**SOFTWARE VERIFICATION, VALIDATION AND TESTING**

TESTING DOCUMENTATION

Lunatik.io

Prepared by:

**Naida Fatić**

**Mirza Krupić**

Proposed to:

**Samed Jukić, Assist. Prof. Dr.**

**Aldin Kovačević, Teaching Assistant**

Date of submission

TABLE OF CONTENTS

Contents

[1. Introduction 3](#_Toc91440922)

[1.1. About the Project 3](#_Toc91440923)

[1.2. Project Functionalities and Screenshots 3](#_Toc91440924)

[2. Test Plan 3](#_Toc91440925)

[2.1. Scope 3](#_Toc91440926)

[2.2. Testing Environment and Tools 3](#_Toc91440927)

[3. Test Execution 3](#_Toc91440928)

[3.1. Test Scenario Name 3](#_Toc91440929)

[3.2. Example Scenario: Visit the Home Page 4](#_Toc91440930)

[9. Conclusion 5](#_Toc91440931)

[9.1. Testing Summary 5](#_Toc91440932)

[9.2. Final Thoughts 5](#_Toc91440933)

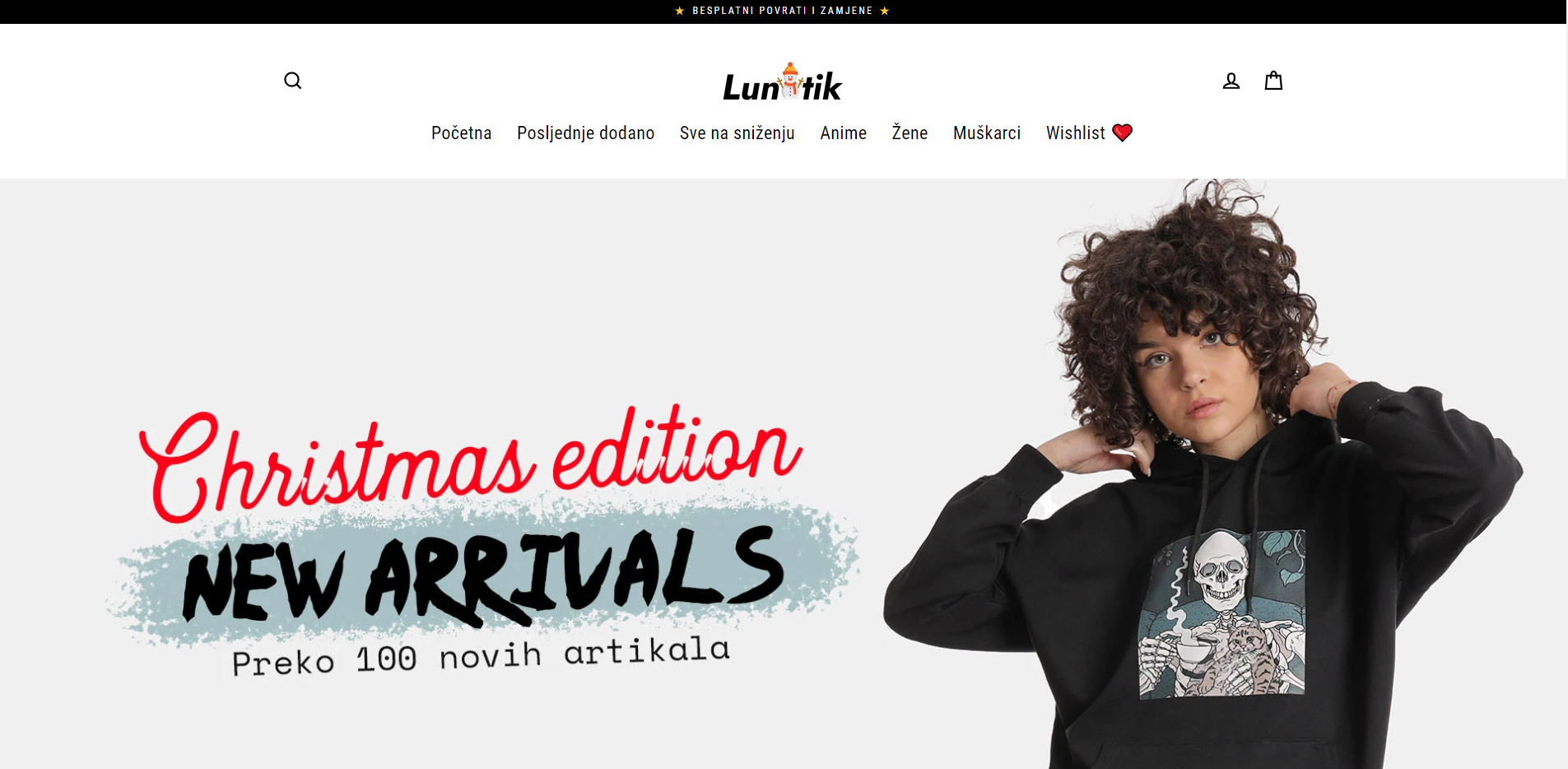
# 1. Introduction

## 1.1. About the Project

On this project we will work on [Lunatik.io](https://lunatik.io/) website. We got this idea from the previous shopping experience with this website, Naida made some orders via this site before. We will test most of the features just like we would do in real life.

