**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)

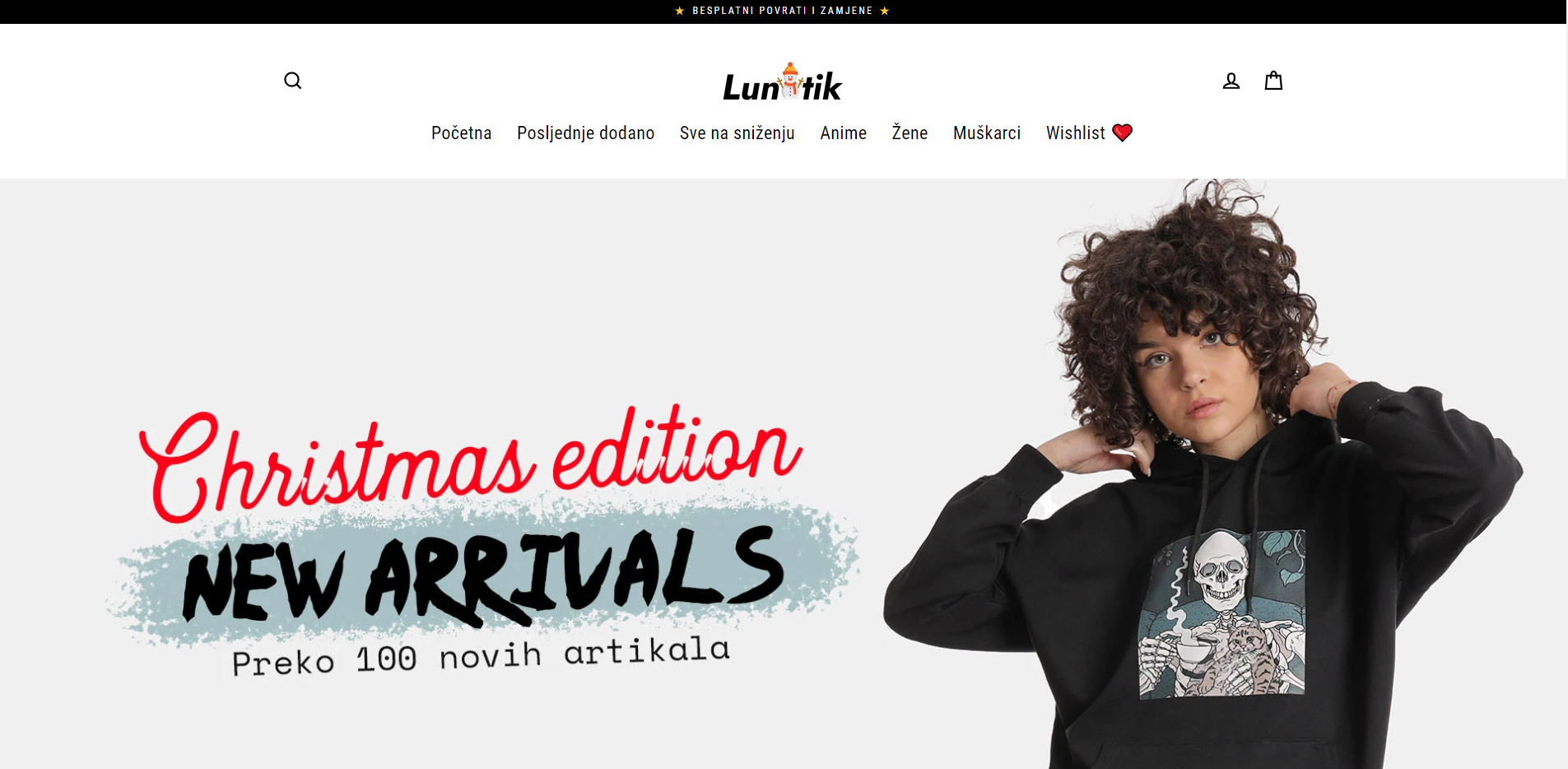
# 1. Introduction

## 