Testing Team

Test Case Template

Faculty of mathematics and informatics

Course: Software quality assurance

Students: Mihaela Ilieva, Pavla Manova

Table of contents

1 Test SignUp Section

1.1.1.1 Test type

___ Functional Test

Tester:	Date:

1.1.1.2 System Under Test

System name: InaEssentials web-site, SignUp Section

Version: 8 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

1.1.1.3 Test Personnel

Name: Mihaela Ilieva   Date: 8.01.2023   Time/h: 3.5

1.1.2 Test Summary

1.1.2.1 Results

Total number of test cases:	6
Total number of test cases passed:	6
Total number of test cases inconclusive:	0
Total number of test cases failed:	0
Total number of bugs found:	1

1.1.3 Test Cases

1.1.3.1 SignUpTestValidInformation

Special Instructions

The system under testing is build to stop the user from liking a comment that has been assessed,

Description	Tests the SignUp form with unused email and navigating to the form from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. Also, the user should not have registered to https://inaessentials.com. Because of being registered, this test will not pass, but if using new information, it will.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-12	Insert Valid data for signing up.	To write the inserted data in the text boxes given.
13	Click the “Създаване на профил” button.	Seeing the main page, hence not seeing the error that that data is already used for another account.
14	Pause so the user can fill out the captcha forms.	To pause the next command for a bit.
15	Assert that there is no error.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The user should insert valid data that has not been used to create a profile before. If the user inserts used before data, they should see an error message for already existing account.

The test uses the following additional commands:

pause – waits for the given amount of milliseconds before checking the next command. It is added because captcha pictures appear before seeing the final result and we need time to fill them out.

assert element not present – checks if the element given as target is really not there

1.1.3.2 SignUpTestInvalidEmail

Special Instructions

NONE

Test Case ID	TC_SignUpTestInvalidEmail	
Description	Tests the SignUp form with invalid email and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. Also, the user should insert non-used before information.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x1042.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-8	Insert valid data about the names.	To write the inserted data in the text boxes given.
9-10	Insert Invalid email data for signing up.	To write the inserted data in the text boxes given.
11-12	Insert valid data about the password.	To write the inserted data as dots in the text box given.
13	Clicking the “Създаване на профил” button.	Show an error message for invalid result (does not happen). It shows the main page instead.
Test verdict	Passed, but should have not. The website allows using fake emails for creating accounts.	

Comments:

The user should insert invalid email data that has not been used to create a profile before. The test shows that a person can Sign Up with a non-existing fake email. It should not happen. We can add one assert element present about an error list, but such does not show

1.1.3.3 SignUpTestWithAlreadyUsedEmail

Special Instructions

Try Signing Up with all valid data, but an email that you have used before.

Test Case ID	TC_SignUpTestWithAlreadyUsedEmail	
Description	Tests the SignUp form with used email and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. The user should try registrating with an email that is already used by a different account.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-12	Inserting Valid data, except for the already used email for signing up.	To write the inserted data in the text boxes given.
13	Clicking the “Създаване на профил” button.	Seeing an error message that says there is already an account created with the given email.
14	Pause for 7 seconds so the user can fill the captcha form.	To pause the next command for seven seconds.
15	Assert that there is an error message.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The user should insert valid data that has not been used to create a profile before. If the user inserts used before data, they should see an error message for already existing account.

The test uses the following additional commands:

pause – waits for the given amount of milliseconds before checking the next command. It is added because captcha pictures appear before seeing the final result and we need time to fill them out.

assert element present – checks if the element given as target is really there.

1.1.3.4 ShortNamesSignUp

Special Instructions

NONE

Test Case ID	TC_ShortNamesSignUp	
Description	Tests the SignUp form with way too short names and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. The user should try registrating with short names.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-8	Insert one letter per each name.	Error message -there is none.
9-12	Insert valid email and password.	To write the inserted data in the text boxes given.
13	Click the “Създаване на профил” button.	Seeing an error message that says the names are too short.
Test verdict	Passed – but it should have more than one letter minimum for names.	

Comments:

After clicking the “Създаване на профил” button we do not see any error, our account is created. The website developers can put a higher limit for names than only one letter for each.

1.1.3.5 SignUpWithoutName

Special Instructions

NONE

Test Case ID	TC_SignUpWithoutName
Description	Tests the SignUp form with inserting no names and navigating to it from the main page.
Applicable for	IE6, Google Chrome

Initial Conditions	The user should have access to Internet. The user should try registrating with no names and valid password.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1708x920.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-8	Insert valid data for the email and password.	To write the inserted data in the text boxes given.
9	Click the “Създаване на профил” button.	Seeing an error message that says the first name is too short.
10	Assert the error.	To be true.
11-12	Insert valid first name.	To write the inserted data in the text box given.
13	Click the “Създаване на профил” button.	Seeing an error message that says the second name is too short.
14	Assert the error.	To be true.
15-16	Insert valid second name.	To write the inserted data in the text box given.
17	Click the “Създаване на профил” button after filling in the first name.	Creating an account.
Test verdict	Passed – as it should have.	

Comments:

An idea for making the user experience better is to show all the errors at once, not one by one until every one is done. This way the user will not have to click the “Създаване на профил” button unnecessary.

We can also add assert element present for an error after creating the account but there will not be any error.

The test uses the following additional commands:

assert element present – checks if the element given as target is really there.

