Yitian Wang

Yitian.wang@agf.com

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

ELEMENTS QUIZ

Documentation

**Table of Contents**

[**Introduction** ii](#_Toc501438977)

[**HTML** 1](#_Toc501438978)

[**Included Scripts and Link** 1](#_Toc501438979)

[**Create the Structure of the Quiz** 1](#_Toc501438980)

[**Report Form** 1](#_Toc501438981)

[**JavaScript** 4](#_Toc501438982)

[**Load JSON-encoded Data from The Server** 5](#_Toc501438983)

[**Global** **Variables** 5](#_Toc501438984)

[**Slides Creation** 6](#_Toc501438985)

[**Function for Slide Content Creation** 6](#_Toc501438986)

[**Cover Slide** 7](#_Toc501438987)

[**Section Slide** 11](#_Toc501438988)

[**Question Slide** 12](#_Toc501438989)

[**Results Slide** 15](#_Toc501438990)

# **Introduction**

The main features for the quiz:

* Question/answer input & backend calculations
* Results page with matched portfolio and pie chart visuals
* PDF download of report with inputs for name + other details

All questions and answers are stored in JSON file. In the JS file, there is a list (array) for storing inputs getting from user. The array will be used for backend calculation then generates recommend investment portfolio.

Both the pie chart and report downloading will be explained later.

The main structure of the quiz is a queue of several slides, linked as a list. Currently, there are four types of slides: cover slide, section slide, question slide and results slide.

To create a new slide, the type of the slide should be included at first in JSON file. The content must be organized according to existing slides with the same type.

# **HTML**

## **Included Scripts and Link**

<script src=**"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"**></script>

<script src=**"https://code.jquery.com/jquery-1.12.4.min.js"**></script>

<script src=**"https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.6.0/Chart.bundle.js**</script>

<script type=**"text/javascript"** src=**"function.js"**></script>

<script type=**"text/javascript"** src=**"data.json"**></script>

<link rel=**"stylesheet"** type=**"text/css"** href=**"style.css"**>

## **Create the Structure of the Quiz**

<img id=**"banner"** src=**"banner.png"**>

<div id=**"frame"**>

<ul id=**"queue"**></ul>

</div>

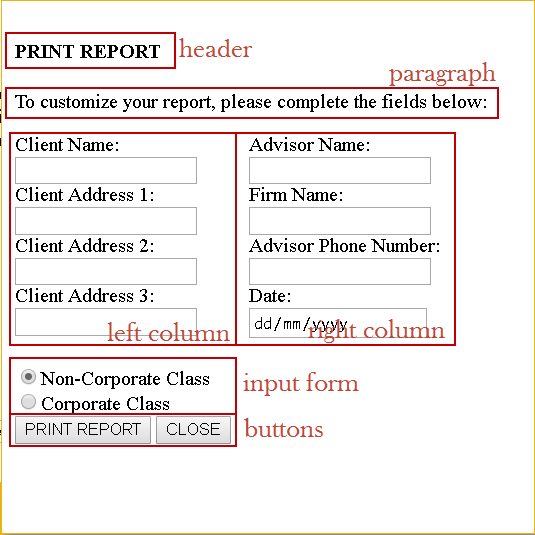
<div id=**"footer"**></div>

## **Report Form**

The report form is not created dynamically (getting data from JSON and creating elements in JS).

The report form takes in basic information from the user, then use that information for full report generation. The display of this window is triggered by clicking the REPORT button located at footer. The REPORT button appears when a portfolio is selected.

