

Code Challenge

Front-end Test Automation Engineer

Overview

This is a code challenge for prospective Frontend Test Automation Engineers in Test.

It is expected that all points are addressed.

This challenge covers:

- Development of a small test automation framework using Cypress
- Using javascript/typescript
- Taking a Behaviour Development Driven (BDD) approach

Deliverable

You are expected to develop a test automation framework aimed at testing websites. The framework should include feature files and step definition classes. The framework could also include any other layer that you deem appropriate such as page object models and configuration files. Commit (regularly) to your git repository.

The project should also include documentation explaining:

- Any taken assumptions
- Any difficulties that were encountered
- A README.md file with instructions on how to install, build and run the tests. *The*

final deliverable should be a link to your git repository.

Expected Goals

- We keep an eye out for the design of the test automation framework so think about the extendibility of your solution and the maintainability of your tests.
- The code is expected to compile and automated tests are expected to pass on any machine with the latest versions of node and npm installed.

Write automated tests for the following features against the website
<https://www.ninjacasino.com/>.

Test Cases

Game

A (logged out) user should be able to find and open the game.

Preconditions

- <https://www.ninjacasino.com> is reached from a desktop device;

Steps

1. navigate to <https://www.ninjacasino.com>;
 - search input field is shown;
2. search for “book of dead”;
 - games matching with ”book of dead” show up;
3. click on “play” button;
 - verify that the game is loading;

Language Switching

A (logged out) user should be able to switch language by the mean of the menu dropdown.

Preconditions

- <https://www.ninjacasino.com> is reached from a desktop device;

Steps

1. navigate to <https://www.ninjacasino.com>;
 - language dropdown is shown in the header;
2. click on the language dropdown;
 - the language menu shows up;
 - click on FI reload the application in Finnish;
3. repeat 2. with EE
4. repeat 2. with RU

Main Site Navigation

A (logged out) user should be able to browse NinjaCasino by the mean of the hamburger menu.

Preconditions

- <https://www.ninjacasino.com> is reached from a mobile device;

Steps

1. navigate to <https://www.ninjacasino.com>;
2. the menu opens upon tap on the burger icon;
 1. the following entries are found Home, Casino, Live Casino, Sport Betting, Live Betting, Ninja Treasures, Promotions and Customer Support;
 - tap on Ninja Treasures redirects to *Ninja Treasures* page;
 - tap on Promotions redirects to *Promotions* page;
 - tap on Customer Support redirects to *Customer Support* page;
3. the game categories sub-menu opens tapping on Casino;
 1. the following categories are found Popular, New, Slots, Live Casino, Jackpots, Table Games, Megaways;
 - tap on Popular redirects to *Popular* page;
 - tap on New redirects to *New* page;
 - tap on Slots redirects to *Slots* page;
 - tap on Live Casino redirects to *Live Casino* page;
 - tap on Jackpots redirects to *Jackpots* page;
 - tap on Table Games redirects to *Table Games* page;
 - tap on Megaways redirects to *Megaways* page;

Create tests to cover the above user journey.