1.1.3.6 SignUpWithShortPassword

Special Instructions

NONE

Test Case ID	TC_SignUpWithShortPasword	
Description	Tests the SignUp form with inserting short password and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. The user should try registrating with short password.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form with an option to SignUp under it.
4	Click the SignUp option.	The SignUp form should be showing.
5-10	Insert valid data, except for the short password for signing up.	To write the inserted data in the text boxes given.
11-12	Insert short password.	To write the inserted data as dots in the text box given.
13	Click the “Създаване на профил” button.	Seeing an error message that says the password is too short.
14	Pause for a bit to fill the captcha form.	To pause for bit.
15	Assert there is an error message showing.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The error that shows is giving information to the user that the minimum length of the password is five symbols. It also cleans the SignUp form.

The test uses the following additional commands:

pause – waits for the given amount of milliseconds before checking the next command. It is added because captcha pictures appear before seeing the final result and we need time to fill them out.
assert element present – checks if the element given as target is really there

2 Test Login Section

2.1.1.1 Test type

___ Functional Test

2.1.1.2 System Under Test

System name: InaEssentials web-site, Login Section

Version: 8 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

2.1.1.3 Test Personnel

Name: Mihaela Ilieva   Date: 8.01.2023   Time/h: 1.5

2.1.2 Test Summary

2.1.2.1 Results

Total number of test cases:	3
Total number of test cases passed:	3
Total number of test cases inconclusive:	0
Total number of test cases failed:	0
Total number of bugs found:	0

2.1.3 Test Cases

2.1.3.1 LoginValidData

Special Instructions

NONE

Test Case ID	TC_LoginValidData	
Description	Tests the Login form with inserting valid data and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x1042.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form.
4-5	Store the data that is going to be given in the form.	Storing the data correctly.
6-9	Insert the data in the form.	To write the inserted data in the text boxes given.
10-14	Check if the data is valid with an if-else construction.	The data to be valid.
15	Assert that the data is valid.	To be true.
16.	Click the “Влизане” button and asserting that there is no error message for wrong email or password.	Seeing no error message but the main page instead.
17.	Assert there will be no error message.	To be true.
Test verdict	Passed – as it should have.	

Comments:

We can add another pause before the assert element not present because of the captcha tests. Otherwise, there is no need.

The test uses the following additional commands:

assert element not present – checks if the element given as target is really not there.

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

store – stores information in the variable declared in the value field

assert – checks if the target's value is like the given in the value field.

2.1.3.2 LoginInvalidEmail

Special Instructions

NONE

Test Case ID	TC_LoginInvalidEmail	
Description	Tests the Login form with inserting invalid email and navigating to it from the main page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x1042.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form.
4,6	Insert the invalid email.	To write the inserted data in the text box given.
5	Store the data that is going to be given in the form.	Storing the data correctly.
7-8	Insert the valid password in the form.	To write the inserted data in the text box given.
9-13	Check if the email is valid with an if-else construction.	The data not to be valid.

14	Click the “Влизане” button and asserting that there is an error message for wrong email or password.	Seeing an error message that the email or password is incorrect.
15	Assert there is an error.	To be true.
16	Pause for a bit to fill the captcha form.	To pause for a bit.
17	Assert that there is an error message.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The test uses the following additional commands:

assert element present – checks if the element given as target is really there.

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

store – stores information in the variable declared in the value field

pause – waits for the given amount of milliseconds before checking the next command. It is added because captcha pictures appear before seeing the final result and we need time to fill them out

assert – checks if the target’s value is like the given in the value field

2.1.3.3 LoginInvalidPassword

Special Instructions

NONE

Test Case ID	TC_LoginInvalidPassword
Description	Tests the Login form with inserting invalid password and navigating to it from the main page.
Applicable for	IE6, Google Chrome

Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x1042.	The window size should be set as the wanted size.
3	Click the person icon on top of the menu.	See the LogIn form.
4-5	Insert the valid email.	To write the inserted data in the text box given.
7	Store the data that is going to be given in the form.	Storing the data correctly.
6-8	Insert the valid password in the form.	To write the inserted data in the text box given.
9-13	Check if the password is valid with an if-else construction.	The data not to be valid.
14	Click the "Влизане" button and asserting that there is an error message for wrong email or password.	Seeing an error message that the email or password is incorrect.
15	Assert there is an error.	To be true.
16	Pause for a bit to fill the captcha form.	To pause for a bit.
17	Assert that there is an error message.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The test uses the following additional commands:

assert element present – checks if the element given as target is really there.

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

store – stores information in the variable declared in the value field

pause – waits for the given amount of milliseconds before checking the next command. It is added because captcha pictures appear before seeing the final result and we need time to fill them out

assert – checks if the target’s value is like the given in the value field

3 Test Basket Section

3.1.1.1 Test type

___ Functional Test

3.1.1.2 System Under Test

System Under Test

System name: InaEssentials web-site, Basket Section

Version: 9 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

3.1.1.3 Test Personnel

Name: Mihaela Ilieva   Date: 9.01.2023   Time/h: 10

3.1.2 Test Summary

3.1.2.1 Results

Total number of test cases:	13
Total number of test cases passed:	11
Total number of test cases inconclusive:	2
Total number of test cases failed:	0
Total number of bugs found:	5

3.1.3 Test Cases

3.1.3.1 OpenBasket

Special Instructions

NONE

Test Case ID	TC_OpenBasket	
Description	Tests the opening of the basket.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. Also, they should have added at least one item to their basket.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the basket icon on top of the menu.	See the basket preview.
4	Click the “Преминаване към плащането” button.	Redirecting us to the /cart page.
Test verdict	Passed – as it should have.	

Comments:

The person should have at least one item added to the basket. Otherwise, the preview of the basket will be different.