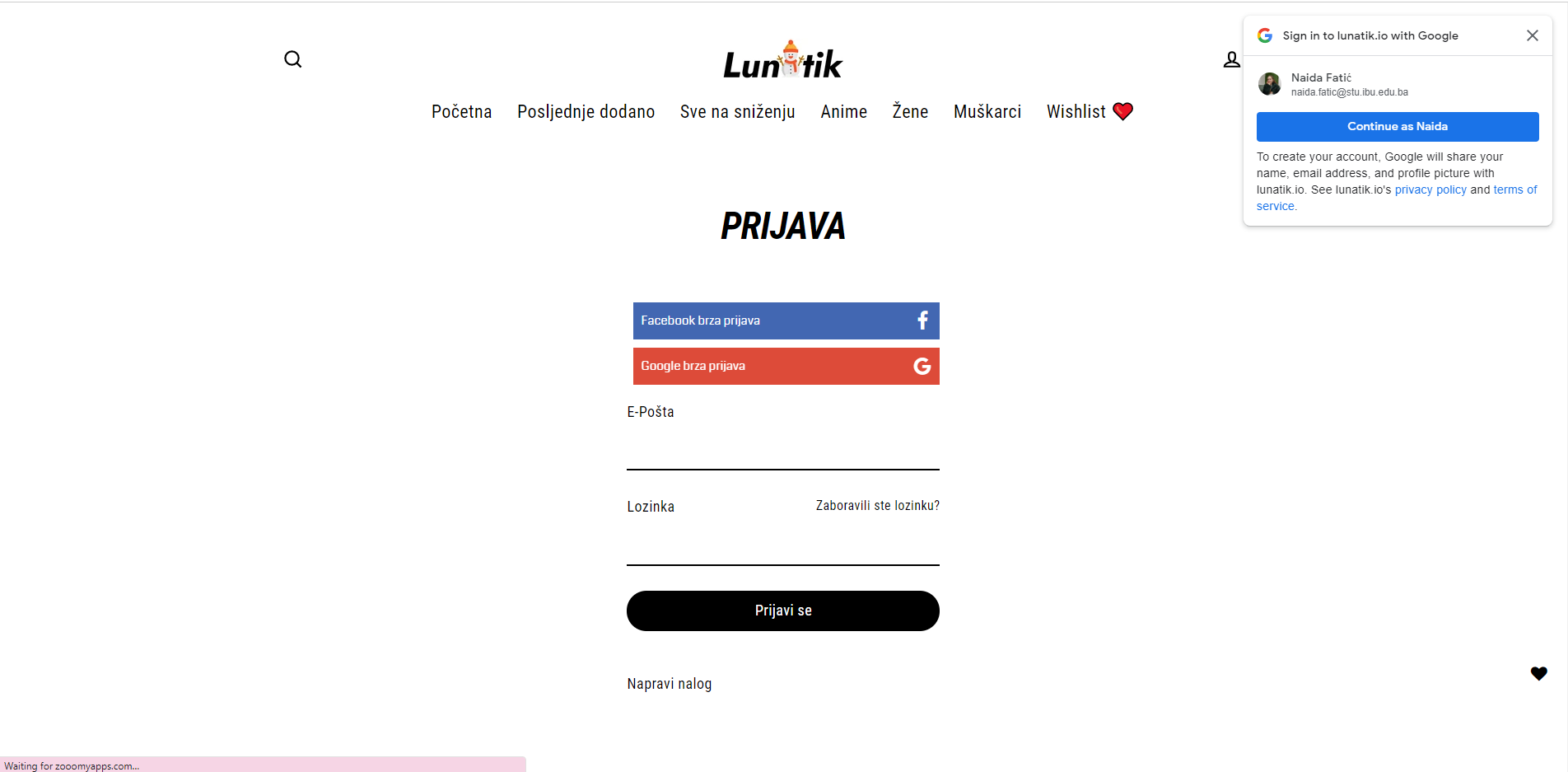
## 1.2. Project Functionalities and Screenshots