* The header and introduction paragraph can be updated through HTML file
* Text fields take users inputs
* Two options of Class
* PRINT REPORT button opens the tab of full report for preview



<!-- create pop-up window for report form -->

<div id=**"report-window"** style=**"display: none"**>

<div id=**"report-title"**>

<h4> **PRINT REPORT**</h4>

<p> **To customize your report, please complete the fields below:**</p>

</div>

<div class=**"report-row"** >

<div class=**"report-column"**>

**Client Name:** <br>

<input id=**"client-name"** type=**"text"**><br>

**Client Address 1:** <br>

<input id=**"address1"** type=**"text"**><br>

**Client Address 2:** <br>

<input id=**"address2"** type=**"text"**><br>

**Client Address 3:** <br>

<input id=**"address3"** type=**"text"**><br>

</div>

<div class=**"report-column"**>

**Advisor Name:** <br>

<input id=**"advisor-name"** type=**"text"**><br>

**Firm Name:** <br>

<input id=**"firm-name"** type=**"text"**><br>

**Advisor Phone Number:** <br>

<input id=**"phone"** type=**"text"**><br>

**Date:** <br>

<input id=**"date"** type=**"date"**><br>

</div>

</div>

<form id=**"classNames"**>

<input type=**"radio"** name=**"className"** value=**"Non-Corporate"** id=**"Non-Corporate"** checked=**"checked"**>**Non-Corporate Class**<br>

<input type=**"radio"** name=**"className"** value=**"Corporate"** id=**"Corporate"**>**Corporate Class** <br>

</form>

<button id=**"print"**>**REPORT**</button>

<button class=**"close"**>**CLOSE**</button>

</div>

# **JavaScript**

The JavaScript contains all the functions used to create this quiz dynamically.

All the data comes from an external JSON file that can be changed at any time. Elements’ styling comes from an external CSS file.

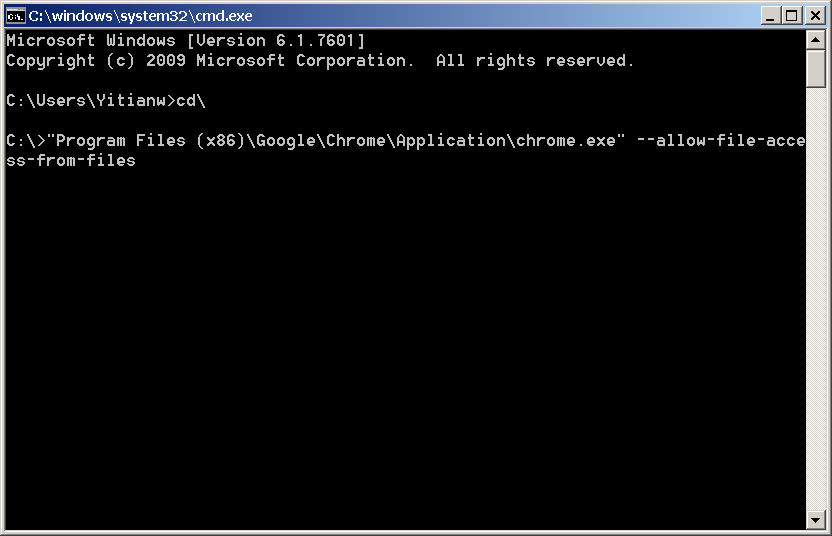
Some of the main features:

* sliding quiz display with dynamic number of slides
* dynamic questions pulled from JSON file
* retrieving the user’s answers
* backend calculations to determine a portfolio match
* dynamic pie chart and fund information
* options to jump directly to portfolios
* pagination
* report generation

All js code has to be in the function “.getJSON” since the js file must access the data in Json then process other functions.

There may have problems for browser opening Json file. The best option is to open the html file is using Chrome.

1. Close all Chrome tabs
2. Open cmd
3. Find the direction of “chrome.exe” using Explorer
4. Insert the direction of chrome in cmd
5. Using the opening window of Chrome



## **Load JSON-encoded Data from The Server**

$**.**getJSON**(**"https://api.myjson.com/bins/gfwb7"**,** **function(**json**)**

Here is the link to the documentation about function “.getjson()” : <http://api.jquery.com/jquery.getjson/>

## **Global** **Variables**

**var** numOfSlides **=** json**.**length**;** // get the amount of slides in the quiz

// (containing coverSlide, questionSlide and resultsSlide)

$**(**"#queue"**).**width**(**numOfSlides**\***800**);** // set the length of slide queue

// each slide has width 800px (set as requirement)

