JavaScript Assessment

Your assessment is to replicate the functionality of a simple website.

There will be four webpages you will need to create:

* [Index Page](#_Index_Page)
* [Trainer Page](#_Trainer_Page)
* [Entry Form Page](#_Form_Submission_Page)
* [Form Data Page](#_Form_Information_Page)

If you get stuck on a section it is advised that you move on.

Each page you build is separated into tasks to help you.

The assessment must be uploaded to your GitHub accounts by the end of the Assessment Deadline with the repository name ‘JavaScript Assessment’.

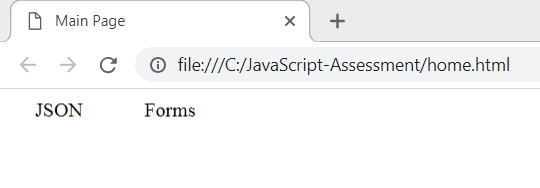
After this is submitted send an email to [Matthew.Hunt@qa.com](mailto:Matthew.Hunt@qa.com) with your GitHub username.

# Index Page

## Task 1

* Have the title of the webpage to be: “Main Page”
* Add two entities side by side and give them unique ID’s.

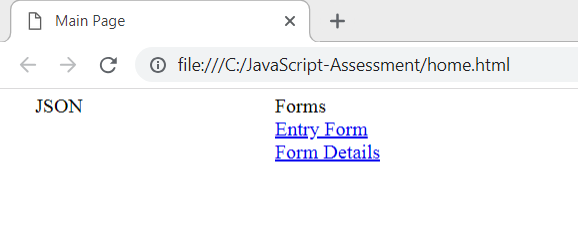
Example:



## Task 2

* Create dropdown menu when you hover over an entity. This should be done with JavaScript and **NOT** with the CSS ‘.entity :onhover {}’ functionality:
  + When you hover over *JSON Assessment* it should reveal:
    - JSON manipulation
  + When you hover over Form Assessment it should reveal:
    - Entry Form
    - Form Data
* The dropdown menu should remain visible when your mouse hovers over an item within the dropdown menu.
  + For example, when you hover over *Form Assessment* and then move your mouse to *Entry Form* the menu shouldn’t disappear.
* Dropdown Item functions:
  + When you click *JSON manipulation* it should take you to trainers*.html*
  + When you click *Entry Form* it should take you to *entryform.html*
  + When you click *Form Details* it should take you to *formdata.html*

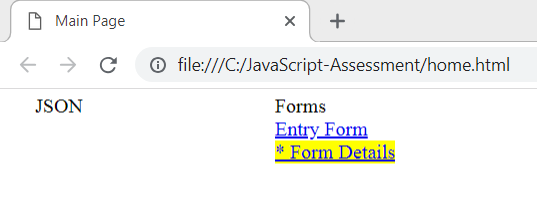
Example:



## Task 3

* When you hover over one of the dropdown menu items, the menu Item should:
  + Add an asterisk by it:
  + Change the background colour
* When your mouse is no longer hovering over the item, it should revert back to normal.

Example:

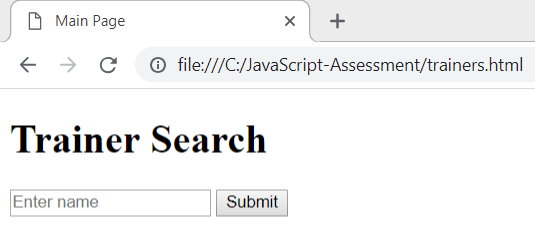


# Trainer Page

## Task 1

* Add an input field into the webpage which has a placeholder of the “Enter name”
* Add a button that when the name is entered will submit the search

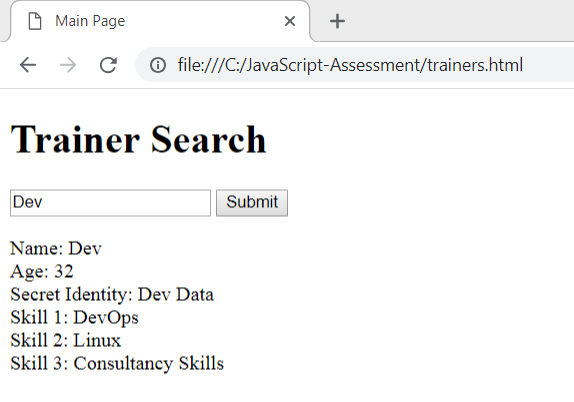
Example:



## Task 2

* From the name entered in the input, have it search through the data in the JavaScript file provided and display relevant information.
* The data should be overwritten each time someone new is searched for
* If the search doesn’t return someone’s name, the information should disappear

Example:



## Task 3

**Display the results in a neat table**

## Task 4

**Create another search field that allows a user to search for the trainers that have a particular skill, for example ‘Java’. The page should then clearly display the names of the trainers with this skill.**

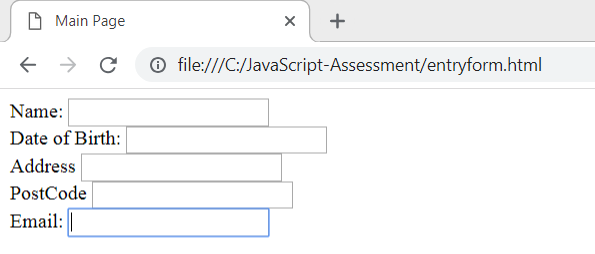
## Entry Form Page

Create a submission page with various fields. Once the input is submitted, it will need to be loaded into the [form data page](#_Form_Information_Page).

### Task 1

* Create a form with input for each of the following fields:
  + Name
  + Date of Birth
  + Address
  + Post Code
  + Email

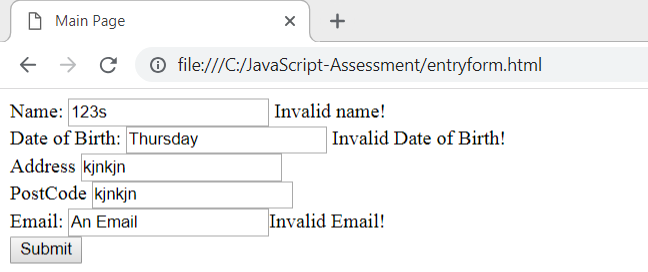
Example:



### Task 2

* Add validation to each input field
  + Name should not have numbers or special characters
  + Date of birth should be formatted in a DD/MM/YYYY
  + Address must not contain special characters
  + Post code must not be longer than 10 characters and not contain special characters
  + Email must contain and @ followed by a .

Example:



### Task 3

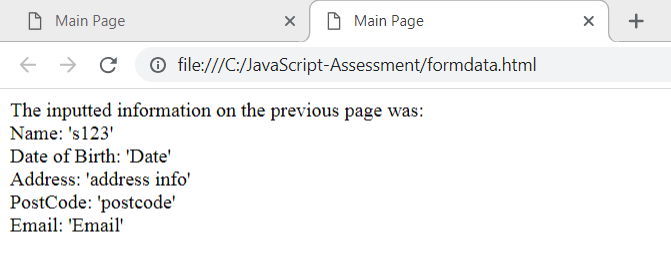
* The submit button should not work until all fields are within specification

## Form Data Page

### Task 1

Retrieve the details from the submitted form and display it on the webpage

Example:



### To achieve this, use:

*localStorage.setItem("key",value);*

To store the value on the form page.

and

*(localStorage.getItem("key"))*

To retrieve the value on the formdata page