JavaScript Developer Test

Instructions:

Unzip the Javascript Project - js-test.zip, and deploy it to a basic http server on your machine and complete the tasks outlined below. This project is based on the backbone.js framework. Backbone allows front end web apps to be developed using an MVC pattern, you are expected to stick to this pattern when completing the tasks below. A key assessment area will be proper use of object-oriented coding practises.

The test should take approximately 1 hour to complete. You will be assessed on you quality of code, code structure and consideration of validation, error handling and consideration of user experience. You are allowed to use internet resources for reference if required but you will be marked down for 'copy/paste' code.

Feel free to use any open source libraries or frameworks to help you complete the tasks below. If you do use any additional frameworks, please specify these when you return the project.

Navigate to the index.html via a web browser and you will see that 4 buttons are loaded initially…

Task 1 – Show Message

When the user presses 'Show Message' the action results in a JavaScript runtime error, investigate the problem and fix any / all syntax errors in the code, so that the ShowMessageView displays a text box which allows the user to display a custom message on screen.

Task 2 - Calculate Number

a) When the user presses 'Calculate Number' the action results in a JavaScript runtime error, investigate the problem and fix any / all syntax errors in the code, so that the CalculateView displays a text box that allows the user to enter a number which should then be used to divide a preset value.

b) Currently, the result shown after calculation is incorrect. Correct the error in the source code, so that the correct value is calculated.

c) Add validation to the code, so that the user can only enter a numeric value to be divided that user is only allowed to submit value that result in real number results.

Task 3 – Enumeration

Implement the 'nextSevenDays' function in views/daysOfWeek.js – The function should enumerate through the array of days (daysOfTheWeek), **starting with the current day**, build and return a list of the next 7 days to be displayed on screen when the user presses the 'Next 7 days' button. It is important that this function be as fast as possible (prioritise speed over memory usage).

Task 4 – Webservice Integration

a) Show Stores

Red Ant Digital Ltd

The following URL format, is a webservice that responds to GET request with the correct parameter.

http://cloudservices.arcadiagroup.co.uk/storestock/storestock?brand=12556&jsonp_callback=results&lat=*{latitude}*&long={longitude}&dist={distance}&res={results}

{latitude} is the latitude to search from eg. 51.511

{longitude} is the longitude to search from eg. -0.1198

{distance} is the search radius in kilometres eg. 50

{results} is the number of results to return in the search eg. 10

The JSON response is a list of nearby stores for a major high street retailer. Implement the following functionality, into the project.

The base URL for the above webservice endpoint, is currently defined in a global 'config' object as config.webserviceURL (see app/main.js) this should be used when specifying the endpoint in code.

NB: The response from this webservice is surrounded by a callback wrapper of the format results(*jsonData*); which will need to be removed before the data can be parsed correctly.

When the user presses the 'Show Stores' button, a text box should appear allowing them to enter a latitude & longitude, and run the search. This webservice should then be called a summary of the top 5 results should be shown in a table in a results area below. The name, fully formed address and telephone number of each store should be shown. Consider loading times, invalid server responses, timeouts, server errors and no results being returned.

b) Location

Alter project, so that when the user initially, presses 'Store Stores' the device's current location is used, to start an initial search, which can then be changed by the user via the text box. Consider, location information being unavailable.

c) Scope

The current implementation of 'window.config.webserviceURL' in main.js is poor practise. Alter the implementation to show who you would set up global configuration variables and prevent potential conflicts with other frameworks.

If you have time, additional merit will be awarded to candidates who can demonstrate creative flair or functionality to improve the user experience of task 3.