* Search of some items or categories

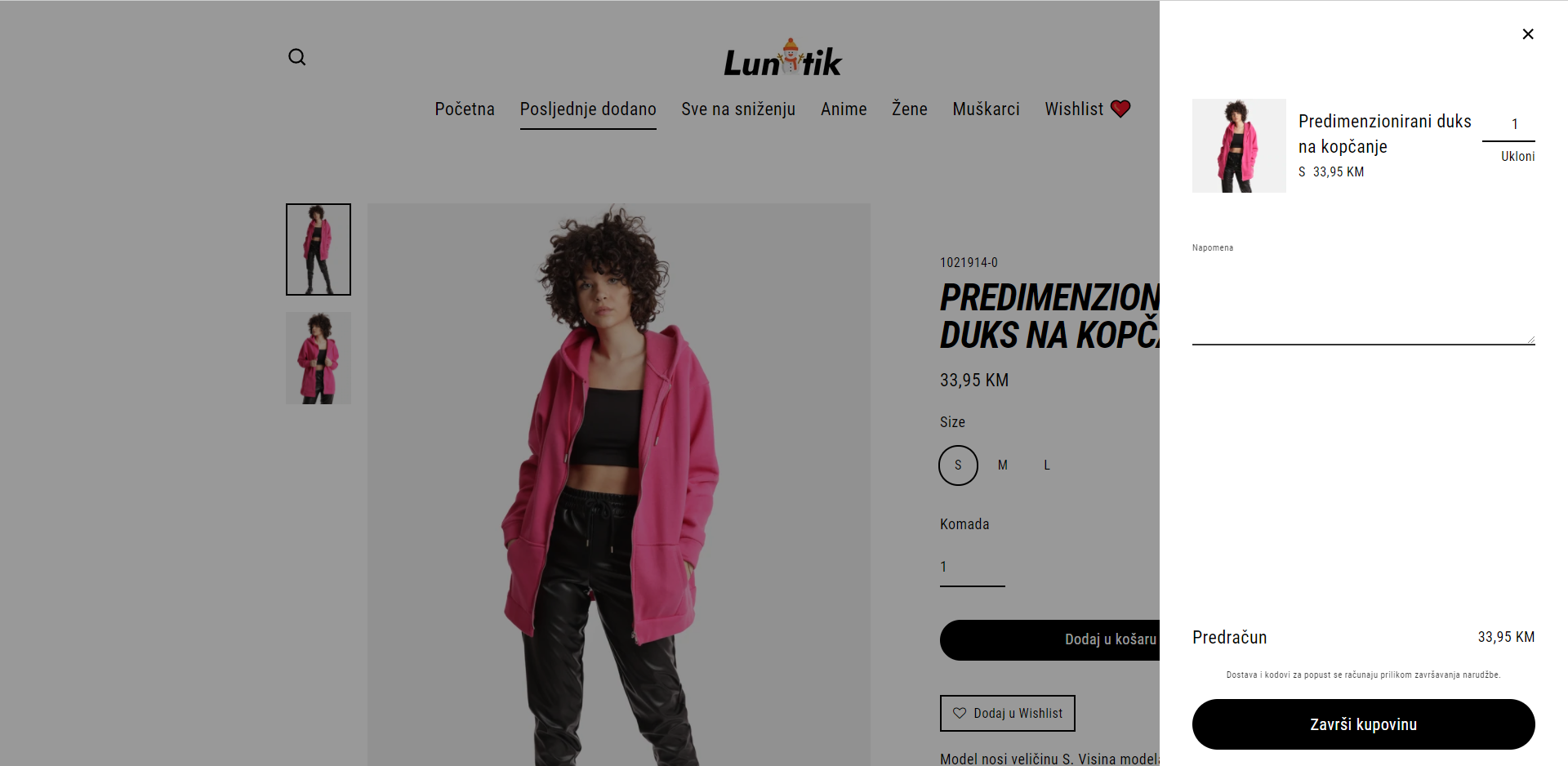




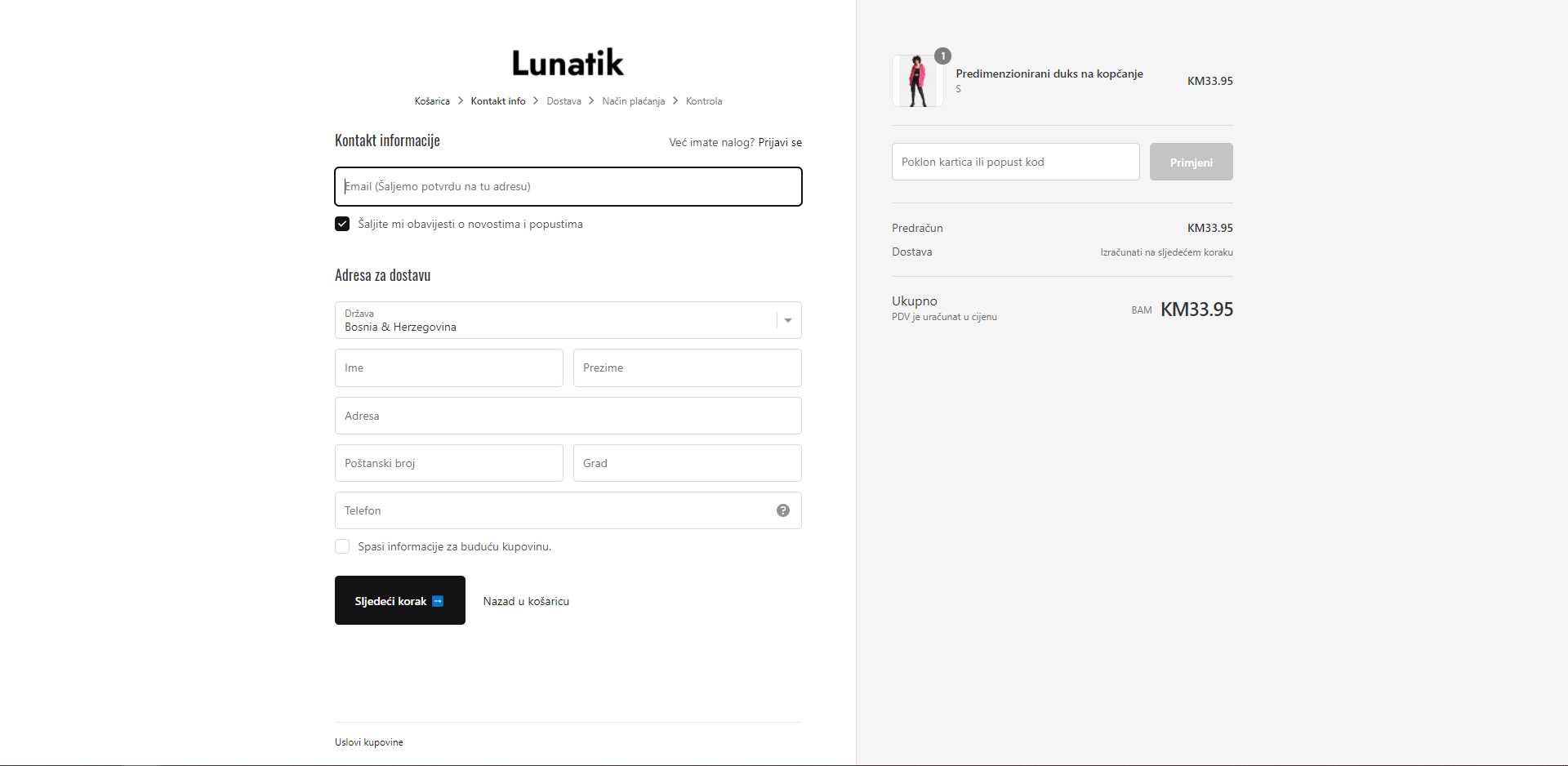
* Links to the categories
* Create an account