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.

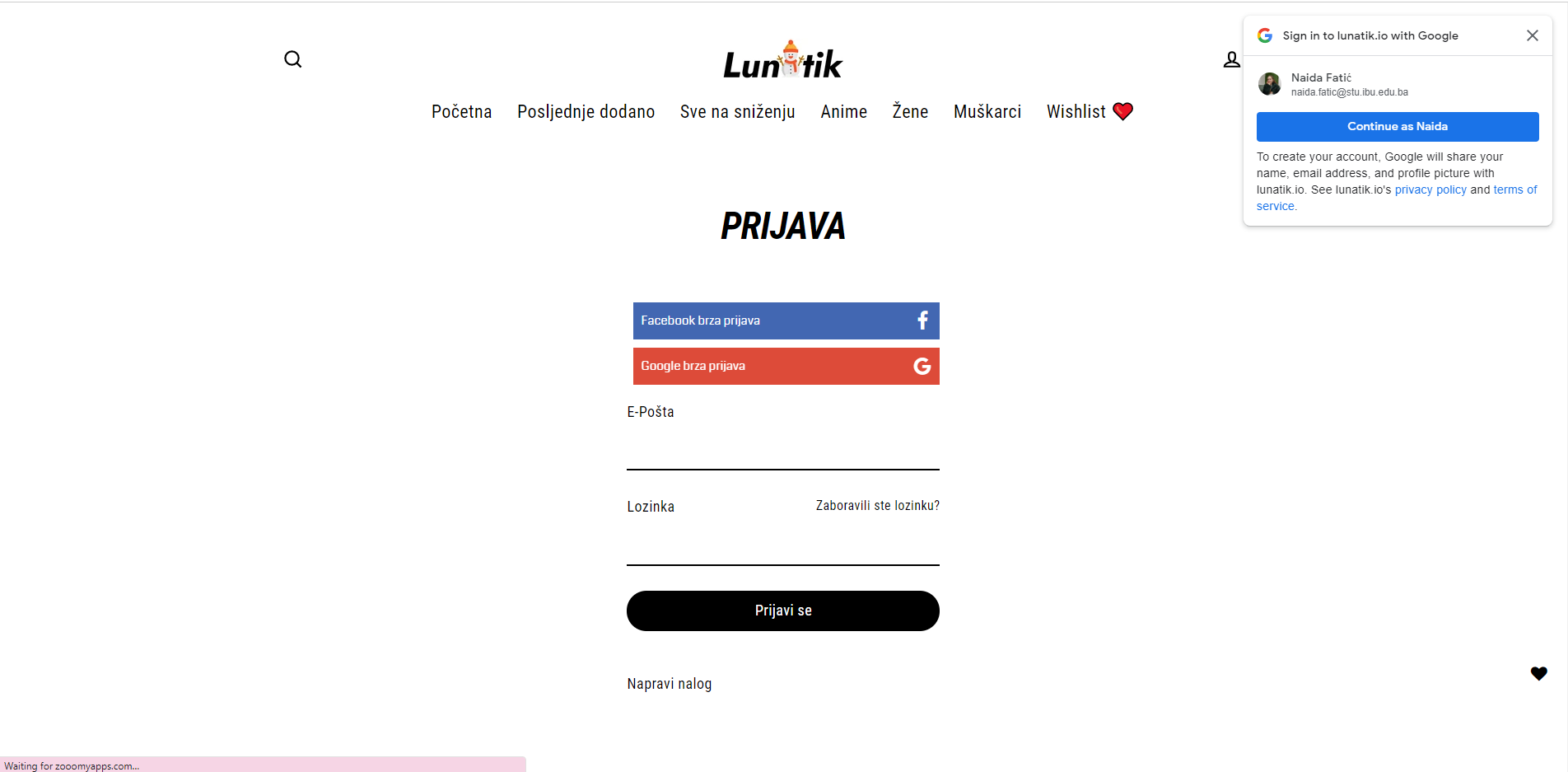
## 1.2. Project Functionalities and Screenshots