3.1.3.2 RedirectingFromEmptyBasket

Special Instructions

NONE

Test Case ID	TC_RedirectingFromEmptyBasket
Description	Tests the redirecting from the empty basket preview menu.
Applicable for	IE6, Google Chrome

Initial Conditions	The user should have access to Internet. They must not have anything in the basket.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the login page.	The main page of inaessentials.com is shown.
2	Set window size at 976x922.	The window size should be set as the wanted size.
3	Click the basket icon on top of the menu.	See empty basket preview.
4	Store the link where we expect to be redirected.	Storing it.
5	Click the redirection button.	Taking us to the link above.
6	Store the link where we actually get redirected to.	Storing it.
7-11	Check if the two links are identical.	They should be.
12	Assert the two links are identical.	To be true.
Test verdict	Passed – as it should have.	

Comments:

The basket should be empty beforehand or there will be no redirecting links.

The test uses the following additional commands:

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

store – stores information in the variable declared in the value field

assert – checks if the target's value is like the given in the value field

3.1.3.3 IncreaseNumberOfProductsInBasket

Special Instructions

NONE

Test Case ID	TC_IncreaseNumberOfProductsInBasket	
Description	Tests the adding of an element to the basket from its page and then increasing its quantity from the basket preview, using the “+” button.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the page of a given product.	The page of the product should be open.
2	Set window size at 976x922.	The window size should be set as the wanted size.
3	Click the “Добави в количката” button.	See the product being added to the basket through the basket preview.
4	Add counter for the five increasings of the product.	To create a counter with a start value=1.
5-8	Click the “+” button for five times.	Seeing the quantity growing bigger.
Test verdict	Passed – as it should have.	

Comments:

There should be 5 more of the item selected in the basket, unless it is a matter of limits. This is described in another test.

The test uses the following additional commands:

do-repeat if construction – loop that executes whatever it is told to execute, as long as a given condition is true

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

3.1.3.4 RemoveNumberOfProductsFromBasket

Special Instructions

NONE

Test Case ID	TC_RemoveNumberOfProductsFromBasket

Description	Tests the removal of an element using the “-” button in the basket preview.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the page of the cart.	The page /cart should be open.
2	Setting window size at 976x920	The window size should be set as the wanted size.
3	Create a counter for the five decreasings of the element.	To create a counter with start value=1.
4-7	Clicking the “-” button five times for the selected item.	See the product count being decreased.
Test verdict	Passed – as it should have.	

Comments:

If there are 5 or more of given item, it will decrease them without a problem. However, if the count is less than 5, the test will fail on the loop, which is the expected behavior.

The test uses the following additional commands:

do-repeat if construction – loop that executes whatever it is told to execute, as long as a given condition is true

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

3.1.3.5 RemoveOneFromManyElementsFromBasket

Special Instructions

The user should have exactly four different elements in their basket

Test Case ID	TC_RemoveOneFromManyElementsFromBasket
Description	Tests the removing of an element through the “Премахване” button.
Applicable for	IE6, Google Chrome

Initial Conditions	The user should have access to Internet. The user should have exactly four different elements in their basket, otherwise we have to change the code of the test.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart.	The page /cart should be open.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the “Премахване” button.	See the product being discarded from the basket.
4	Check if the number of elements has decreased- the last element of the collection should not be present.	The last element of the collection not be present.
Test verdict	Passed – as it should have.	

Comments:

The user should have exactly four different elements in their basket, otherwise we have to change the code of the test – it will be a small change on the target of the assert element not present command.

The test uses the following additional commands:

assert element not present – checks if the element given as target is really not there.

3.1.3.6 RemoveOnlyElementFromBasket

Special Instructions

The user should have exactly one different element in their basket

Test Case ID	TC_RemoveOnlyElementFromBasket	
Description	Tests the removing of the only element in the basket through the “Премахване” button.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. The user should have exactly one different element in their basket.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart.	The page /cart should be open.

2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the "Премахване" button.	See the product being discarded from the basket.
4	Check if the collection of elements is not there.	The collection of elements should be missing.
Test verdict	Passed – as it should have.	

Comments:

The user should have exactly one different element in their basket

The test uses the following additional commands:

assert element not present – checks if the element given as target is really not there.

3.1.3.7 BasketPreviewForEmptyBasket

Special Instructions

The user should have an empty basket

Test Case ID	TC_BasketPreviewForEmptyBasket	
Description	Inspects the basket preview for an empty basket.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet. The user should have empty basket.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the website.	The main page should be open.
2	Set window size at 976x922.	The window size should be set as the wanted size.
3	Click the basket icon at the top of the menu.	See the basket preview open.
4-6	Click on the different elements of the basket preview	Nothing happening, just inspecting the elements of the basket preview.
7	Click on the "X" button	Close the basket preview.
Test verdict	Passed – as it should have.	

Comments:

The user should have an empty basket.

3.1.3.8 BasketFakeDiscountCode

Special Instructions

NONE

Test Case ID	TC_BasketFakeDiscountCode	
Description	Tests the behavior of the website when adding a fake discount code, to see whether the code will be accepted as real or not.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart.	The page /cart should be open.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the discount code field.	Get rights to write there.
4	Type the fake value.	None.
5	Click the “Приложи” button.	Error for invalid code.
6	Pause for a bit to wait for the next page to load.	To pause for a bit.
7	Assert if the expectations are correct.	They should be correct.
Test verdict	Passed – as it should have.	

Comments:

The user will see an error message for invalid code after inserting a fake one.