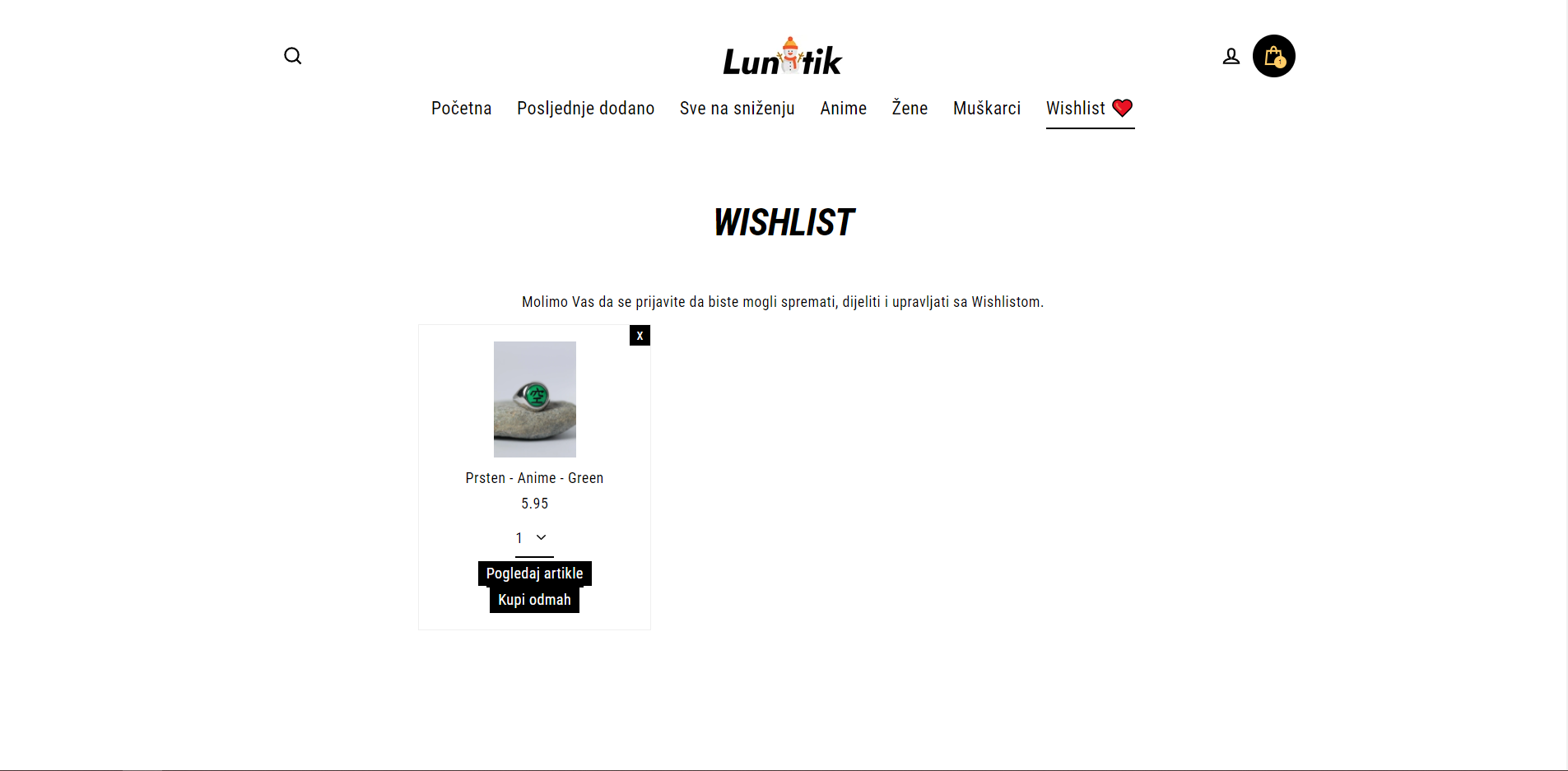
* Cart



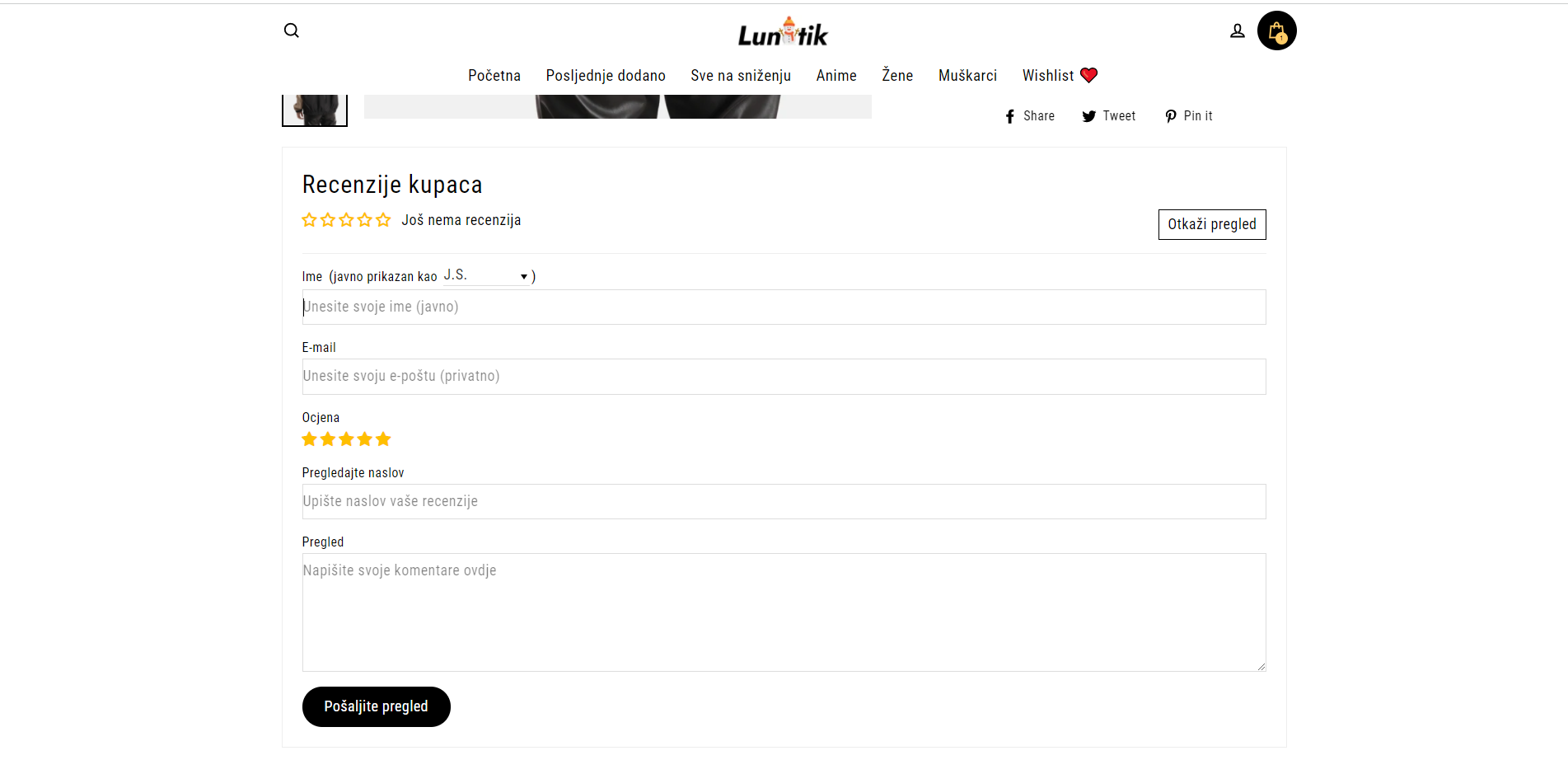
* Checkout



* Wishlist



* Review of the product



# 2. Test Plan

## 2.1. Scope

Describe what you are planning to test on the project: are you testing the entire application, or only focusing on some parts/features of it? Is there something you are not planning to test, etc?

Focus on this test will be on the shopping experience. We will test if the website is working properly: if we can go to all of the category links, if the prices are correctly added to the cart, we will also proceed to the checkout but we will not test the checkout feature. Also, we will test the wish list and add to cart feature. We are planning just to test desktop version(maximize window).

## 2.2. Testing Environment and Tools

Describe what kind of tools and/or frameworks are you going use during your testing, what programming language you are using, etc.

We will be using Selenium tool for the test and write the tests in Java language.

# 3. Test Execution

In the “Test Execution” section, you need to list your **test scenarios** and **test cases**. *Test scenario* is a statement whose purpose is to inform what particular area of the application will be tested. Test scenarios consist of multiple *test cases*, which test specific parts of an application. You can use the following template approach for test scenarios and test cases, but you can also use a custom template or your own “design”, as long as you have all the necessary information.

## 3.2. Test Scenario Name DELETE AFTER

Give a brief one-two sentence description of the test scenario. Afterwards, list the test cases in that test scenario. You can use the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Short test name | | | | |
| **Description:**  Test description | | | | |
| **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. Number the steps needed to execute this test case.  2. Like this. | **Test Data:**  - is there any specific input data you are using for this test (e.g. some input field values, credentials, etc.) | **Expected Result:**  What was the expected result of this test? | **Actual Result:**  What was the actual result of this test, once you executed it? | **Status:**  PASS or FAIL (you can color code them) |
| **Notes:** Are there any notes about this test you would like to add? If not, leave this blank. | | | | |

Provide a screenshot of your test code after the test case table.

