

TIE-21201 Ohjelmistojen testaus (Tampere University)
Testing a photo album web application

author: Joonas Ruutinen
latest version: 8.12.2019

1. Introduction

This document describes the contents and usage of the files in test-scripts/ folder and gives an overall explanation of the project that was part of a course “TIE-21201 Ohjelmistojen testaus”.

1.1. Contents of the repository

```
| - - Documentation
|   | - - project-description.pdf
|
| - - Test-scripts
|   | - - Login-test
|     | - - login-test.py
|     | - - test-cases.json
|     | - - test-log.txt
|   | - - Date-search-test
|     | - - date-search-test.py
|     | - - test-cases.json
|     | - - test-log.txt
|   | - - Keyword-search-test
|     | - - keyword-search-test.py
|     | - - test-cases.json
|     | - - test-log.txt
```

1.2. About the course project

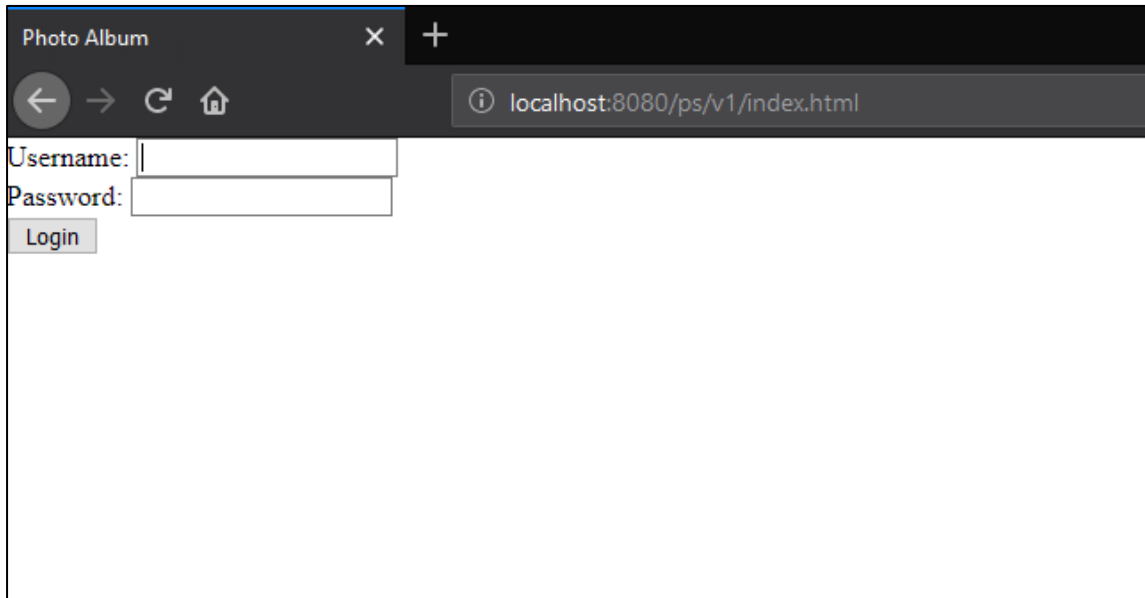
Instructions for the course project was to plan and execute a testing process for a photo album web application. Making a test plan documentation was the first part of the project and second part was to follow the testing plan and perform the tests to the actual application as specified in the document. Testing was to be made as black-box testing (functional testing) without having further knowledge of the source code of the application. Project was done in pairs. Part of our testing plan was to include test-automation in form of test-scripts to test the functionality of the web page. Scripts were written in Python and Helium’s Python library (*) was used for executing actions in browser.

In order to actually test the application, it has to be running in the background or the tests won’t give any actual results. The application is not included in this repository, but its functionality is described with screenshots. Helium’s Python library is also not included in this repository.

