**LIVE DASHBOARD TEST**

**Web Development**

A start-up stock broker company has the need to display the latest stock indices at certain key points in their office building.

They have decided to use raspberry pi devices connected to television screens. The raspberry pi will use Chromium that connects to the website <https://www.sharenet.co.za/v3/indices.php>

The data on the screen needs to be refreshed every 5 minutes.

If for some reason the raspberry pi loses connectivity the web application still needs to show a cached version of the data.

An indicator on the screen then needs to display that this is not the latest data due to connectivity loss.

As soon as connectivity has been restored the web application needs to refresh the page to indicate the latest data. The on-screen indicator should then be removed.

You were tasked with creating the above solution for the client. An MVP needs to be created within 4 days. You can use any language to build the application.

Also consider corner cases during your development.

**Web Application Requirements**

1. **API** 
   1. To send the latest stock indices RESPONSES to Web App
   2. To display SERVER-SIDE cached stocked indices during loss of connectivity
2. **FRONT-END APP** 
   1. To display the latest stock indices
   2. To auto refresh stock indices
   3. To display/hide an on-screen indictor depending on connectivity to the internet & API

**Languages / Frameworks / Libraries**

|  |  |  |  |
| --- | --- | --- | --- |
| **Options** | **API** | **FRONT-END APP** | **Selected option** |
| 1. | ASP.NET 4.8 or Core | 1. ASP.NET MVC C# Web App 2. Html + CSS & libraries + JavaScript & libraries | **✔** |
| 2. | PHP script or framework | 1. Angular 2. React | ✘ |

**Corner Case**

1. Send API stock indices in what format? E.g., **JSON object**
2. Receive API call in the **front** or back end? E.g., **JavaScript** (**fetch Api,** Axios, JQuery or XMLHttpRequest**)**
3. Should API stock indices be called when JSE markets are Closed on outside work hours.
4. Each time a request is sent to fetch the stock values, A promise that the request will be fulfilled is made. Therefore, use **async await** to cater for different response times and use a **catch** for other errors.
5. **Caching** increases performance and response time. AWS allows for API response caching when using its environment. Note the cache limit.