// have to be decided after getting the amount of slides

**var** arr **=** **[];**

**for** **(var** i **=** 0**;** i **<** numOfSlides**;** i**++)** **{** // create an array to store inputs (radio button inputs on questionSlide)

// the length of array is same as number of all slides

// the index of slide is same as the index of item in array

arr**.**push**(**0**);**

**}**

**var** sum**;** // create sum to store the sum of items in array

**var** all\_slides **=** **document.**getElementById**(**"queue"**);** // get the div of slide queue (have created in html)

**var** footer **=** **document.**getElementById**(**"footer"**);** // get the div of footer (have created in html)

// variables used in report form

**var** clientName**,** address1**,** address2**,** address3**,** advisorName**,** firmName**,** phone**,** date**,** userSelectedClass**,** userOption**;**

## **Slides Creation**

// create all slides

**for** **(var** i **=** 0**;** i **<** numOfSlides**;** i**++)** **{**

// create all slides

**var** slide **=** **document.**createElement**(**"li"**);**

slide**.**id **=** "slide" **+** i**;**

all\_slides**.**appendChild**(**slide**);**

**if** **(**json**[**i**].**type **==** "coverPage"**)** **{**

createCoverPageSlide**(**slide**,** i**,** **(**numOfSlides **-** 1**),** footer**);** // the index of last slide

**}** **else** **if** **(**json**[**i**].**type **==** "section"**)** **{**

createSectionSlide**(**slide**,** i**);**

**}** **else** **if** **(**json**[**i**].**type **==** "question"**){**

createQuestionSlide**(**slide**,** i**);**

**}** **else** **if** **(**json**[**i**].**type **==** "results"**){**

createResultsSlide**(**slide**,** i**,** arr**,** sum**);**

**}**

**}**

## **Function for Slide Content Creation**

The whole quiz is divided into four types of pages:

* Cover Page – the very first page of the quiz containing the quiz name and brief introduction
* Section Page – the quiz contains different section. For each section, there is an introduction page
* Question Pages – pages containing question and answer that user is going to take
* Results Page – the page that displays the recommend portfolio based on the questions that user takes

**Cover Slide**

function createCoverPageSlide (slide, i, last\_slide\_index, footer)



// create slide title

**var** title **=** **document.**createElement**(**"h2"**);**

title**.**appendChild**(document.**createTextNode**(**json**[**i**].**title**));**

title**.**className **=** "remove"**;** // to be removed

slide**.**appendChild**(**title**);**

// create intro para

**var** intro **=** **document.**createElement**(**"p"**);**

intro**.**appendChild**(document.**createTextNode**(**json**[**i**].**intro**));**

intro**.**id **=** "cover-slide-para"**;**

intro**.**className **=** "remove"**;** // to be removed

slide**.**appendChild**(**intro**);**

// create button to go to next slide

**var** nextButton **=** **document.**createElement**(**"button"**);**

nextButton**.**appendChild**(document.**createTextNode**(**"Take the questionnaire >>"**));**

