11. Scroll to bring the card into view. 12. Click the card to view its content. 13. Verify that the current URL is not equal to "https://www.bhg.com/chicken-	Email address: lamija.setic@stu.ibu.edu.ba Verification Code: A valid code provided during the test	Clicking on the card should not navigate to the URL.	The current URL is not equal to URL.	PASS

fricassee-8738389."				
Notes:				

```
@Test
@Order(5)
public void testCard() throws InterruptedException {
    webDriver.get(baseUrl);
    WebElement loginBtn= webDriver.findElement(By.xpath("//*[@id=\"homepage_1-0\"]/header/a"));
    loginBtn.click();
    Thread.sleep(2000);
    WebElement loginEmail = webDriver.findElement(By.xpath("//*[@id=\"kc-social-providers\"]/ul/li[1]/button"));
    loginEmail.click();
    WebElement emailAddressBox = webDriver.findElement(By.xpath("//input[@id='username']"));
    emailAddressBox.sendKeys("lamija.setic@stu.ibu.edu.ba");
    Thread.sleep(2000);
    WebElement continueBtn = webDriver.findElement(By.xpath("//input[@id='kc-login']"));
    continueBtn.click();
    Thread.sleep(2000);
    WebElement codeBoxes = webDriver.findElement(By.xpath("//input[@id='code0']"));
    codeBoxes.click();
    Thread.sleep(30000);
    WebElement logMeInBtn = webDriver.findElement(By.xpath("//*[@id=\"logMeIn\"]"));
    logMeInBtn.click();
    Thread.sleep(2000);
    WebElement card = webDriver.findElement(By.xpath("//*[@id=\"mntl-card-list-items_1-0\"]"));
    JavascriptExecutor js = (JavascriptExecutor) webDriver;
    js.executeScript("arguments[0].scrollIntoView(true);", card);
    Thread.sleep(2000);
    card.click();
    assertNotEquals("https://www.bhg.com/chicken-fricassee-8738389", webDriver.getCurrentUrl());

}
```

4. Conclusion

4.1. Testing Summary

Testing Tool	Total Tests	Passed Tests	Failed Tests
Selenium WebDriver in IntelliJ IDEA	79	79	0

4.2. Final Thoughts

As for our closing statements and final thoughts about this project that we tested, we must say that, in terms of implementation, the website is very well-developed. There were no major errors, bugs, or issues during the testing process. Everything ran smoothly, efficiently, and as intended. There were no delays or rendering issues with the elements, which is commendable. From a functionality perspective, the website operates flawlessly.

However, regarding the website's structure, we find it overly complicated. We believe this could have been achieved in a much simpler way, without merging multiple pages into one website. While this may have been a compromise by the stakeholders, it creates some confusion for users. Specifically, every recipe or linked element leads to an entirely different page, opening new windows or tabs. This can be confusing and even overwhelming for users navigating the website. Personally, this was the biggest downside of the project.

Another aspect we'd like to highlight is the login and registration process, which is unique but also problematic. Every time a user logs in or registers, the system requires a verification code sent to their email. While this enhances security, it becomes tedious for the user. If someone already has an account, they shouldn't need to verify themselves with a code every time they log in. This was particularly inconvenient during testing, as we had to wait for the code, manually input it, and continue testing within the WebDriver session. This process added unnecessary complexity to the testing workflow.

Additionally, testing was further complicated due to the structure involving multiple pages within one website. For instance, we couldn't initialize the user login in the beforeAll setup since the login process occurs on a completely different page (so we run each test manually). This made it impossible to standardize the setup across all tests, as not all tests began on the same page. This structural inconsistency caused significant challenges during testing, particularly for login and registration functionalities.

To conclude, while the website is well-implemented and functions effectively, its structure and navigation create unnecessary complexity for users and testers alike. Features like constant verification during login, multiple windows, and the fragmented page structure detract from the user experience. Other than these points, everything else is well-executed, and the critical functionalities perform as expected.

However, this was a very interesting experience for us in terms of testing this website. We learned many new things, and it was definitely a much broader testing experience compared to what we did in the labs, mainly because there were more functionalities to explore. We also had to research how to test certain aspects of the site.

Of course, we encountered some issues with locating certain elements on the page, but we managed to resolve them all. Overall, we made a significant effort to test the majority of the functionalities related to the website. We believe we did a good job and tested everything that was realistically possible to test.