The following section will be one test scenario example for you, with two tests.

## 3.1. Example Scenario: Visit the Home Page

Users want to be able to visit the home page, and get a quick glance at the most important features it has to offer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test top bar links | | | | |
| **Description:**  Check if all top bar items are clickable and lead to their respective page. Click links one after another. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the landing page  2. Check title of page  3. Check if all links have been rendered  4. Click on each link individually  5. Check if the URL of the opened page corresponds to the link in the sidebar | **Test Data:** | **Expected Result:**  The user is taken to the correct new page when clicking on the link. | **Actual Result:**  The user is taken to the correct new page when clicking on the link. | **Status:**  PASS |
| **Notes:** | | | | |





## 3.2. Example Scenario: EXAMPLE FAIL DELETE AFTER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test “Find out more” button | | | | |
| **Description:**  Check if “Find out more” button works and leads to proper page (either a product page, or an “About us” page) | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Wait until “Find out more” button is loaded  2. Click on the button  3. Check if it leads to the expected page | **Test Data:** | **Expected Result:**  The user is taken to a new page. | **Actual Result:**  The page gives a 404 error because the requested page does not exist. | **Status:**  FAIL |
| **Notes:** | | | | |



## 3.2. Example Scenario: Test “Collections” section links

Users want to be able to visit the home page and to click on the links that home page offer in the collection section that is at the bottom of the page.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test “Collections” section links | | | | |
| **Description:**  Check if link on “Collections” section are clickable and if they take user to the correct page. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the landing page  2. Check title of page  3. Scroll down to the “Collections” section  4. Check if all links have been rendered  5. Click on each link individually  6. Check if the URL of the opened page corresponds to the link in the sidebar | **Test Data:** | **Expected Result:**  “Collections” sections is rendered properly and user is taken on the correct page when clicking each of the links in this section. | **Actual Result:**  “Collections” sections is rendered properly and user is taken on the correct page when clicking each of the links in this section. | **Status:**  PASS |
| **Notes:** | | | | |



## 3.3. Example Scenario: Search

Users want to be able to search some items in order to see what site offers faster.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test search feature | | | | |
| **Description:**  Check if search feature works both for items which exist and items which doesn’t exist in the database. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the landing page  2. Click on the search button in the upper left corner  3. Input search parameter for item which exists in the database  4. Click on the search button  5. Check if system returned any results  6. Click on the search button in the upper left corner  7. Input search parameter for item which doesn’t exists in the database  8. Click the enter button  9. Check if system returned any results | **Test Data:**  “majice” and “aass” text we used for the search input. | **Expected Result:**  Search should return certain amount of items if item’s name contains phrase we entered in the search field or it should not return any results if there is no item which contains in its name phrase we searched for. | **Actual Result:**  Search returns certain amount of items if item’s name contains phrase we entered in the search field or it doesn’t not return any results if there is no item which contains in its name phrase we searched for. | **Status:**  PASS |
| **Notes:** | | | | |



## 3.4. Example Scenario: Item sort

Users want to be able to sort items by some value in order to search see items.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test item sorting | | | | |
| **Description:**  Check if sorting on shop pages work properly. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the “posljednje dodano” page  2. Check current items  3. Select next option in dropdown menu  4. Check if items in the item list changed  5. Repeat the process for every menu item | **Test Data:** | **Expected Result:**  Whenever we change option in the dropdown menu, item list should change order of items rendered. | **Actual Result:**  Whenever we change option in the dropdown menu, item list changes order of items rendered. | **Status:**  PASS |
| **Notes:** | | | | |



