Testing Plan for Orinoco Camera App

Prepared for: Sheila Pruca, CEO Orinoco

Prepared by: Maya Küchle, Junior Web Developer

ORINOCO

May 21, 2020

Proposal number: 123-4567

# Overview

## Introduction

This plan describes the testing approach that will serve as the fundamental strategy for the Orinoco camera app. Orinoco offers specific, themed apps which each sell one group of products. The purpose of this app is to sell vintage cameras that they can buy over this app.

The product requirements for the To-Do app are as follows:

A user can see details of an individual camera on redirected single product page

A user can add a camera to the cart

A user can access the cart page and checkout

A user arrives on order confirmation page

A user can ask for more information on contact page

## Features to Test

1. A user can see all cameras on the list view page with the correct informations of each individual camera
   * Frontend:
     + The user can go on the list view page and see all cameras available
     + Each camera has the correct description and price
   * Backend:
     + Fetching data from endpoint from API (API:/cameras) adds details of cameras to the webpage
2. A user can see details on the redirected single product pages with the correct camera selected
   * Frontend:
     + User can click on « Info » on the list view page to go to specific single product pages
     + New page appears with detailed information of the camera which will dynamically show the item selected by the user, display a description and price in dollars, and allow users to personalize the product and add it to their cart
     + The « Info » button directs you to the correct camera page
   * Backend:
     + Calling endpoint from API (API:/cameras) adds details of camera to the webpage by using URL query parameters and returns item corresponding to given \_id
3. A user can add a camera to the cart
   * Frontend:
     + User can add camera to the cart by clicking « Add to Cart » on single product page
   * Backend:
     + We declare a variable to add an "addcart" button that adds the item to the cart with the addEventListener 'click'
     + We place camera in cart by uploading the camera id to the localStorage (with the number of times selected with the TouchSpin) with the « localStorage.setItem » feature
4. A user can determine how many times one camera can be added
   * Frontend:
     + User can specify on single product page how many times he/she would like to add the camera to the cart
     + The user can click on the touchspin « + » and « - » in order to do so
     + You can add at least one camera
   * Backend:
     + The touchspin allows to add each item with a minimum of 1 and maximum of at least 20 times
5. A user can not add 0 items to the cart
   * Frontend:
     + The touchspin on the single product page makes sure that there can always be added at least 1 item to the cart
     + Adding 0 items is disabled
   * Backend:
     + The touchspin is disabled to select 0. The minimum is 1
6. A user can access the cart page and checkout
   * Frontend:
     + User can click on cart icon to access cart page
     + The cart page shows a summary of products in the cart, the total price, and a form with which to submit an order
     + Only after filling out the submit form, can you checkout of the page
   * Backend:
     + Using the JavaScript localStorage functionality, the elements inside the cart, get called with the « localStorage.getItem » feature
     + Only by filling out the submit form correctly, does the checkout button at the bottom activate to redirect to the confirmation page (order.html)
7. A user gets alerted when submit form was not correctly filled out
   * Frontend:
     + User has to submit form on cart page before checking out
     + If form was not correctly filled out, user gets alerted by red instructions under the submit form, explaining which part of the form was not filled out correctly
   * Backend:
     + Submit form consists of if statements that have to be filled out correctly
     + If variable « isValid » returns as « false », then red alerts are shown
8. A user arrives on order confirmation page
   * Frontend:
     + After clicking « Checkout » on the cart page, the user gets redirected to an order confirmation page, that thanks the user for their order, showing the total price and the order ID returned by the server
   * Backend:
     + With the JSON request containing a contact object and a products array, the API returns the contact object, products array and total price
9. A user manages to navigate through the website
   * Frontend:
     + Website navigation consists of burger menu on the top left and access to the cart page on the top right cart icon
     + Options of the burger menu are « Sales » and Contact
     + On the Sales Page you have all the cameras and can navigate through each one by clicking on their individual « Info » button
10. The website is responsive so that the user can use the website on the computer, tablet or phone
    * Frontend:
      + The website adjusts to different screens. The website is therefore not limited to solely work as an app

## Features Not to Test

1. A user can remove items from cart
2. A user can change quantity of individual cameras in cart
3. A user can add cameras to the cart on the list view page