Test plan – Evidencija Racunarske Opreme

ITAcademy

TESTER

Alen Suljakovic,

QA Tester

MENTOR

Atif Becirbasic,

Senior Dev

Januar, 2024
Tuzla

1. Introduction

This test plan will cover the basics of the process of testing Evidencija Racunarske Opreme at https://puppies-closet.com/evidencija/. This test plan is based off the Documentation provided to us and knowledge shown during our classes at ITAcademy.

Testing and evaluating the functionality of aformentioned webpage was handled manually and automatically. Using the following tools allowed us to access, manipulate and test our webpage in an automated manner:

1. Visual Studio Code as our code editor
    a. The Selenium library in order to manipulate our webapge
    b. The OpenPyXl library for data processing and manipluating it through excel
2. Postman for API request testing and verifying[1]
3. Apache Jmeter for load testing and performance analysis
4. OBS Open Broadcaster Software for screen recording and video documenting issues

All the tests as well as this document are named and located inside the "Zavrsni radA.Suljakovic" folder for easy access, reading and modification.

[1] API GET requests were sent to rendomuser.me webpage

2. Website Overview

All the functions and features https://puppies-closet.com/evidencija/ provides are written with a intricate level of detail in the “Dokumentacija - Evidencija racunarske opreme” file (here). The author provides a rich description of elements which allowed us to understand and test the webpage better and with a higher level of accuracy on what each element should do and react.

In this chaper I will go over which features were tested and what should their basic functionality be.

I. **Header and Footer**
The header located at the top of the landing page allows us to navigate through the rest of the webpage, to access subsites like Equipment, Reports, ect.
The footer is located at the bottom of the landing page allows us to find the user manual, support and contact info

II. **Employees**
The Employees section allows admin users to create, edit and delete employee entires as well as add and manage equipment the empolyees have taken or returned.

III. **Equipment**
The Equipment section allows us to create equipment entires as well as edit or delete them if they are not taken by any employees.

IV. **Reports**
The Reports section allows us to create reports on equipment based on Organization Units, Offices, Employees or just to get a report on all the currently free equipment.

V. **Type/Brand of Equipment**
The Type and Brand of Equipent allows us to create new entries on the new types and brands of equipment as well as edit or delete them.

VI. Office/Organization Unit

The Office and Organization Unit section allows us to create new entries on the new Offices and Organization Units as well as edit or delete them.

VII. Account management

The account management section allows us to create, edit or delete user or admin accounts.

3. Used Functions and Apps Overview

Under this segment we will scrape over the basic functionalities of Selenium and OpenPyXl capabilities. (Selenium Documentation) (OpenPyXl Documentation)

Selenium allows us to manipulate web browsers such as Mozzila Firefox, Google Chrome, Micorsoft Edge ect. We mostly used the find_element, send_keys, text, click() and clear() functions.

Find_element allows us to locate webpage elements and load them into a Python variable which we can futher use for different kinds of manipulation such as send_keys or .text.

Send_keys allows us to send different kinds of inputs into our browser, it supports the full UTF-8 tedx encoding as well as allowing us to use button combos like Shift + Letter to send a caplital letter or copy paste using Ctrl+c and Ctrl+v.

The .text function allows us to read text from a previously loaded web element. In our testing procedures we used it to read data from tables and compare that to our own variables to check if the table data actually updated after applying changes on the frontend.

Click() allowed us to click on loaded webelements, we mostly used it to navigate to different sections of our webpage (clicking on the report header from the landing page and similar)

Clear() function was handy when updating already existing data as when we click on the Update button the previous data is already filled and presented to us. In order to clear those we loaded in that textfield and sent the Clear() function to it.

OpenPyXl was used to manipulate the excel workbooks. Main functions were loading in a workbook with openpyxl.load_workbook(Name), writing data into cells using WorksheetName[cell] = “data to be written”, formatting functions as fill, alignment and finnaly workbook.save(Name).

4. Modules and Priority Level

In this segment, we will touch on priority levels for bug/issue fixing. The priority level allows the dev team to determine the severity of the issue on hand. Higher priority level puts more important tasks higher on the schedule, resting the important functionality sooner.

The Priority Levels sorted by their severity are:

- Critical [C] – The website is not functional without this feature
- High [H] – The websire is heavily affected without this feature
- Medium [M] – The website is functional but moderately affected
- Low [L] – The website will operate normally without this feature

Here or inside the priovided older you can find the issues and priority levels of the https://puppies-closet.com/evidencija/ webpage.

5. Conclusion

After conducting multiple Load as well as UI, Unit, and Integration tests it is evident that the webpage is not yet ready for Production or being pushed into the live enviroment. Our testing uncovered a wide specter of different issues ranging from basic functionality as showing database reports on taken or free equipment to being able to decomission equipment from employees and printing 'revers'.

Aside from functionality tests we have to keep in mind the logic behind these functions. One illogical vulnerability I have found while testing is being able to delete your admin account while in it. This might create an issue with having anyone to actually create new accounts as there is not register option on the login page. Deleting the only admin account will result in loss of access to the webpage which could be resolved only thorugh the backend.

Another illogical piece is printing the reverse when decommissioning equipment, as the revers should only be printed and signed by both parties when commissioning equipment.

Jmeter stress tests show a viable response time and little latency. There is still room for further optimization, but in the current development state these results should not present an issue.

In conclusion this webpage will need further development on the backend as well as on the frontend. Out of the 47 automated tests 13 tests have fully failed. Further development is required to meet the standards required for a production-ready website.

The list of conducted test, both automated and manual, can be found inside the file named QA Checkbook (Date and time).xlsx within the Automated scripts folder- or by clicking on this link.