nextButton**.**className **=** "next"**;**

nextButton**.**className **+=** " remove"**;** // to be removed

slide**.**appendChild**(**nextButton**);**

// add portfolio drop-down

**var** numOfOptions **=** json**[**i**].**dropDown**.**length**;** // get amount of options

**var** dropDown **=** **document.**createElement**(**"select"**);**

dropDown**.**id **=** "move-dropDown"**;**

dropDown**.**className **=** "drop-down"**;**

slide**.**appendChild**(**dropDown**);**

// add options to drop-down

**for** **(var** j **=** 0**;** j **<** numOfOptions**;** j**++)** **{**

**var** **option** **=** **document.**createElement**(**"option"**);**

**if** **(**j **==** 0**)** **{** // set the first option as default

**option.**disabled **=** "true"**;**

**option.**selected **=** "true"**;**

**}**

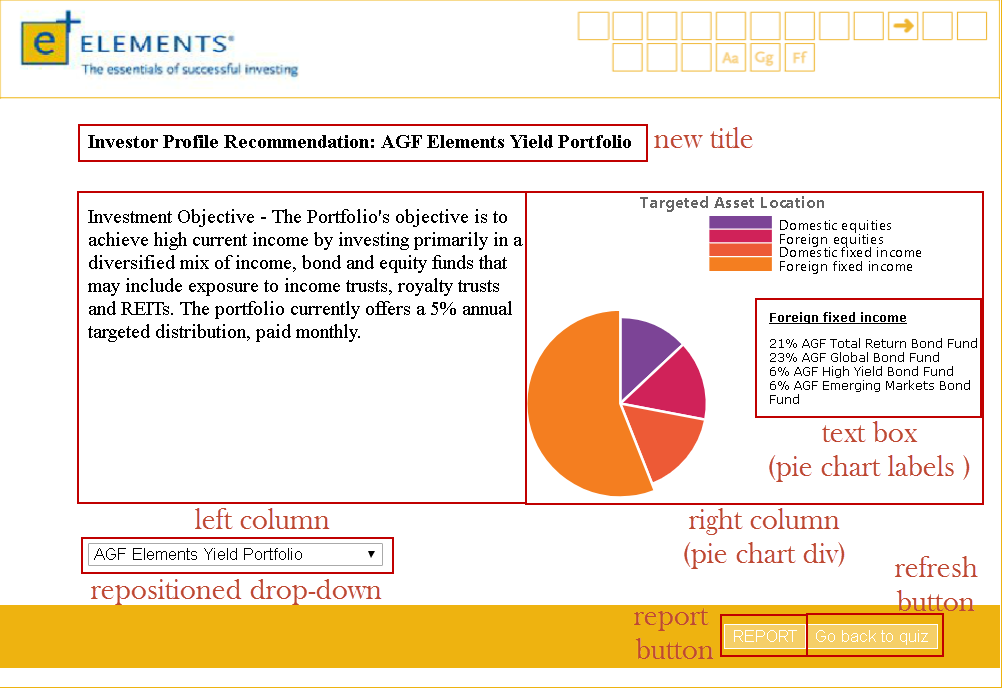
**option.**appendChild**(document.**createTextNode**(**json**[**i**].**dropDown**[**j**].**name**));**

**option.**value **=** json**[**i**].**dropDown**[**j**].**value**;**

dropDown**.**appendChild**(option);**

**}**

When user selects an option from the drop-down, it will process to a results page with the portfolio that user just selected. To create the new slide, elements on the slide are removed (except drop-down is repositioned) then new elements are created on the slide.



// replace elements

**var** flag **=** **false;**

$**(**"#slide" **+** i **+** " .drop-down"**).**change**(function(){**

$**(**"#move-dropDown"**).**addClass**(**"move-dropDown"**);** // reposition the dropDown to the bottom of the slide

$**(**".pagination"**).**hide**();** // hide pagination

$**(**"#report"**).**show**();** // display REPORT button

**if** **(**flag **==** **false)** **{** // remove old elements once

// create new elements once

flag**=true;**

$**(**".remove"**).**remove**();** // remove elements from cover page

// create new elements

//

**var** title **=** **document.**createElement**(**"h4"**);** // create title

title**.**id **=** "results-slide-title"**;**

slide**.**appendChild**(**title**);**

**var** row **=** **document.**createElement**(**"div"**);** // create one row for two columns

row**.**className **=** "results-row"**;**

slide**.**appendChild**(**row**);**

**var** leftColumn **=** **document.**createElement**(**"div"**);** // left column

leftColumn**.**className **=** "results-column"**;**

leftColumn**.**id **=** "leftColumn"**;**

row**.**appendChild**(**leftColumn**);**

**var** rightColumn **=** **document.**createElement**(**"div"**);** // right column

rightColumn**.**className **=** "results-column"**;**

rightColumn**.**id **=** "rightColumn"**;**

row**.**appendChild**(**rightColumn**);**

**var** pieChart **=** **document.**createElement**(**"div"**);** // add pieChart div to right column

pieChart**.**id **=** "canvas-holder"**;**

rightColumn**.**appendChild**(**pieChart**);**

**var** fundBox **=** **document.**createElement**(**"div"**);** // add text box for pieChart labels

fundBox**.**id **=** "fundBox"**;**

rightColumn**.**appendChild**(**fundBox**);**

**var** funds **=** **document.**createElement**(**"p"**);**

funds**.**id **=** "funds"**;**

fundBox**.**appendChild**(**funds**);**

**var** refreshButton **=** **document.**createElement**(**"button"**);** // add REFRESH button for returning back to quiz