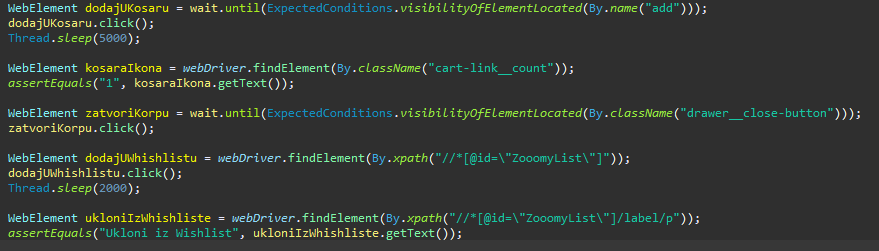


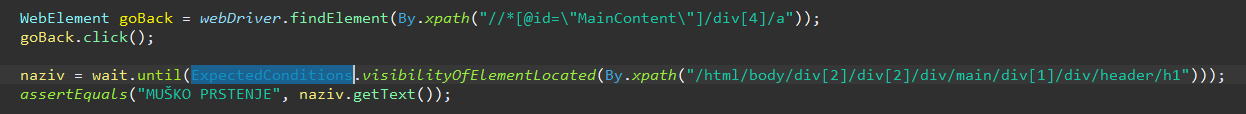
## 3.5. Example Scenario: Item page

Users want to see items in order to get their description, add them to cart, add them to Wishlist or to leave review.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test item page | | | | |
| **Description:**  Check if item page works properly. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the home page.  2. Navigate to the “Muškarci” link on the top bar.  3. Hover over it and hover over “Nakit” link  4. Click on the “Prstenje” link  5. Check if the title of the page says “Muško prstenje”  6. Check if the price of first item is 15.95  7. Click on the first item and check the price of the item if it is 15.95  8. Add item to the cart  9.Close cart asset that cart number is 1  10. Add item to the Wishlist and check if the button says “Ukloni iz Wishlist”  11. At the bottom of the page we can go back to the previous page we are going to test that button | **Test Data:** | **Expected Result:**  The “Muško prstenje” page should appear, we should be able to click on the item and add it to the cart or Wishlist. After all of that we want to be able to go back to the “Muško prstenje” page. | **Actual Result:**  The page loaded, we are able to click on the item and add it to the cart or Wishlist. The cart number incremented. We were able to go back to the “Muško prstenje” page. | **Status:**  PASS |
| **Notes:** | | | | |







## 3.5. Example Scenario: Successful registration

In order for user to be able to add items to wishlist, to save addresses and to track down history of his purchases, he has to register so he can use that account for keeping and accessing all of those informations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test successful registration | | | | |
| **Description:**  Check if registration system passes when user enters valid data. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the login page by clicking account button in upper right corner.  2. Go to registration page by clicking on “napravi nalog” located bellow login form  3. Fill in form with valid data  4. Click on “Napravi” button located bellow the form  5. Get redirected to home page while being logged in in the same time. | **Test Data:** | **Expected Result:**  User should be able to create new account on the webpage after entering valid data into the registration form and clicking “Napravi” button. | **Actual Result:**  User is able to create new account on the webpage after entering valid data into the registration form and clicking “Napravi” button. | **Status:**  PASS |
| **Notes:** | | | | |



## 3.5. Example Scenario: Email already used during registration process

During the registration process, user might enter the email which is already used by some other account. In that case we want to test if the user gets error prompt which says that email is taken with link to go to password recovery page and we also need to test if that link works.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test item email already used during registration | | | | |
| **Description:**  Check if system of detecting if user entered email which is already in used and test if link which sends user to password recovery page works. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the login page by clicking account button in upper right corner.  2. Go to registration page by clicking on “napravi nalog” located bellow login form  3. Fill in form including email which is already used previously to create some other account  4. Click on “Napravi” button located bellow the form  5. Get error prompt which says that this email is in use and prompt includes link which should lead user to password recovery page  6. Click on password recovery link inside error prompt | **Test Data:** | **Expected Result:**  User should be prompted with error that email is already in use when he clicks on “Napravi” button. When user clicks on “resetuj lozinku” he should be redirected to password recovery page. | **Actual Result:**  User is prompted with error that email is already in use when he clicks on “Napravi” button. When user clicks on “resetuj lozinku” he is not redirected to password recovery page. He is redirected to the URL which resembles password reset but that URL leads user to the homepage rather than password recovery page. | **Status:**  FAIL |
| **Notes:** | | | | |

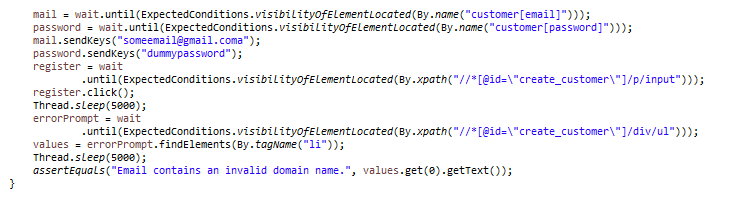


## 3.5. Example Scenario: Registration form validation

In order for user to be able to register successfully, developer should implement form validation for registration form in order to guide user to enter valid data inside the form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Registration form validation | | | | |
| **Description:**  checking all possible invalid data scenarios inside registration form validation. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to the login page by clicking account button in upper right corner.  2. Go to registration page by clicking on “napravi nalog” located bellow login form  3. Fill in form without inputting any data  4. Click on “Napravi” button located bellow the form  5. Get redirected to the same page  6. Check error prompt  7. Fill in form without inputting email  8. Click on “Napravi” button  9. Get redirected to the same page  10. Check error prompt  11. Fill in form without inputting password  12. Click on “Napravi” button  13. Get redirected to the same page  14. Check error prompt  15. Fill in form by entering invalid email  16. Click on “Napravi” button  17. Get redirected to the same page  18. Check error prompt | **Test Data:** | **Expected Result:**  When user tries to register with invalid data inside the form he should not be allowed to do it and he should be prompted with the appropriate error prompts. | **Actual Result:**  When user tries to register with invalid data inside the form he is not allowed to do it and he is prompted with the appropriate error prompts. | **Status:**  PASS |
| **Notes: This test works but when user enters invalid form of email he is prompted with error message on English language while every other error prompt is in Bosnian which is not a good practice since this is supposed to be website for the Bosnian people.** | | | | |