* Search of some items or categories

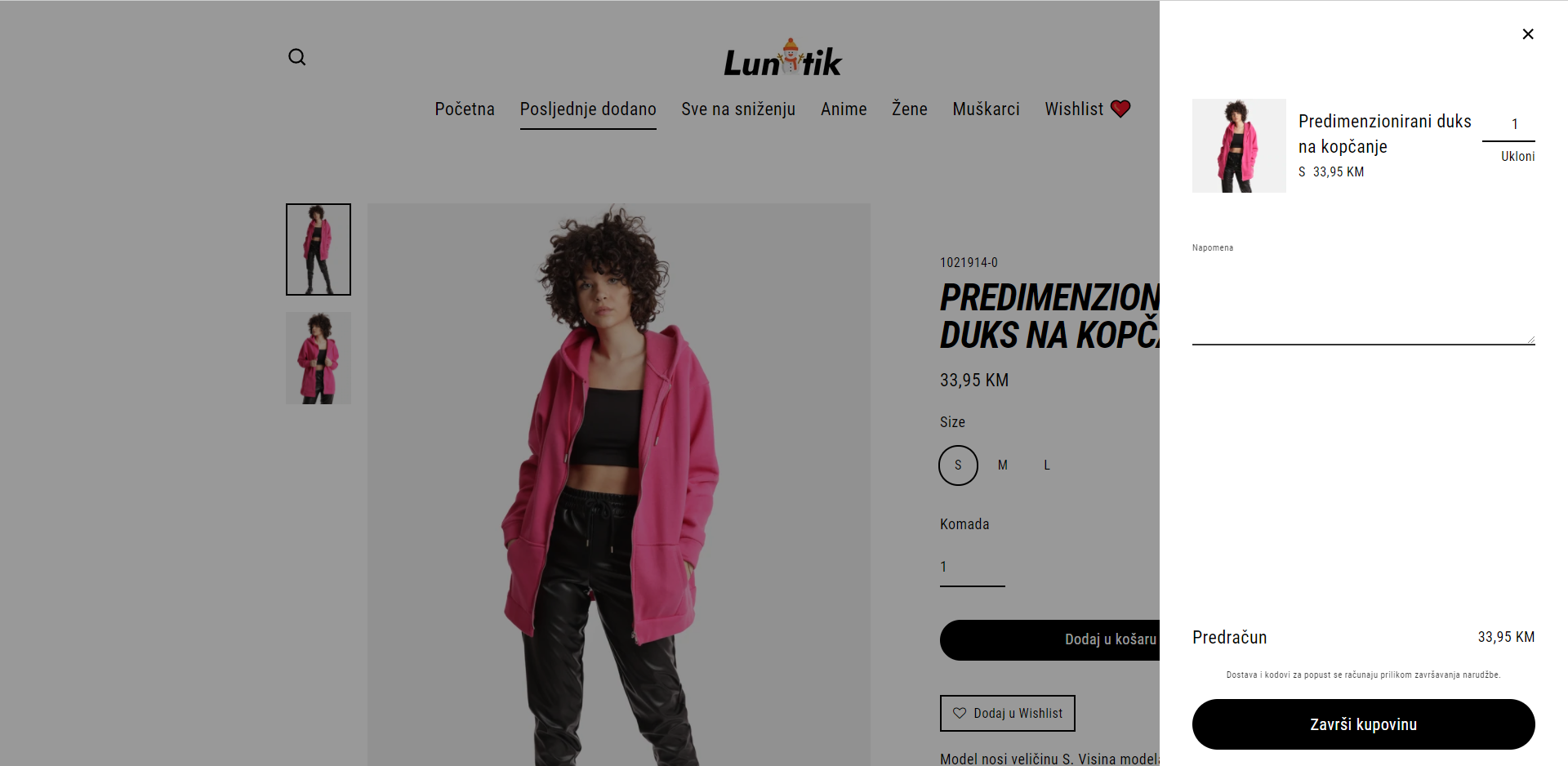




