

# Front-end Exercise

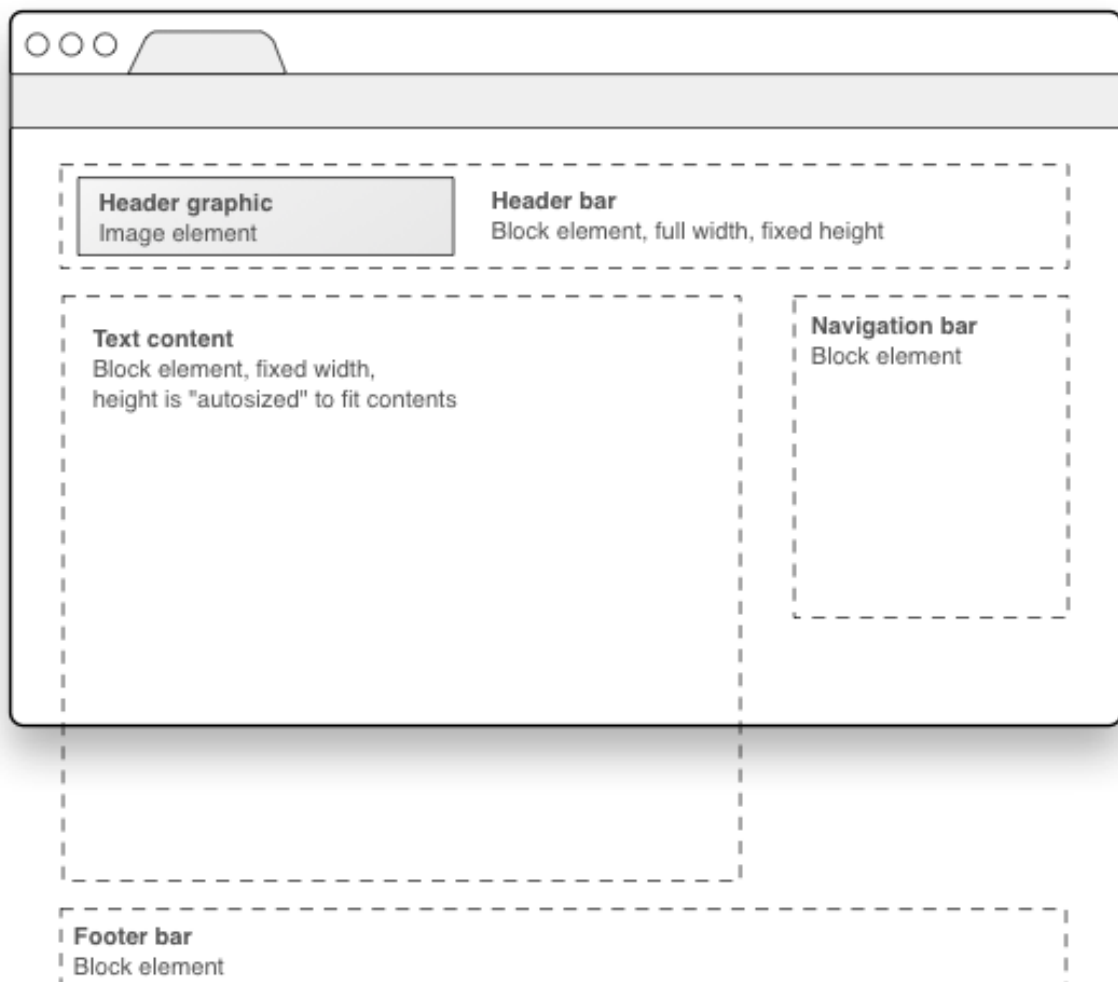
## Introduction

The purpose of this exercise is for developers to exhibit familiarity with front-end technologies, various implementation techniques, to learn and to have some fun 😊

## Exercise

Please create a small application that will demonstrate basic knowledge of fundamental Front-end implementation techniques and best practices.