refreshButton**.**appendChild**(document.**createTextNode**(**"Go back to quiz"**));**

refreshButton**.**id **=** "refresh"**;**

footer**.**appendChild**(**refreshButton**);**

$**(**"#refresh"**).**click**(function(){**

**window.location.**reload**();**

**});**

**}**

// get piechart info from the last slide (Results slide)

i **=** last\_slide\_index**;**

$**(this).**find**(**"option:selected"**).**each**(function(){**

**var** optionValue **=** $**(this).**attr**(**"value"**);** // get value from dropDown option then create pieChart

userOption **=** optionValue**;**

**var** arg **=** eval**(**"json[i]."**+** optionValue**);** // get the type of portfolio

// remove exitsing pieChart and iframe to avoid duplication

$**(**"iframe"**).**remove**();**

$**(**"canvas"**).**remove**();**

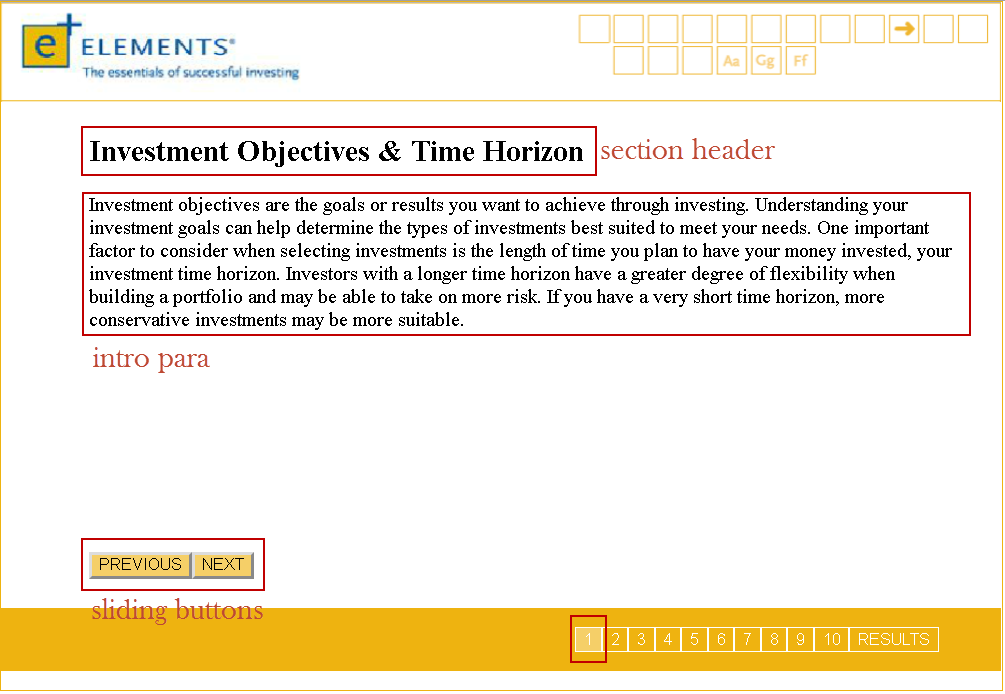
createPieChart**(**slide**,** i**,** arg**);** // call function to create pie-chart