The test uses the following additional commands:

assert element present – checks if the element given as target is really there

pause – waits for the given amount of milliseconds before checking the next command. It is added because the error message takes some time to appear

3.1.3.9 BasketRealDiscountCode

Special Instructions

NONE

Test Case ID	TC_BasketRealDiscountCode	
Description	Tests the behavior of the website when adding a real discount code, to see whether the code will be accepted as real or not.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart.	The page /cart should be open.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the discount code field.	Get rights to write there.
4	Type the real value.	None.
5	Click the “Приложи” button.	Label for applied code and no error.
6	Pause for a bit to wait for the next page to load.	To pause for a bit.
7	Assert if the expectations are correct.	They should be correct.
Test verdict	Passed – as it should have.	

Comments:

The user will see a label with their code, indicating that it is applied to the basket and there will not be an error message as in the previous test.

The test uses the following additional commands:

assert element present – checks if the element given as target is really there

assert element not present – checks if the element given as target is really not there

pause – waits for the given amount of milliseconds before checking the next command. It is added because the error message takes some time to appear, so we are sure that it really does not appear

3.1.3.10 BasketPaymentFormRedirecting

Special Instructions

NONE

Test Case ID	TC_BasketPaymentFormRedirecting	
Description	Tests the behavior of the website when clicking the “Плащане” button. Checking what values the user can give to the different types of information fields and if they are all admissible.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet and something in the basket that they want to buy.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart.	The page /cart should be open.
2	Set window size at 1722x936.	The window size should be set as the wanted size.
3	Click the “Плащане” button.	Get redirected to the payment page.
4-5	Choose values for address id and country.	The values to be shown as chosen.
6-7	Type in mandatory address with one letter only.	Error message.
8-9	Type in non-mandatory address with one letter only.	Nothing to happen.
10-11	Type in ZIP code with one letter only.	Error message.
12-13	Type in city with one letter only.	Error message.
14-15	Type in phone number with one letter only.	Error message.
16	Click the “Продължи” button.	A lot of error messages. However, there is an error only for the zip code.
17	Assert there is an error with the ZIP code.	To be true.
18-19	Correct the ZIP code error.	Nothing to happen.
20	Click the “Продължи” button.	A lot of error messages. However, there are none. The user can continue with the payment with the invalid information given by them.
Test verdict	Inconclusive – there should have been way more errors based on the invalid data information, but there is only one error. This is certainly the system’s fault, as it doesn’t have enough validations. The test is written to pass, but those misses from the developers should be fixed.	

Tester: Date:

Comments:

The test uses the following additional commands:

assert element present – checks if the element given as target is really there

3.1.3.11 BasketPaymentContinue

Special Instructions

NONE

Test Case ID	TC_BasketPaymentContinue	
Description	Tests the behavior of the website when clicking the “Продължи” button. Checking what values the user can give to the different types of information fields and if they are all admissible.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet and something in the basket that they want to buy and to have already filled in the name-address form from the previous test.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the order form.	The page /25186068/checkouts/3b3bce3b4972ac718ba5c71e1f2a37fb?previous_step=contact_information&step=shipping_method should be open.
2	Setting window size at 976x920.	The window size should be set as the wanted size.
3-6	Choose a delivery option.	Mark the wanted option from the menu.
7	Click the “Продължете към плащане” button.	Redirect to the next page,
8-10	Fill out the number of the card with only one number.	Error message.
11-14	Fill out the name on the card with one letter.	Error message, but there is no such message.
15-18	Fill out the expiry date with wrong value.	Error message, but there is no such message.
19-22	Fill out the security code with one number only.	Error message.
23-25	Choose between paying in cash or with card -> choose card	Nothing to happen.

Tester: Date:

26	Click the “Платете сега” button.	The error messages to show. Only some of them are displayed.
27	Pause for a bit so the next page can load properly.	To pause for a bit.
28-30	Assert that there are indeed three error messages.	The assumption to be true.
Test verdict	Inconclusive – there should have been way more errors based on the invalid data information, but there are only three errors. This is certainly the system’s fault, as it doesn’t have enough validations. The test is written to pass, but those misses from the developers should be fixed.	

Comments:

The fact that some validations are not thought about by the developers’ team is described as bugs in the Traceability matrix.

The test uses the following additional commands:

assert element present – checks if the element given as target is really there

select frame – to perform an action in a given frame that is a part of an iframe; in our case this is the form for the card information

pause – waits for the given amount of milliseconds before checking the next command. It is added because the error message takes some time to appear, so we are sure that it really does appear.

3.1.3.12 BasketSetQuantityToLessThanPermissed

Special Instructions

The user should have at least two elements in their basket.

Test Case ID	TC_BasketSetQuantityToLessThanPermissed	
Description	Tests the behavior of the website when adding more than it is permited from a given product. Also seeing what the maximum quantity for an element is.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet and at least two things in the basket for comparison.	
Test Step ID	**Test Step Description**	**Expected Result**

1	Open the cart form.	The page /cart should be open.
2	Set window size at 1722x936.	Setting it the wanted size.
3-4	Put the number of items of type first product to -15.	Visualizing -15 in the field.
5	Store the value in the text box field.	To be -15.
6-10	Check if the field information is really -15.	It is not.
11	Assert that the information in the field cannot be -15.	To be true.
12-13	Put the number of items of type second product to 0.	Visualizing 0 in the field.
14	Check if the element is still there	It is not.
14	Assert that the assumptions above are correct.	The assumptions to be true.
Test verdict	Passed – This test gives us information about what the minimum quantity of a given element is (1);	

Comments:

The test shows us how to alter the quantity field of an element and what the minimum quantity for an element is. The system does not allow to enter negative numbers (it alters them to 1) and removes the element from the basket if the value is 0.