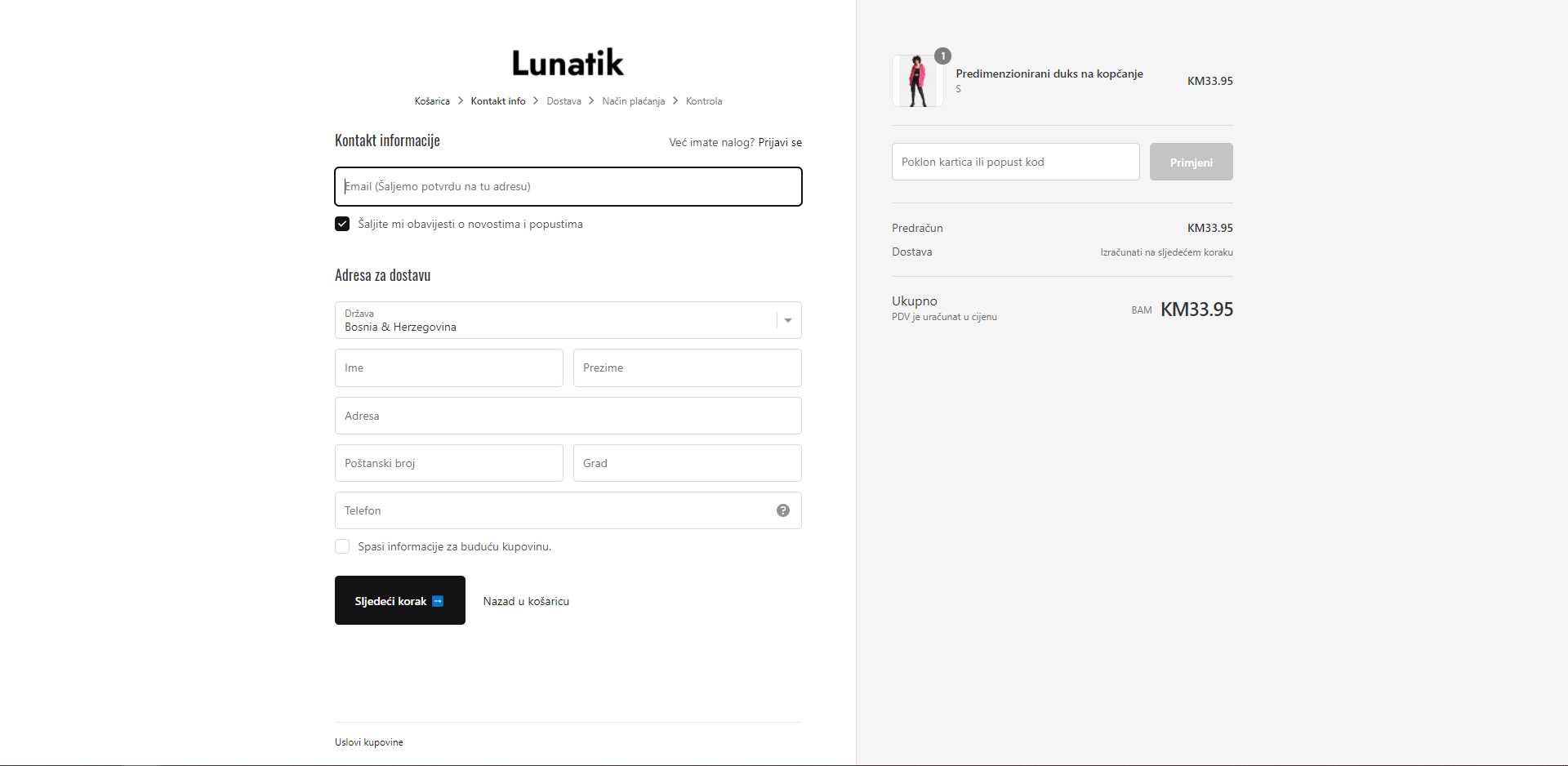
* Links to the categories
* Create an account