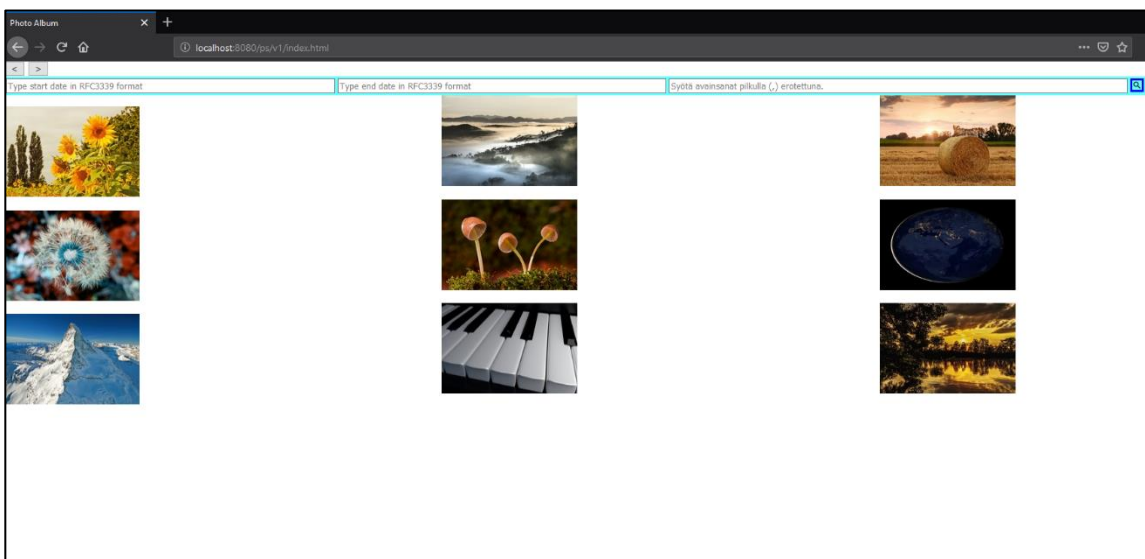
* <https://heliumhq.com/>

1.3. The application to be tested

The target of the testing process was a photo album web application with Login-page, album-view and photo-view. The test-scripts of the repository are testing the login-function and the date-search-function of the album-view.



The login page of the application in Mozilla Firefox



The album view of the application in Mozilla Firefox

2. Test modules

2.1. Login-test

The *login-test* -module performs tests to application's login function by trying out different *username* and *password* combinations in browser and checking if the album-view was loaded on the page after clicking on the *Login-button*. Logging in is allowed only with the correct *username* and *password* combination.

All username and password combinations are loaded from *test-cases.json* file and iterated through in the script. The results are written into *test-log.txt* file. Test log is overwritten every time the test is performed.

2.2. Date-search-test

The *date-search-test* -module performs tests to application's search-by-date function in the album view. Photos can be searched by giving a start and end date parameters in RFC3339 date format. Expected result is that when a search is performed on the web application's album view, page is reloaded and populated with images that match the given specifications. In the test script these images are recognized by taking the 'src-attribute' of a loaded image and separating an ID from the url. Then they are compared to an existing list of images with matching IDs and checking if they were supposed to be loaded in this search.

All date combinations are loaded from *test-cases.json* file and iterated through in the script. The results are written into *test-log.txt* file. Test log is overwritten every time the test is performed.

2.3. Keyword-search-test

The *keyword-search-test* -module performs tests to application's search-with-keywords function in the album view. Photos can be searched by giving one or more keyword parameters separated with comma. Expected result is that when a search is performed on the web application's album view, page is reloaded and populated with images that match the given specifications. By default, when starting the application, there is no keywords added to any images. At the start of the test some keywords are added to certain images and search results are compared to those.

All keyword combinations are loaded from *test-cases.json* file and iterated through in the script. The results are written into *test-log.txt* file. Test log is overwritten every time the test is performed.

More detailed information about the modules and their functionality is documented in the actual script-files (*login-test.py*, *date-search-test.py* and *keyword-search-test.py*). Test log files provide information about the latest performed tests and their results.