**Test plan for Hillel auto web-site**

**Introduction**

Hillel auto web-site is a project, which helps to calculate and keep car maintenance expenses. This resource also provides instructions for different car models to repair a car by your own.

Hillel auto web-site has two types of personal area: user and guest profiles.

**Scope of testing**

1. **Features to be tested**

All features which must be covered by tests are written in the requirements to this project (which by the way also should be tested, the link to requirements is: confidential link.

So the features are:

* Auto List (list of instructions for presented car models)
* Main page of website (it consists of Header, About, Contacts and Footer sections)
* Section with profiles (both user and guest profile)

The process of testing will be held in Chrome browser (ver. 96.0.4664.55) and MacOS (ver.11.6)

1. **Features not to be tested**

* Code at backend of website Hillel auto
* Database

**Test objective**

The objective of testing Hillel auto website is verify that it provides correctly all its functionalities which are in the header, main page and footer. To succeed in this we use manual testing, API testing, Accessibility and Performance testing.

**Test criteria**

1. **Test entry criteria**

* All requirements to a web-site are reviewed and verified
* Test plan is approved
* Test cases are created and checked
* QA environment is prepared for testing
* Needed tools are ready for use (Test Rail, Postman, Lighthouse, WebPageTest)
* All features to be tested are deployed at QA environment
* Enough time and human resources for testing are available

1. **Test exit criteria**

* Test cases are completed
* All types of bugs are found and discussed with a team (or if necessary with a Client)
* Major+ bugs are fixed
* General report about testing is formed
* Product is ready to be deployed at Production environment

**Test methods**

The purpose of testing Hillel auto website is to find bugs and to make sure that in the exit we receive valid and quality product for our Customer and users. To achieve this goal we use several test methods:

* Manual functional testing, which is the most important and time-consuming method of testing at the same time. Due to it we analyze requirements to a project, investigate main sections of the web-site (header, main page, about and footer sections), create and run test cases
* API testing with a usage of Postman. This program allows us to create test scenarios from several requests which can be run in the end. Here we also can use test-scripts, variables to include them into several requests etc.
* Accessibility testing with Lighthouse tool in Google Chrome helps in checking and analysis of web-pages.
* Performance testing with web-tool WebPageTest shows and estimates page loading at all stages

**Test tools**

All process of testing includes usage of several tools which are presented in the table below:

|  |  |
| --- | --- |
| Requirements | Confidential link |
| Manual functional testing | Jira, Test Rail |
| API testing | Postman and API documentation at link:  confidential link |
| Accessibility testing | Lighthouse (an extension for Google Chrome) |
| Performance testing | WebPageTest at site https://www.webpagetest.org/ |

**Testing schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Activity** | **Begin date** | **End date** | **Work duration** |
| 1 | Requirements reviewing and creating bugs connected with them in Jira |  |  |  |
| 2 | Test plan creating |  |  |  |
| 3 | Writing test cases and run them in Test rail+creating bugs in Jira |  |  |  |
| 4 | API testing |  |  |  |
| 5 | Accessibility+Performance testing |  |  |  |
| 6 | Report about testing |  |  |  |
| 7 | Recheck bugs using correct version of website |  |  |  |