The application should have proper HTML5 build blocks that at least should include usage of semantically correct elements like header, footer, main, article, menu, side, small, section, nav, h1 – h6 and so on. The basic structure can look something similar like the image below:



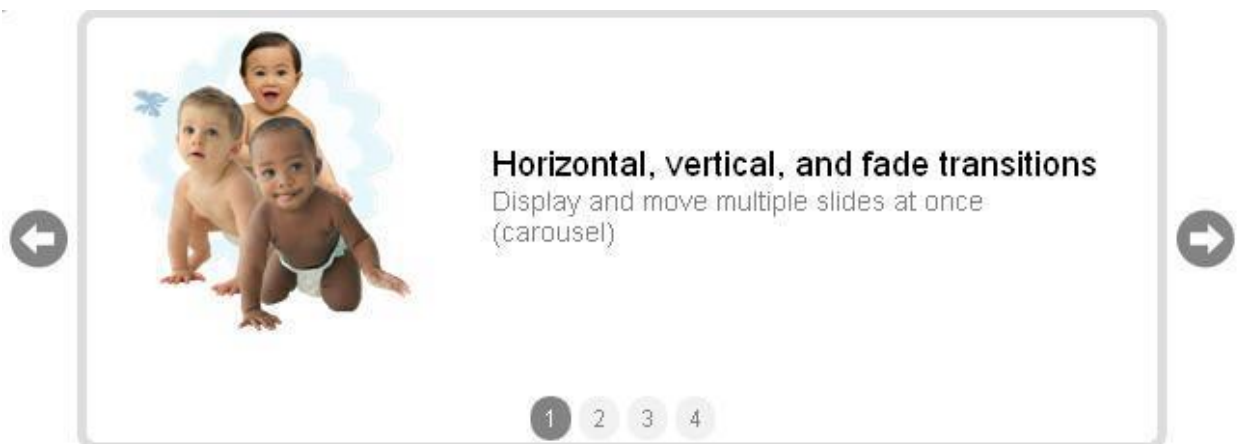
You are free to put any visual and markup content you find fit. Also for reference of all official HTML5 elements per W3C specification, please use this link <https://developer.mozilla.org/en/docs/Web/HTML/Element>.

For styling it isn't necessary to do some magic or create special design. Only utilize Bootstrap 3 grid system with default break points and predefined styles declaration for styling all elements and layout on the page. What would be required is that entire application should be full responsive or adaptive. Or even in combination of responsive / adaptive synergy.

We would only require from you to style form submit button element as per image below, utilizing only CSS methods without images, any graphical elements and without changing HTML markup. Only use one single HTML element for this (a, button, or input type button) without any children elements or maybe targeted wrapper:



As part of the main content first we would like you to create simple content slider using jQuery or maybe native JavaScript ES5 / ES6 without utilizing 3<sup>rd</sup> party plugins or jQuery UI library. It should resemble on structure level somehow like the image below:



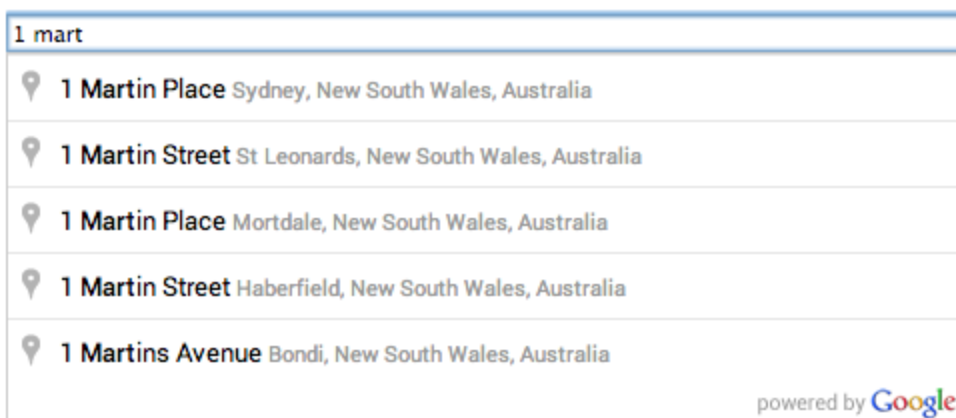
For the second part of the main content, we would like you to create a multi-step form with as many elements as you like. Try to use as many different forms elements as possible and in different way. For reference please lookup elements at <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Forms> and <http://nativeformelements.com/> resources.