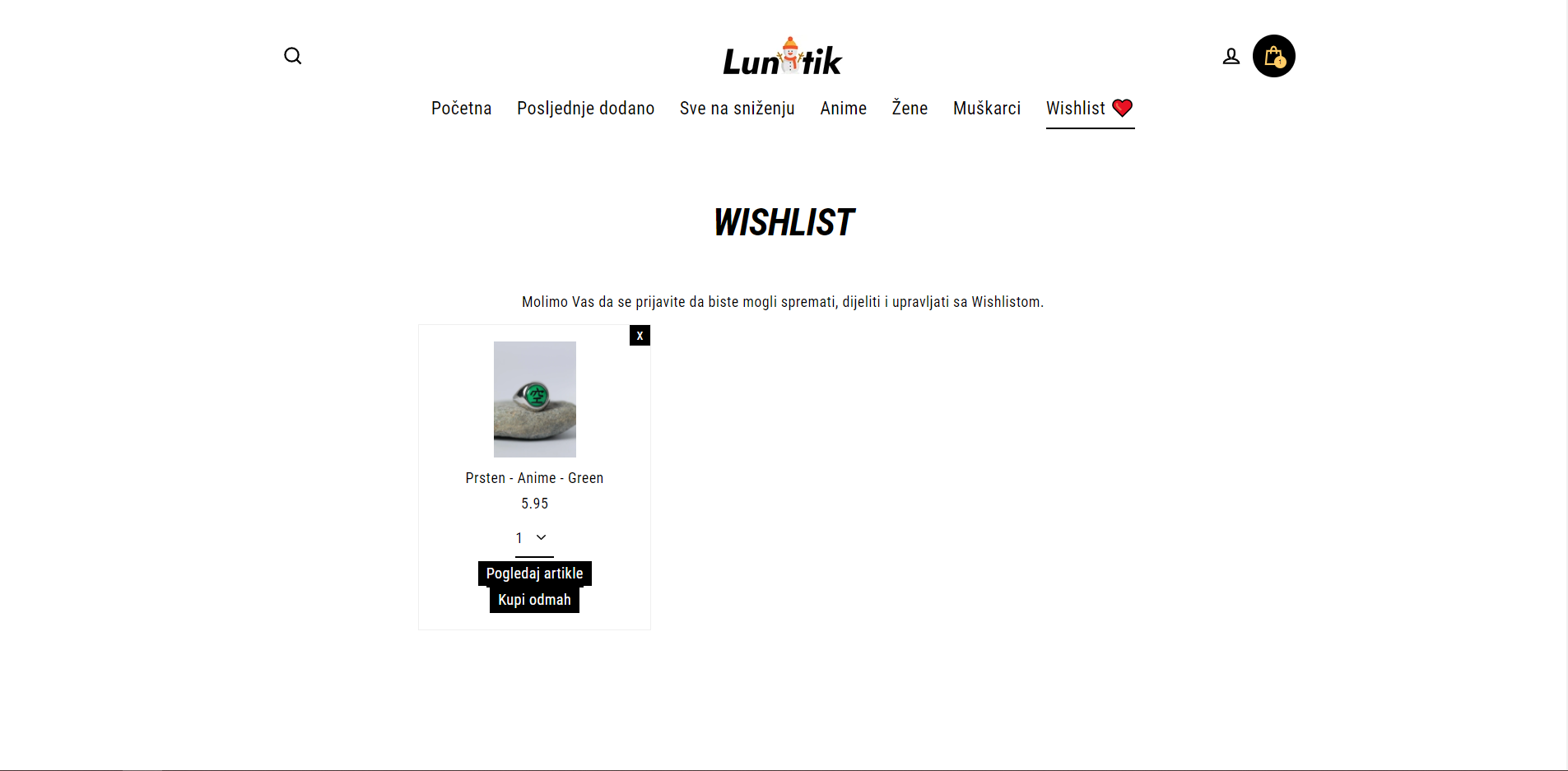
* Cart



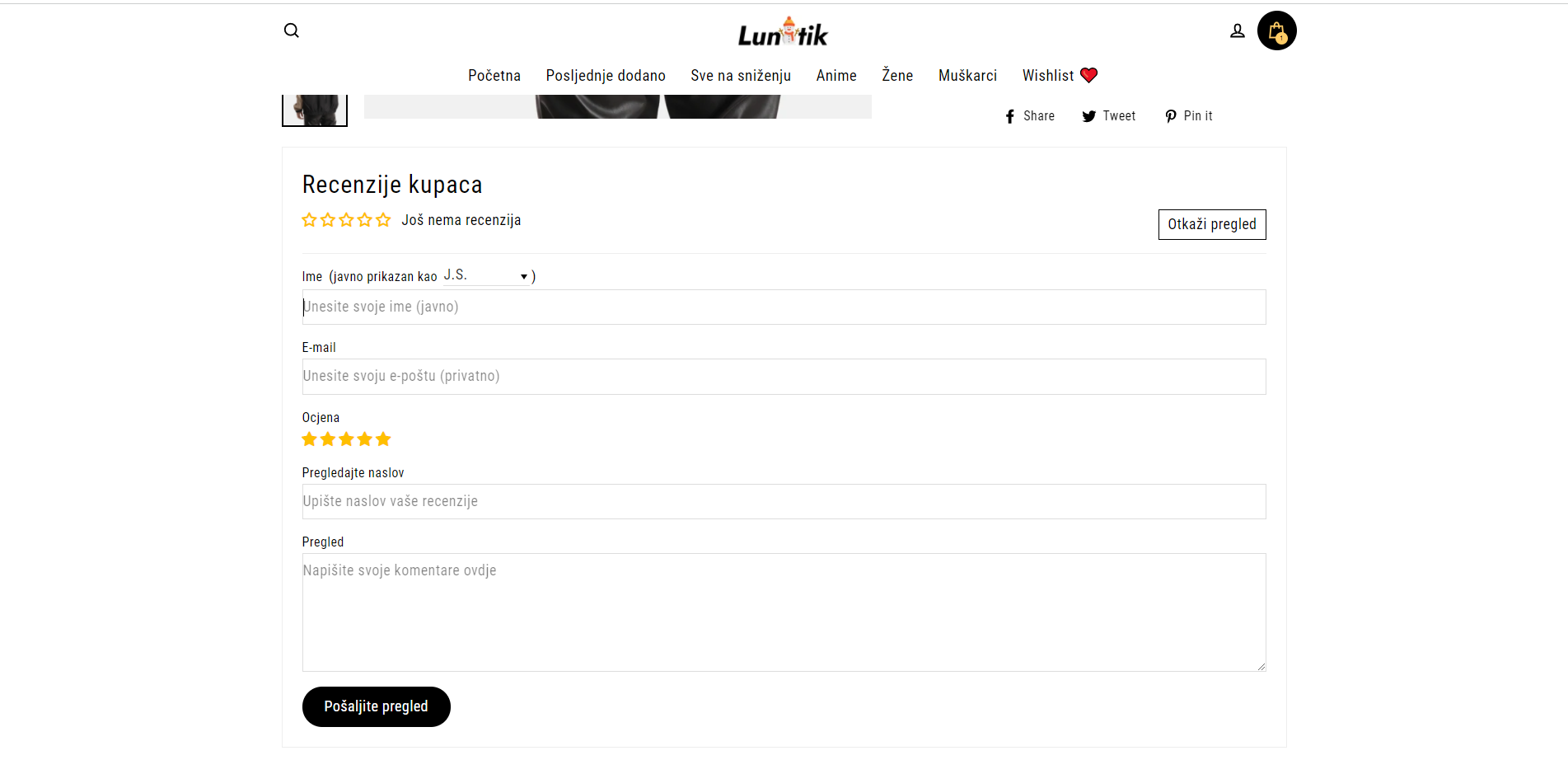
* Checkout



* 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.

## 3.1. Test Scenario Name

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.

## 3.2. 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:** | | | | |





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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:** | | | | |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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:** | | | | |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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:** | **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:** | | | | |



...

# 9. Conclusion

## 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?