The test uses the following additional commands:

store value – used to retrieve and store the value attribute value of the located UI element into a variable in Selenium IDE.

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

assert – checks if the target’s value is like the given in the value field.

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

assert – checks if the target’s value is like the given in the value field

3.1.3.13 BasketAddMoreThanPermissed

Special Instructions

The user should have at least two elements in their basket.

Test Case ID	TC_BasketAddMoreThanPermissed	
Description	Tests the behavior of the website when setting the quantity to a zero or negative number. Also seeing what the minimum quantity for an element is.	
Applicable for	IE6, Google Chrome	
Initial Conditions	The user should have access to Internet and at least two things in the basket for comparison.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open the cart form.	The page /cart should be open.
2	Set window size at 1722x936.	Setting it the wanted size.
3-4	Put the number of items of type first product to 100000.	Visualizing 100000 in the field.
5	Store the value of the field of information.	To be 100000.
6-10	Check if the field information is really 100000.	It is.
11	Assert that the user can enter 100000 items of the same type of element.	To be true.
12-15	Put the number of items of type second product to 100001.	Visualizing 100001 in the field.
16	Store the value of the field of information.	To be 100001.
17-21	Check if the field information is really 100001.	It is not.
22	Assert that the assumptions above are correct.	The assumptions to be true.
Test verdict	Passed – This test gives us information about what the maximum quantity of a given element is (100000);	

Comments:

The test shows us how to alter the quantity field of an element and what the maximum quantity for an element is. The system does not allow to enter more than the maximum amount.

The test uses the following additional commands:

store value – used to retrieve and store the value attribute value of the located UI element into a variable in Selenium IDE.

if-else construction – for checking conditions, if the first condition is true, do something; else – do something else

end – puts an end to the if-else construction

execute script – does what is written in the target and value fields; in our case returns a string to an output variable

assert – checks if the target's value is like the given in the value field

1, 2, 3 Traceability matrix

Test Case ID	Bug Description	Note
TC_ SignUpTestInvalidEmail	When trying to SignUp with a fake email, it doesn't stop us, but instead creates an account for us.	It should instead show an error message and ask for a valid email. This can be easily fixed with a regex expression check for the email data.
TC_BasketPaymentFormRedirecting	The system allows us to give address with one letter only.	There should be a bigger minimum count of letters, as well as validation that the given address is correct and existing.
TC_BasketPaymentFormRedirecting	The system allows us to give city with one letter only.	There can be made a drop-down menu with all the cities in the country for the user to choose from, this way no further validation will be needed for the cities.

Test Case ID	Bug Description	Note
		It can be automatically filled based on the ZIP code as well.
TC_BasketPaymentFormRedirecting	The system allows us to give phone number with one letter only.	There should be a regex expression validation to check if the given number is correct, as well as a drop-down menu for the different starts of the phone numbers, based on country.
TC_BasketPaymentContinue	The system allows us to type a single letter for the name on the card.	There should be a validation with two fields for each name of the card and a minimum of symbols required, also there must not be allowed to enter numbers.
TC_BasketPaymentContinue	The system doesn’t allow us to write letters for the expiry date, but it allows writing invalid dates.	There should be a validation to check whether the given date is correct. Another option is to put a drop-down calendar there, so the users cannot choose a wrong date, even if they wanted to.

4 Test Single Product Page Feedback Section

4.1 Test Single Product Reviews Section

4.1.1.1 Test type

___ Functional Test

4.1.1.2 System Under Test

System name: InaEssentials web-site, Single Product Page Feedback Section

Version: 8 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

4.1.1.3 Test Personnel

Name: Pavla Manova   Date: 8.01.2023   Time/h: 6

4.1.2 Test Summary

4.1.2.1 Results

Total number of test cases:	8
Total number of test cases passed:	7
Total number of test cases inconclusive:	1
Total number of test cases failed:	0
Total number of bugs found:	0

4.1.3 Test Cases

4.1.3.1 Liking Comment

Special Instructions

The system under testing is build to stop the user from liking a comment that has been assessed, without noticing him. So the special conditions for this test are to test 'untouched' comment, although the test handles the condition 'liking liked/disliked comment'.

Test Case ID	TC_LikeComment	
Description	Test likes of comment incrementing by 1, when like button clicked.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	The product has comment section.
3	Click on Like button	Likes count is incremented by 1.
4	Verify that span element for likes count has text value increased by 1.	
Test verdict	Inconclusive. We expect the test to echo in the log console either 'Comment Liked' or 'Comment Already Assessed', but if there is trouble adding one like, the test will not detect the error, masked as the condition 'Comment Already Assessed'. Best solution – test on new comment, expect 'Comment Liked', test again, expect 'Comment Already Assessed'' – in this case test is passed.	

Comments:

The test uses the following additional commands:

Store initial likes count – **store text**

Store likes count after clicking – **store text**

Increments initial count by one and saves it in variable – **execute script, parseInt()**

If-else construction – checks new likes count == initial+1 – **echo** 'Comment Liked' checks new likes count == initial – **echo** ‘Comment Already Assessed’

Saves output in variable – **execute script**

Asserts the variable value is ‘correct’ - **assert**

4.1.3.2 Disliking Comment

Special Instructions

The system under testing is build to stop the user from disliking a comment with already given feedback(like or dislike), without noticing him. So the special conditions for this test are to test ‘untouched’ comment, although the test handles the conditions ‘disliking liked/disliked comment’.

