Project Summary

Test Assignment

By: Daniel Derhami

26th of August 2016

Abstract

The aim of this project is to create a simple web application, based on the scenario "*Test assignment*", requested by "Globalmouth". [1]
The final codes are available within the github repository: https://github.com/shahriar-derhami/TestAssignment.

Contents

1- Introduction

The "Sports Poll" app is a dead simple poll system around some sports events which are accessible from a JSON file. These sport events divided into five categories as Football, Tennis, Snooker, Ice Hockey and Handball. The "Sports Poll" app should provide random event to user in order to let him/her to decide and guess the result. These showing events are completely random and they based on the sport category.

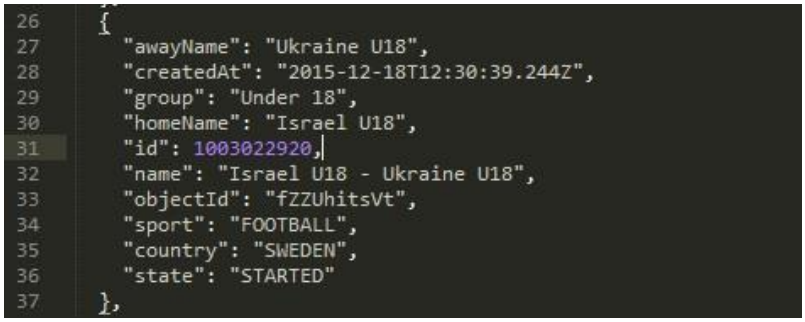

Figure 1 - Events data in JSON file

The way of voting to each event by user is simply based on choosing between three options: Home Team win (Team A), Draw and Away Team win (Team B).

2- Our Approach

2-1 Stakeholder(s)

Regarding the scenario we have ahead, we could have two stakeholders. With a simple definition we can call these two stakeholders "*User*" and "*System*". While this app is a dead poll system and there is no actual connection with a system behind and all the data are available only by JSON file, we can refer our frontend system as the "*System*" in our scenario.

2-1-1 User

User has the control to only choose his/her desired option among the presented ones by the system. *User* can only choose one option.

User has no need to keep track of the choices he/she made.

2-1-2 System

As it mentioned before, *System* would be the frontend side of our system. More detailed it is the codes and functions exist in the codes and the main duty of *System* is to represent the random event and choosing options to *User*.

2-2 Development Tools

Html5 and Css3 used as programing languages for design and modify the pages. JavaScript used for handling our functionalities.
Sublime Text 3 used as an editor for writing down the codes. [2]
C9 is used for creating an online repository and then push the codes to GitHub. [3][4]

2-3 Html

The Html files created for this task are “index.html” and “welcome.html”.

2-3-1 Welcome.html

This file works as a welcome page for user and it should be run first. Welcome page is greeting user and by pushing a button it leads her/him to the next html file, index.html.

Figure 2 - View of Welcome page

The body tag contains one main div. This div includes a button and a script tag which leads user to the next html page.
Main div has an animation background which transitions between five pictures as the five sport categories we have.

2-3-2 Inddex.html

The index file contains our JSON file as an array name *var* in one of the two script tags in body tag.
The second script tag includes *myFunction().* This function start automatically (window.onload = myFunction;) and presents the random events to user.
The only div tag exists in index.html file is for show the outcome results to User and let him/her to vote.

Figure 3 – index.html view

2-4 Css

The Css file contains the style definition of our div section for showing the results and –webkit- functions for showing the animation background.
The same way is happening for the only div section in our welcome page.
The coding of Css file was mostly focused on animation section and tried to present the best outcome in to the page.

```
#demo {
    padding-top: 15px;
    border:3px solid black;
    width: 500px;
    height: 200px;
    display: block;
    position: absolute;
    top:0;
    bottom: 0;
    left: 0;
    right: 0;
    margin: auto;
    -webkit-animation-name: example;
    -webkit-animation-duration: 5s;
    animation-name: example;
    animation-duration: 40s;
    animation-iteration-count: infinite;
}
@-webkit-keyframes example {
    0%   {background: url(images/L6.jpg);}
    9%   {background: url(images/L4.jpg);}
    18%  {background: url(images/L5.jpg);}
    27%  {background: url(images/L7.jpg);}
    36%  {background: url(images/L8.jpg);}
    45%  {background: url(images/L9.jpg);}
    54%  {background: url(images/L10.jpg);}
    63%  {background: url(images/L11.jpg);}
    72%  {background: url(images/L12.jpg);}
    81%  {background: url(images/L13.jpg);}
    90%  {background: url(images/L14.jpg);}
    100% {background: url(images/L15.jpg);}
    }
```

Figure 4 - Css code for index.html div section

The only problem happened to this section was due to changing some ids and classes in main codes, keep tracking of the correct Css method was faced to some unexpected problems.

2-5 JavaScript

As mentioned before, Java script methods and functions are used for handling the main part of this app.
In index.html file, the second script tag (*myFunction()*) has the duty to first pick a random index of JSON array and then check it through six ”if” statements. Then the correct condition

tries to pick up the necessary information from the related index and prepare it to show in div section.

```
251     if (info[x].sport == "TENNIS" && info[x].state == "STARTED") { //check tennis games which started
        already
252     document.getElementById("demo").innerHTML = " OnGoing TENNIS Match "+
253   "<br>" +
254     [info[x].homeName +" <== Meets ==> "+ info[x].awayName]
255     + "<br>" + "<br>"
256     +'<button class="button1" type="button" onclick="myFunction()">Home Win</button>'+
257      '<button class="button2" type="button" onclick="myFunction()">Draw</button>'+
258      '<button class="button3" type="button" onclick="myFunction()">Away Win</button>';
259
260 }
```

Figur 5 - If condition

Using more than one makes it possible to check for state of the event to. It can help to avoid presenting the games which are finished already!
The main problem while coding these parts happened as ”Syntax Error”. Missing one + caused code break and it needed to be checked again. In each line!!

2-6 TimeLine

The process of coding of this assignment started at 15:00, 26th of August, 2016. Two parallel scenarios went forward as a backup plan in case of any unexpected problem or code fail.
The actual finish time of coding was at 22:00 o’clock and it is going to send after finishing this document.
More Css design could be added but they are just ignored due to respect to the estimate time mentioned on the task description.

3- Refrences

[1] “Globalmouth - Test assignment, Q3 2016”, Globalmouth. [Online]. Available: https://docs.google.com/document/d/1e2Y9L-HPe6oE9JZbXZNMT6l_GHom-lphnKh0DXwSILI/pub [Accessed: 26 of August 2016]

[2] Sublime Text. [Online]. Available: https://www.sublimetext.com/ [Accessed: 26 of August 2016]

[3] Cloud 9. [Online IDE]. Available: https://c9.io [Accessed: 26 of August 2016]

[4] GitHub. [Online Repository]. Available: https://www.sublimetext.com/ [Accessed: 26 of August 2016]