**});**

**});**

### **Section Slide**

**function** createSectionSlide**(**slide**,** i**)**



// create section header

**var** title **=** **document.**createElement**(**"h2"**);**

title**.**appendChild**(document.**createTextNode**(**json**[**i**].**title**))**

slide**.**appendChild**(**title**);**

// create section intro paragraph

**var** intro **=** **document.**createElement**(**"p"**);**

intro**.**appendChild**(document.**createTextNode**(**json**[**i**].**intro**));**

intro**.**className **=** "section-intro"**;**

slide**.**appendChild**(**intro**);**

// create Previous and Next buttons to control sliding

**var** previouButton **=** **document.**createElement**(**"button"**);**

previouButton**.**appendChild**(document.**createTextNode**(**"PREVIOUS"**)section );**

previouButton**.**className **=** "previous"**;**

slide**.**appendChild**(**previouButton**);**

**var** nextButton **=** **document.**createElement**(**"button"**);**

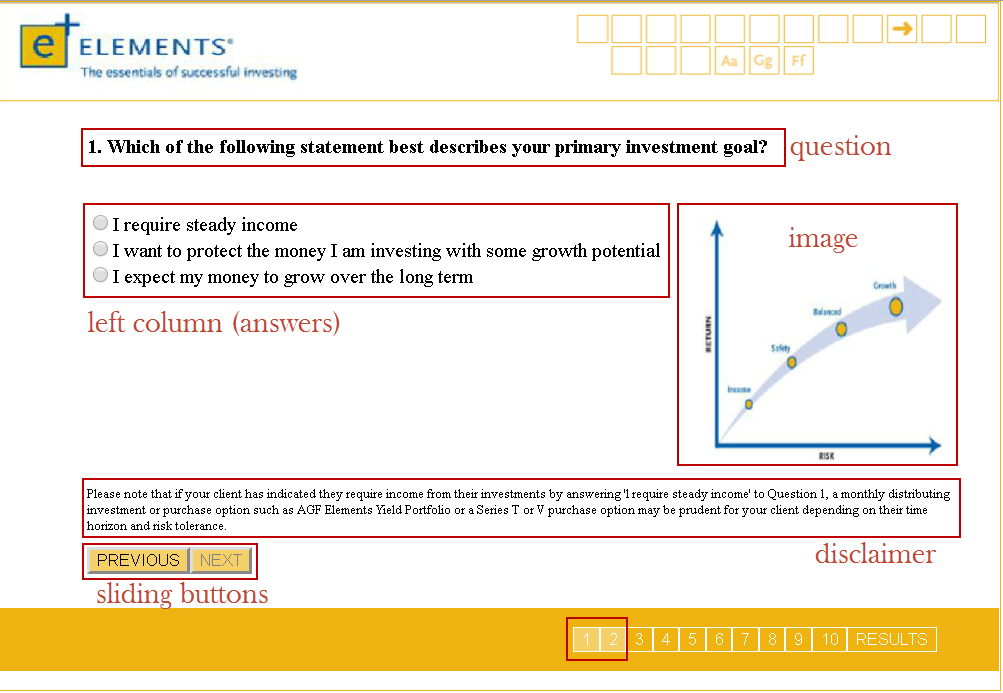
nextButton**.**appendChild**(document.**createTextNode**(**"NEXT"**));**

nextButton**.**className **=** "next"**;**

slide**.**appendChild**(**nextButton**);**

**Question Slide**

**function** createQuestionSlide**(**slide**,** i**)**



// create question

**var** question **=** **document.**createElement**(**"h4"**);**

question**.**appendChild**(document.**createTextNode**(**json**[**i**].**question**));**