This form also need to be functional in a way that it needs to have complex validation, various notifications / messages, and error handling. For instance, some validation pattern that form should have are minimum / maximum length, required fields, fields that are input type number should have validation that user can only enter numbers, for text it should be that user can enter only numbers and lower / upper case characters and so on. Basically, implement as many validation patterns as you want.

For validation messages please consult best practices on following resource URL

<https://medium.com/@andrew.burton/form-validation-best-practices-8e3bec7d0549#.2zpw2arav>.

Beside validation, one of the field should be an “autocomplete” field, where you should fire AJAX request to retrieve list of predefined items options as user type characters in that field. Please try to create this functionality from scratch, without using any jQuery 3<sup>rd</sup> party plugins or UI library. For simple example, please check live implementation here <http://easyautocomplete.com/examples> and small screenshot below:



The screenshot shows a search bar with the text "1 mart" entered. Below the search bar, a list of five suggestions is displayed, each preceded by a location pin icon. The suggestions are: "1 Martin Place Sydney, New South Wales, Australia", "1 Martin Street St Leonards, New South Wales, Australia", "1 Martin Place Mortdale, New South Wales, Australia", "1 Martin Street Haberfield, New South Wales, Australia", and "1 Martins Avenue Bondi, New South Wales, Australia". At the bottom right of the suggestions list, it says "powered by Google".

Suggestion
1 Martin Place Sydney, New South Wales, Australia
1 Martin Street St Leonards, New South Wales, Australia
1 Martin Place Mortdale, New South Wales, Australia
1 Martin Street Haberfield, New South Wales, Australia
1 Martins Avenue Bondi, New South Wales, Australia

When switching between form steps please fire AJAX request to retrieve new step, do some DOM manipulation and apply events to newly retrieved markup (validation, click / touch events and so on). On last step simulate form submit events in a form of JSON respond from BE. Try to shows us some advanced DOM and events handling and manipulations. Also, try to use jQuery build in animation transition between form steps if possible.

For visual representation please utilize Bootstrap 3 predefined markup and styles for elements control as per simple example below:

EXAMPLE

Email address

Email

Password

Password

File input

Choose File No file chosen

Example block-level help text here.

☐ Check me out

Submit

## Browsers support

Your solution should work on all latest build of modern browsers. That include Chrome, Firefox, IE 11, Edge, Safari on Mac, Safari on IOS mobile devices, Opera, and Android native browser.

## Architecture

Please structure your solution in that way that everything is maintainable and if possible show us some of the best practices of how that should be done. Organize static resources in that way, that someone who doesn't know anything about the project could easily find the way around it. One of the most important thing for us is to see a readable arrangement of and relations between the parts or elements of complex code structure. In other words, please write meaningful comments, JSDoc and methods declarations, provide proper naming convention for functions, utilities, methods, folders, and files.

Also, try to think about performance impact on client side and how to optimize your implementation for best small footprint. Where should all static resources and in which way be included on page, what is the most expansive call during page rendering and so on.

## Back-end

No back-end implementation is required for this exercise. Everything should be done on client side and should be visible and functional just by opening index.html in any browser.

## Bonus

These additional implementations aren't obligatory in any way and will just provide extra bonus and a look at how you think and what approach you'll take:

- Modular approach – Separate logic into different and physically independent modules
- CSS pre-processors – Use LESS or SASS to compile CSS
- Build tools NPM, Gulp / Grunt – Use Node.js to utilize npm modules and create various tasks using some build tools
- User Bower for resources package management
- Performance – Pay attention to jQuery performance penalties and optimizations <https://learn.jquery.com/performance/>
- QUnit tests – Create unit tests using QUnit framework
- JSHint (<http://jshint.com/>) / JSLint (<http://www.jshint.com/>) – Check your JavaScript implementation versus popular JS linting tools and fix issues
- W3C validator – Check your markup and CSS style versus W3C validation service <https://validator.w3.org/> and <https://jigsaw.w3.org/css-validator/>
- Create content slider without jQuery or JavaScript influence and with only CSS3
- Cache functionality – Utilize HTML5 API storage cache to reduce HTTP requests
- Use data attributes or DOM elements data objects
- Create README file in root of your application with detail explanation of decisions and implementation you have taken (markdown can be freely used)