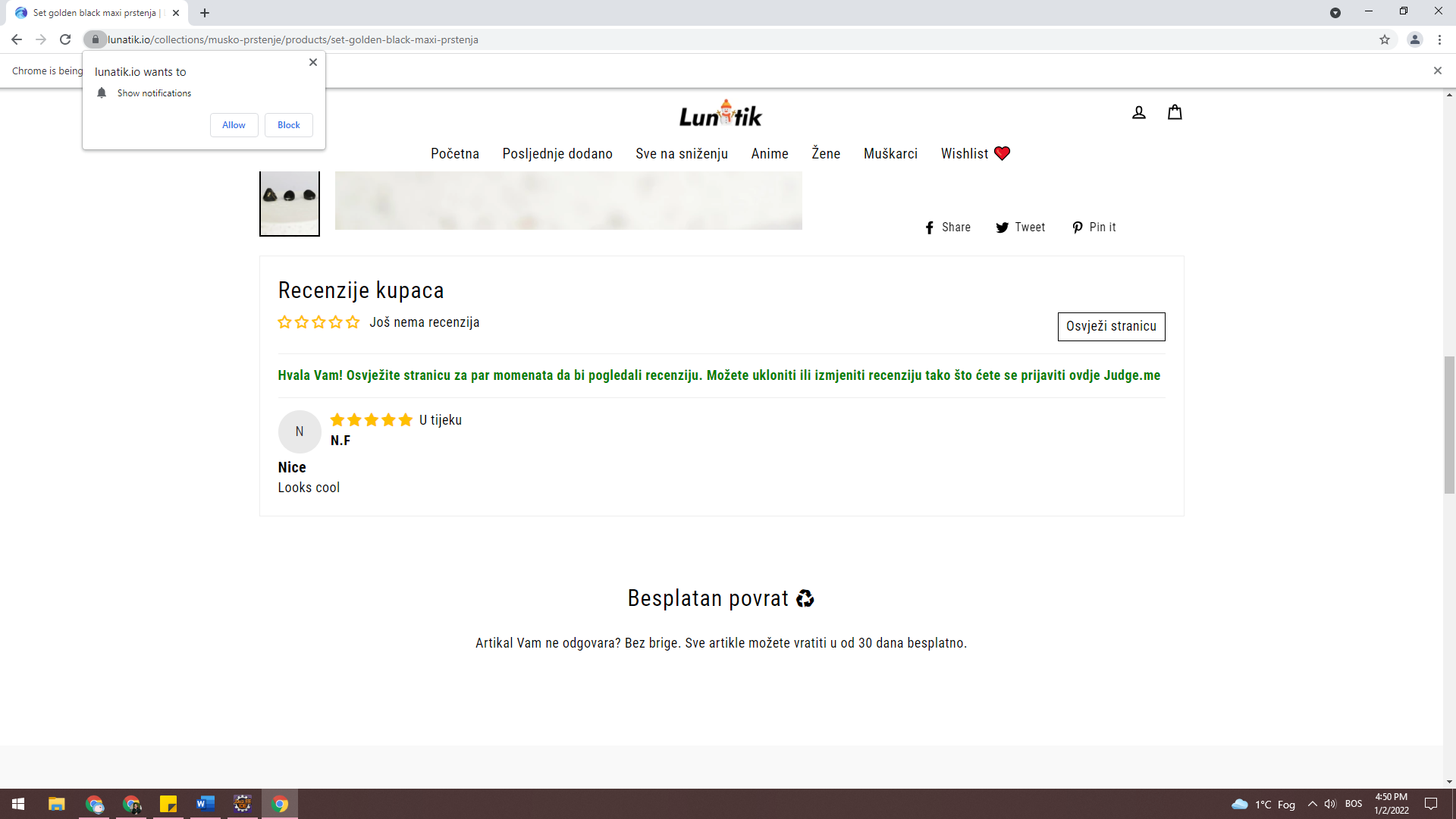




## 3.6. Example Scenario: Review comment

Users want to be able to leave review of item in order to state my opinion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name:** Test review of the item | | | | |
| **Description:**  Check if user can leave review on item. | | | | |
| **Pre-condition(s):** | | | | |
| **Test Steps:**  1. Go to item that we used in previous task.  2. Go to the bottom and click on the “Napisite svoje misljenje”  3. Populate all of the fields and press “Pošaljite pregled”  4. On purpose left the name part empty and assert that “Ovo polje je obavezno.” text appear.  5. After that we input name and send review | **Test Data:**  We user “N.F” for the name input, “[example@example.com](mailto:example@example.com)” for the email input, “Nice” text for the “Naslov” input and “Very cool” for the “Pregled” input. | **Expected Result:**  We should be able to left review and see our review after we refresh. | **Actual Result:**  The review is there we can see it on this [link](https://lunatik.io/collections/musko-prstenje/products/set-golden-black-maxi-prstenja). | **Status:**  PASS |
| **Notes: We are not going to write test that this review pass because we don’t want to spam page. The prof that this test passes is below and also on the link. If we want to write this test we can do that with the assertEquals on the green text below.** | | | | |





# 9. Conclusion

Sometimes test fails from unknow reasons but after we run it again it works fine.

## 9.1. Testing Summary

Provide a summary of all your executed tests. Something like this would be alright:

|  |  |  |  |
| --- | --- | --- | --- |
| **Testing Tool** | **Total Tests** | **Passed Tests** | **Failed Tests** |
| Framework or tool(s) used for testing. If you wrote tests in multiple different tools, create a row with the number of tests for each of them. | total number of tests | # of passing tests | # of failing tests |

If there are failing tests, provide a list of their names. Ideally, you can also create bookmarks in the document and make links to those failing test cases.

## 9.2. Final Thoughts

Provide some closing statements or your final thoughts about the project you tested. Did you find that it was implemented well? Did you notice any obvious mistakes or flaws in it? Are there some recommendations or observations you would note for the site?