CONTENTS

JavaScript Fundamentals Hackathon part 1

Overview

In this part Hackathon, you will build on a partially developed solution (whether that be your previous iteration, or the provided starting point) for QA Cinemas' website, by adding validation to the form. All the necessary tools, knowledge and techniques have been covered in the course so far.

Objectives

This part of the Hackathon is intended to help you develop your JavaScript skills and knowledge, to be able to validate a 'Sign-Up' form for users of the QA Cinemas website. The form will be validated before it's sent to be held on the server.

User story

As a User, I want to be able to see an immediate validation of my input into the sign-up form, so that I know the information I enter is acceptable.

Acceptance criteria

The following table gives information about the validation that should be implemented on form fields. Supplementary information is provided under the table:

FORM FIELD	VALIDATION	FURTHER INFORMATION
TITLE	Required	
FIRST NAME	Required	Only lower and upper case letters are allowed. Otherwise, the "Only lower and upper case letters are allowed" message will be displayed below the First Name field in red and the field itself will have a red border. Only names longer than 1 character and shorter than 15 characters are allowed. Otherwise, a message reading "The First Name entered is either too short or too long." will be displayed below the First Name field and the field will have red border.
LAST NAME	Required	As above but applied to Last Name
EMAIL	Required	Every valid email address follows this pattern: at least one character, '@' sign, one or more characters, '.' character and one or more characters. In case an invalid email address is entered, a message reading "Email address not valid" will be displayed below the field in red and the field's border will also change to red.

- If information entered into a required field is valid, the field's border will remain unchanged and no additional message will be displayed.
- The validation is performed immediately after the user finishes typing into a field and the cursor is changed into another field.

- Other fields on the form do not require validation.

Coding style and standards

- Code should be written using the latest JavaScript specification, but production code should transpiled to target ES5.
- The use of Object-Oriented patterns is preferred.
- The use of modules is preferred.

JavaScript Fundamentals Hackathon part 2

Overview

In this part Hackathon, you will build on a partially developed solution (whether that be your previous iteration or the provided starting point), for QA Cinemas' website, by allowing submission of the user data from the form to a remote backend. This should be simulated by using json-server. All the necessary tools, knowledge and techniques have been covered in the course so far.

Objectives

This part of the Hackathon is intended to help you develop your skills and knowledge, so you can use JavaScript to submit data from a 'Sign-Up' form, for users of the QA Cinemas website.

User story

As a User, I want to be able to submit my data and receive confirmation that it has successfully sent, so that I know that the information I submitted has been received.

Acceptance criteria

- The form should check that the first name, last name and email are valid before preparing the data to send.
- The data should be sent to http://localhost:3000/users as a POST request in the form of a JSON string.
- The request must have cors and a Content-Type of application/json.
- The system should confirm that the data has been successfully received or indicate that something went wrong.

Coding Style and Standards

- Code should be written using the latest JavaScript specification, but production code should transpiled to target ES5.
- The use of Object-Oriented patterns is preferred.
- The use of modules is preferred.