Tests for Software Engineers

You can consult documentation, stackoverflow and other external resources. You are NOT supposed to ask for anybody's help either directly or through a discussion board.

# HTML and JavaScript

Here is a JSON containing data about crypto currencies.

let json = [

{ id:1, ticker:"BTC", name:"Bitcoin", createdOn: "2008-01-03", chart:

[

{date:"2018-01-01", price:12556},

{date:"2018-01-02", price:15001},

{date:"2018-01-03", price:16300},

]

},

{ id:2, ticker:"ETH", name:"Ethereum", createdOn: "2012-01-23", chart:

[

{date:"2018-01-01", price:657},

{date:"2018-01-02", price:732},

{date:"2018-01-03", price:590},

]

},

{ id:3, ticker:"DASH", name:"Dash", createdOn: "2014-01-14", chart:

[

{date:"2018-01-01", price:1501},

{date:"2018-01-02", price:1634},

{date:"2018-01-03", price:1490},

]

},

]

For each task, create a separate working HTML page. Name them task1.html, task2.html .... taskN.html. Feel free to combine multiple tasks into a single page. If your page refers to external resources e.g. Bootstrap or Moment.js or JQuery, provide these resources together with the pages so that the examiner doesn't need an Internet connection. HTML should be Chrome-compatible. You can use any JS or CSS library e.g. Angular, Moment, React, Bootstap, etc.

Task 1. Create a page, which lists all cryptocurrencies in a form of a table (use any HTML tags as long as it displays as a table).

The table should contain the following columns: id, ticker, name.

Task 2. Decorate the page from task #1 with CSS via using Bootstrap or by developing your own styles.

Task 3. Add a column "Created On" to the table. Use moment.js to display how long ago each currency was created (e.g. "3 days ago" or "10 years ago").

Task 4. Split the page into left and right parts. The left part should contain the list of currencies including ticker and name.

When user clicks on a list item, the right hand side should display id, ticker, name, and creation date of a selected currency.

Task 5. Test #4 plus add a price chart to the right-hand side. Use any charting library or do it via SVG.

Task 6. Test #4 plus implement ctyptocurrency search on the left hand side. User starts typing a currency name or ticker, and the page displays matching results.

Task 7. Task #4 plus the page should preserve the selection after user clicks "Refresh" in the browser.

Task 8. Task #4 plus once a user selects a cryptocurrency, the page should change the URL. User should be able to copy and paste the URL into a different window, and the page should display the selection.

Task 9. Task #4 plus make the page responsive. On the cell phone the right hand side should display at the bottom.

# C#

Here is an API that provides cryptocurrency-related information.

https://min-api.cryptocompare.com/data/pricemulti?fsyms=BTC,ETH,DASH&tsyms=USD

This API will be used in all C#-related tasks.

For each task, create a separate console executable and provide the source code.

Note that the examiner will NOT have Visual Studio to compile your projects.

Task 1. Fetch today's price for Bitcoin (BTC) and display it in a standard output.

Task 2. Task #1 plus ask a user to input a cryptocurrency ticker. Call the API using this ticker as an input. If the API returns an error, display it gracefully.

Task 3. Call the API and fetch prices for BTC, ETH, DASH. Display them in standard output.

Task 4. Instead of standard output, save price information as a JSON file on a desktop.

Task 5. Instead of standard output, save price information as an HTML file on a desktop. Make sure special characters are properly encoded.