Test Case ID	TC_DisikeComment	
Description	Test dislikes of comment incrementing by 1, when dislike button clicked.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/promo-set-kosopad is loaded.
2	Scroll to comments section	The product has comment section.
3	Click on Like button	Likes count is incremented by 1.
4	Verify that span element for likes count has text value increased by 1.	
Test verdict	Inconclusive. We expect the test to echo in the log console either ‘Comment Disliked’ or ‘Comment Already Assessed’, but if there is trouble adding one like, the test will not detect the error, masked as the condition ‘Comment Already Assessed’. Best solution – test on new comment, expect ‘Comment Liked’, test again, expect ‘Comment Already Assessed’ – in this case test is passed.	

Comments:

The test uses the following additional commands:

Store initial dislikes count – **store text**

Store dislikes count after clicking – **store text**

Increments initial count by one and saves it in variable – **execute script, parseInt()**

If-else construction – checks new dislikes count == initial+1 – **echo** ‘Comment Disliked’

checks new dislikes count == initial – **echo** ‘Comment Already Assessed’

Saves output in variable – **execute script**

Asserts the variable value is ‘correct’ – **assert**

4.1.3.3 Pagination on Comments Section

Special Instructions

The user must be on single product page.

Test Case ID	TC_Pagination	
Description	Test pagination, previous and next page, certain clicked page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section’s pagination	The product has comment section pagination.
3	Click on next page button	New comments load.
4	Verify that pagination element is still present.	
5	Click on previous page button	The previous results are displayed.
6	Verify that pagination element is still present.	
7	Click on button for page 3	New comments load in correct order according to page number.
8	Verify that pagination element is still present.	
9	Click on last page button	New comments load on the last page.
10	Verify that pagination page is still present.	
11	Click on last page button	Comments on first page are loaded.
12	Verify that pagination page is still present.	
Test verdict	Passed.	

Comments:

The test uses the following additional command – **assert element present** – for verifying pagination element is present.

4.1.3.4 Sort Comments By Latest Date

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SortDateDescending	
Description	Test sorting by latest date of comments section on a single product page.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	
3	Click on sort dropdown menu	Dropdown menu is displayed.
4	Select sorting by latest date – 'Най-скорошен'	Comments section reloads
5	Store first comment data-content attribute	Attribute value is stored
6	Store second comment data-content attribute	Attribute value is stored
7	Assert that the first date is bigger than the second date	Test passing with success
Test verdict	Passed.	

Comments:

The test uses the following additional commands:

Store attribute – to get first and second comment data in Coordinated Universal Time (UTC)

Execute script – to execute comparison of the two stored data values

Assert – to assert that the first date is newer than the second

4.1.3.5 Sort Comments By Rate Descending

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SortRateDescending	
Description	Test sorting comments on single product page by biggest rate.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access, The product must have at least two reviews with different rates.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	
3	Click on sort dropdown menu	Dropdown menu is displayed.
4	Select sorting by latest date – 'Най-висока оценка'	Comments section reloads
5	Store first comment data-score attribute	Attribute value is stored
6	Go to last page of comments	Comments section reloads
7	Store last comment data-score attribute	Attribute value is stored
8	Assert that the first score is bigger than the last score	Test passing with success
Test verdict	Passed.	

Comments:

The test uses the following additional commands:

Store attribute – to get first and last comment score

Execute script – to execute comparison of the two stored score values

parseInt – to parse score values from string to integer, for correct comparison

Assert – to assert that the scores are decreasing

4.1.3.6 Sort Comments By Rate Ascending

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SortRateAscending	
Description	Test sorting comments on single product page by smallest rate.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access, The product must have at least two reviews with different rates.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	
3	Click on sort dropdown menu	Dropdown menu is displayed.
4	Select sorting by latest date – 'Най-ниска оценка'	Comments section reloads
5	Store first comment data-score attribute	Attribute value is stored
6	Go to last page of comments	Comments section reloads
7	Store last comment data-score attribute	Attribute value is stored
8	Assert that the first score is smaller than the last score	Test passing with success
Test verdict	Passed.	

Comments:

The test uses the following additional commands:

Store attribute – to get first and last comment score

Execute script – to execute comparison of the two stored score values

parseInt – to parse score values from string to integer, for correct comparison

Assert – to assert that the scores are increasing

4.1.3.7 Sort Comments By Images Only

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SortPicsOnly	
Description	Test sorting comments on single product page by having pictures.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/promo-set-kosopad is loaded.
2	Scroll to comments section	
3	Click on sort dropdown menu	Dropdown menu is displayed.
4	Select sorting by latest date – 'Само снимки'	Comments section reloads
5	Check if there is pagination – if there are pages of products with pictures	
6	Store variable i=1 for iterating through while cycle	i is stored
7	Check if i is <= the number of pages	Continue with next steps or terminate the cycle
8	Count all reviews	The amount is stored
9	Count reviews with images	The amount is stored
10	Check if the amount of all reviews is equal to the amount of reviews with pictures. If true – continue, else – terminate test with error	The if statement is true
11	Check if there are still pages to be assessed	
12	If there are more pages – go to next page through pagination	Go to next page if there is one
13	Increase i with 1	
Test verdict	Passed.	

Comments:

The test uses the following additional commands in order:

Execute script – executes document.querySelector() function to check for pagination

If-Else – to check if there is pagination and store the number of pages if any

Store attribute – to save attribute for last page

While – to iterate through all pages with reviews

Execute script – executes document.querySelector() function to return number of all displayed reviews

Execute script – executes document.querySelector() function to return number of all reviews that have pictures

parseInt – to parse score values from string to integer, for correct comparison

Assert – to check if all reviews = reviews with images

If-Else – to go to next page if there is one

Execute script – to save and change variables

4.1.3.8 Sort Comments By Most Likes

Special Instructions
The user must be on single product page.