question**.**id **=** "question"**;**

slide**.**appendChild**(**question**);**

// create one row for two columns

**var** row **=** **document.**createElement**(**"div"**);**

row**.**className **=** "row"**;**

slide**.**appendChild**(**row**);**

// create left column for answer list

**var** leftColumn **=** **document.**createElement**(**"div"**);**

leftColumn**.**className **=** "question-column"**;**

leftColumn**.**className **+=** " question-left"**;**

row**.**appendChild**(**leftColumn**);**

// create right column for img

**var** rightColumn **=** **document.**createElement**(**"div"**);**

rightColumn**.**className **=** "question-column"**;**

rightColumn**.**className **+=** " question-right"**;**

row**.**appendChild**(**rightColumn**);**

// create Form for answer list

**var** formElement **=** **document.**createElement**(**"form"**);**

formElement**.**id **=** "form" **+** i**;**

leftColumn**.**appendChild**(**formElement**);**

// add Answers to the slide

**var** numOfAnswers **=** json**[**i**].**answers**.**length**;**

**for** **(var** j **=** 0**;** j **<** numOfAnswers**;** j**++)** **{**

**function** addRadioButton**(text,** value**){**

**var** label **=** **document.**createElement**(**"label"**);**

**var** **radio** **=** **document.**createElement**(**"input"**);**

**radio.**className **=** "radio-button"**;**

**radio.**setAttribute**(**"type"**,** "radio"**);**

**radio.**setAttribute**(**"name"**,** "radioButton"**);**

**radio.**setAttribute**(**"value"**,** value**);**

**radio.**id **=** i**;**

label**.**appendChild**(radio);**

label**.**appendChild**(document.**createTextNode**(text));** // add text to radio buttons

formElement**.**appendChild**(**label**);**

**}**

addRadioButton**(**json**[**i**].**answers**[**j**].text,** json**[**i**].**answers**[**j**].**value**);** // add value to radio buttons

**}**

// add disclaimer

**if** **(**json**[**i**].**disclaimer **!=** **null){**

**var** disclaimer **=** **document.**createElement**(**"p"**);**

disclaimer**.**className **=** "disclaimer"**;**

slide**.**appendChild**(**disclaimer**);**

disclaimer**.**appendChild**(document.**createTextNode**(**json**[**i**].**disclaimer**));**

**}**

// add img to right column

**if** **(**json**[**i**].**img **!=** **null)** **{**

**var** img **=** **document.**createElement**(**"IMG"**);**

img**.**className **=** "img"**;**

img**.**setAttribute**(**"src"**,** json**[**i**].**img**);**

img**.**setAttribute**(**"alt"**,** "chart display error"**);**

rightColumn**.**appendChild**(**img**);**

**}**

// create two buttons to control sliding

//

**var** previouButton **=** **document.**createElement**(**"button"**);**

previouButton**.**appendChild**(document.**createTextNode**(**"PREVIOUS"**));**

previouButton**.**className **=** "previous"**;**

slide**.**appendChild**(**previouButton**);**

**var** nextButton **=** **document.**createElement**(**"button"**);**

nextButton**.**appendChild**(document.**createTextNode**(**"NEXT"**));**

nextButton**.**className **=** "next"**;**

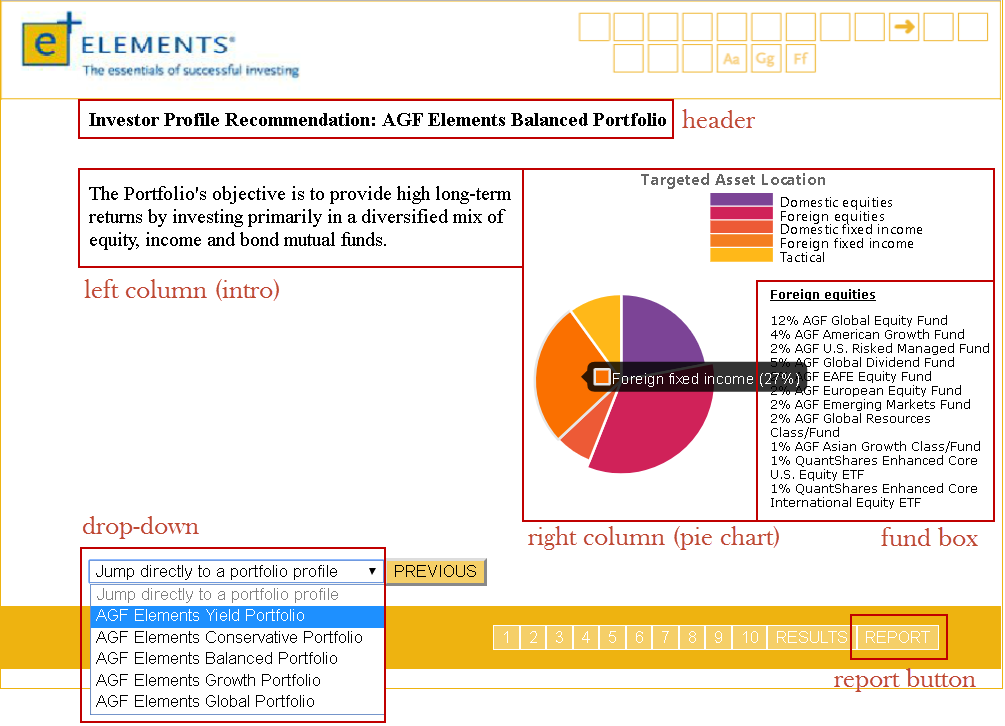
nextButton**.**id **=** **(**"next" **+** i**);**