Test Case ID	TC_SortByMostLikedComment	
Description	Test sorting comments on single product page by usefullness.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access.	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	
3	Click on sort dropdown menu	Dropdown menu is displayed.
4	Select sorting by latest date – 'Най-полезни'	Comments section reloads

5	Store first comment likes span data-thumb-count attribute	Attribute value is stored
6	Go to last page of comments	Comments section reloads
7	Store last comment likes span data-thumb-count attribute	Attribute value is stored
8	Assert that the first comment likes are more than last comment likes.	Test passing with success
Test verdict	Passed.	

Comments:

The test uses the following additional commands:

Store attribute – to get first and last comment likes count

Execute script – to execute comparison of the two stored values

parseInt – to parse likes values from string to integer, for correct comparison

Assert – to assert that the likes are increasing

4.2 Test Single Product Leave Comment Form

4.2.1.1 Test type

___ Functional Test

4.2.1.2 System Under Test

System name: InaEssentials web-site, Single Product Page Leave Comment Form

Version: 9 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

4.2.1.3 Test Personnel

Name: Pavla Manova   Date: 9.01.2023   Time/h: 2

4.2.2 Test Summary

4.2.2.1 Results

Total number of test cases:	5
Total number of test cases passed:	4
Total number of test cases inconclusive:	1

Total number of test cases failed:	0
Total number of bugs found:	0

4.2.3 Test Cases

4.2.3.1 Test Form Showing on Click

Test Case ID	TC_ShowCommentForm	
Description	Test if form for leaving a comment is shown on clicking the button „Напиши коментар“	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Scroll to comments section	The product has comment section.
3	Click on „Напиши коментар“ button	Form for comment is shown
Test verdict	Pass	

Comments:

The test uses the following additional commands:

Assert element present – checks if the div with certain class is shown – this is the form we expect to see

4.2.3.2 Annealing Comment Form

Test Case ID	TC_HideForm	
Description	Test form hiding when „Анулиране на отзива“ button is pressed	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**

1	Open single product page.	https://inaessentials.com/products/promo-set-kosopad is loaded.
2	Scroll to comments section	The product has comment section.
3	Click on „Напиши коментар“ button	Form for comment is shown
4	Click on „Анулиране на отзива“ button	Form for comment is hidden
Test verdict	Passed	

Comments:

The test uses the following additional commands:

Execute script – to execute window.getComputedStyle(), which gets the div’s, wrapping the form, display value. If it is ‘none’ – then the form is hidden

Assert – to assert that a Boolean is true

4.2.3.3 Test Sending Comment Form Without data

Special Instructions

The user must be on single product page.

Test Case ID	TC_SendFormWithoutData	
Description	Test if error messages are shown and form is not sent.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Click on „Напиши коментар“ button	Form for comment is shown
3	Click on „Изпратете ревюто“ button	Error message is shown
Test verdict	Passed.	

Comments:

The test uses the following additional command – **assert element present** – for verifying error message is shown.

4.2.3.4 Test Sending Form with non-YouTube link

Special Instructions

The user must be on single product page.

Test Case ID	TC_SendNonYouTubeLink	
Description	Test Sending Form with non-YouTube link	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/rozova-maskina-za-normalna-do-zrqla-koja-60g is loaded.
2	Click on „Напиши коментар“ button	Form for comment is shown
3	Type non-YouTube link in the field for YouTube link	Can be typed in this field
4	Click on „Изпратете ревюто“ button	Error message is shown
Test verdict	Passed.	

Comments:

The test uses the following additional commands:

Assert text – to assert the error message text is correct

4.2.3.5 Send Comment Form With Correct Data

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SendCommentForm
Description	Test sending form with correct data
Applicable for	IE6, Google Chrome
Initial Conditions	Internet access

Test Step ID	**Test Step Description**	**Expected Result**
1	Open single product page.	https://inaessentials.com/products/promo-set-kosopad is loaded.
2	Scroll to comments section	The product has comment section.
3	Click on „Напиши коментар“ button	Form for comment is shown
4	Type in name input	Can be typed in this field
5	Type in email input	Can be typed in this field
6	Choose rate from 1 to 5	The stars are clickable
7	Type in heading input	Can be typed in this field
8	Type in review input	Can be typed in this field
9	Click on „Изпратете ревюто“ button	Success message is shown
Test verdict	Inconclusive. Inputs are generated with hash values every time and the test cannot be played twice	

Comments:

The test uses the following additional commands:

Assert element present – for success message

5 Test Sorting Of All Products

5.1.1 Test type

___ Functional Test

5.1.2 System Under Test

System name: InaEssentials web-site, All Products Page

Version: 12 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

5.1.3 Test Personnel

Name: Pavla Manova   Date: 12.01.2023   Time/h: 6

5.2 Test Summary

5.2.1 Results

Total number of test cases:	3
Total number of test cases passed:	0
Total number of test cases inconclusive:	0
Total number of test cases failed:	3
Total number of bugs found:	3

5.3 Test Cases

5.3.1 Sort Products By Name Ascending

Special Instructions

The user must be on all products page.

Test Case ID	TC_ SortProductsAlphabeticallyAsc	
Description	Test sorting by name in ascending alphabetical order.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open all products page.	https://inaessentials.com/collections/frontpage is loaded.
2	Click on sort dropdown menu	Dropdown menu is displayed.
3	Select sorting by name asc – 'По азбучен ред, A – Z'	Products list reloads
6	For every two products get their names	The names are stored
7	Assert the names are in correct order.	The names are in ascending order
8	If there is next page, go to it and repeat steps 6 and 7	All products' names are in correct order
Test verdict	Failed. Two products' names in wrong order found.	

Comments:

The test uses the following additional commands:

Execute script – create Boolean command whether to continue the while cycle or nor

While – cycle that repeats its body until there are no more pages

Execute script – to get all products on current page count with query selection

Execute script – save variable i for iterating through second while cycle

While – iterate through all products in page

Execute script – get previous (i-1) product name with query selection

Execute script – get current (ith) product name with query selection

Execute script – save in Boolean variable if the two texts are in ascending order

Assert – assert true name order

Execute script – to increase I with 1

Execute script – check if there is next page with query selection

If-Else – if there is next page – click on button for next page and continue the outer while cycle; if not – break the outer while cycle

5.3.2 Sort Products By Name Descending

Special Instructions

The user must be on all products page.

Test Case ID	TC_ SortProductsAlphabeticallyDesc	
Description	Test sorting by name in descending alphabetical order.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open all products page.	https://inaessentials.com/collections/frontpage is loaded.
2	Click on sort dropdown menu	Dropdown menu is displayed.
3	Select sorting by name asc – 'По азбучен ред, Z – A'	Products list reloads
6	For every two products get their names	The names are stored
7	Assert the names are in correct order.	The names are in descending order
8	If there is next page, go to it and repeat steps 6 and 7	All products' names are in correct order
Test verdict	Failed. Two products' names in wrong order found.	

Comments:

The test uses the following additional commands:

Execute script – create Boolean command whether to continue the while cycle or nor

While – cycle that repeats its body until there are no more pages

Execute script – to get all products on current page count with query selection

Execute script – save variable i for iterating through second while cycle

While – iterate through all products in page

Execute script – get previous (i-1) product name with query selection

Execute script – get current (ith) product name with query selection

Execute script – save in Boolean variable if the two texts are in descending order

Assert – assert true name order

Execute script – to increase I with 1

Execute script – check if there is next page with query selection

If-Else – if there is next page – click on button for next page and continue the outer while cycle; if not – break the outer while cycle

5.3.3 Sort Products By Rating

Special Instructions

The user must be on single product page.

Test Case ID	TC_ SortProductsRatingAsc	
Description	Test sorting products by rating.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open all products page.	https://inaessentials.com/collections/frontpage is loaded.
2	Click on sort dropdown menu	Dropdown menu is displayed.
3	Select sorting by rate asc – ‘Препоръчани’	Products list reloads

6	For every two products get their average rating attributes’ values	The ratings are stored
7	Assert the ratings are in correct order.	The ratings are in ascending order
8	If there is next page, go to it and repeat steps 6 and 7	All products’ rates are in correct order
Test verdict	Failed. A product in the middle with rating 5.0 was found.	

Comments:

The test uses the following additional commands:

Execute script – create Boolean command whether to continue the while cycle or nor

While – cycle that repeats its body until there are no more pages

Execute script – to get all products on current page count with query selection

Execute script – save variable i for iterating through second while cycle

While – iterate through all products in page

Execute script – get previous (i-1) product average rate, which is stored in data-average-rating attribute- with query selection

Execute script – get current (ith) product average rate, which is stored in data-average-rating attribute- with query selection

Execute script – save in Boolean variable if the two numbers are in ascending order

Assert – assert true rate order

Execute script – to increase I with 1

Execute script – check if there is next page with query selection

If-Else – if there is next page – click on button for next page and continue the outer while cycle; if not – break the outer while cycle

5.4 Traceability matrix

Test Case ID	**Bug Description**	**Note**
TC_SortProductsAlphabeticallyAsc	Products are not sorted in alphabetical ascending order	The bug was found in the page frontpage.html
TC_SortProductsAlphabeticallyDesc	Products are not sorted in alphabetical descending order	The bug was found in the page frontpage.html
TC_ SortProductsRatingAsc	Products are not sorted by rate in ascending order	The bug was found in the page frontpage.html

6 Test Performance

6.1.1 Test type

___ Performance Test

6.1.2 System Under Test

System name: InaEssentials web-site

Version: 15 Jan, 2023

Short description of the system:

E-commerce website for selling Bulgarian cosmetics, made entirely from natural ingredients. It is a family business, promoting the Bulgarian natural herbs and plants for cosmetic purposes.

6.1.3 Test Personnel

Name: Pavla Manova   Date: 15.01.2023   Time/h: 0.25

6.2 Test Summary

6.2.1 Results

Total number of test cases:	1
Total number of test cases passed:	0

Total number of test cases inconclusive: 0
Total number of test cases failed: 1
Total number of bugs found: 1

6.3 Test Cases

6.3.1 Test Responsiveness After Button Click

Special Instructions

Test Case ID	TC_ TestLoadTime	
Description	Test responsiveness after button click.	
Applicable for	IE6, Google Chrome	
Initial Conditions	Internet access	
Test Step ID	**Test Step Description**	**Expected Result**
1	Open all home page.	https://inaessentials.com/ is loaded.
2	Save current time	
3	Click on ‘Виж всички продукти’	
4	Save current time	
5	Assess elapsed time less than 3 seconds	
Test verdict	Failed. The page performance is slow	

Comments:

The test uses the following additional commands:

Execute script – to store current time

Assert – to check if elapsed time is less than 3s

6.4 Traceability matrix

Test Case ID	Bug Description	Note
TC_TestLoadTime	Page response is slow	The bug was found in the page index.html