nextButton**.**disabled **=** **true;**

slide**.**appendChild**(**nextButton**);**

### **Results Slide**

**function** createResultsSlide**(**slide**,** i**,** arr**,** sum**)**



// slide title

**var** title **=** **document.**createElement**(**"h4"**);**

title**.**id **=** "results-slide-title"**;**

slide**.**appendChild**(**title**);**

**var** row **=** **document.**createElement**(**"div"**);**

row**.**className **=** "results-row"**;**

slide**.**appendChild**(**row**);**

**var** leftColumn **=** **document.**createElement**(**"div"**);**

leftColumn**.**className **=** "results-column"**;**

leftColumn**.**id **=** "leftColumn"**;**

row**.**appendChild**(**leftColumn**);**

**var** rightColumn **=** **document.**createElement**(**"div"**);**

rightColumn**.**className **=** "results-column"**;**

rightColumn**.**id **=** "rightColumn"**;**

row**.**appendChild**(**rightColumn**);**

**var** pieChart **=** **document.**createElement**(**"div"**);**

pieChart**.**id **=** "canvas-holder"**;**

rightColumn**.**appendChild**(**pieChart**);**

**var** fundBox **=** **document.**createElement**(**"div"**);**

fundBox**.**id **=** "fundBox"**;**

rightColumn**.**appendChild**(**fundBox**);**

**var** funds **=** **document.**createElement**(**"p"**);**

funds**.**id **=** "funds"**;**

fundBox**.**appendChild**(**funds**);**

// add portfolio drop-down

**var** numOfOptions **=** json**[**i**].**dropDown**.**length**;** // get amount of options

**var** dropDown **=** **document.**createElement**(**"select"**);**

dropDown**.**className **=** "drop-down"**;**

// add options to drop-down

**for** **(var** j **=** 0**;** j **<** numOfOptions**;** j**++)** **{**

**var** **option** **=** **document.**createElement**(**"option"**);**

**if** **(**j **==** 0**)** **{**

**option.**disabled **=** "true"**;**

**option.**selected **=** "true"**;**

**}**

**option.**appendChild**(document.**createTextNode**(**json**[**i**].**dropDown**[**j**].**name**));**

**option.**value **=** json**[**i**].**dropDown**[**j**].**value**;**

dropDown**.**appendChild**(option);**

**}**

slide**.**appendChild**(**dropDown**);**

// create only previous button

**var** previouButton **=** **document.**createElement**(**"button"**);**

previouButton**.**appendChild**(document.**createTextNode**(**"PREVIOUS"**));**

previouButton**.**className **=** "previous"**;**

slide**.**appendChild**(**previouButton**);**

// create pop-up window for report form

**var** modal **=** **document.**createElement**(**"div"**);**

modal**.**id **=** "myModal"**;**

slide**.**appendChild**(**modal**);**

// append the pre-build report window (from html file) to pop-up window

**var** reportContent **=** **document.**getElementById**(**"report-window"**);**

modal**.**appendChild**(**reportContent**);**

$**(**"#next" **+** **(**i **-** 1**)).**click**(function(){** // when user process to last slide (results slide)

sum **=** 0**;**

**for** **(var** j **=** 0**;** j **<** arr**.**length**;** j**++)** **{** // calculate the sum getting from user

sum **=** sum **+** **parseInt(**arr**[**j**]);**

**}**

// remove existing pieChart and iframe to avoid duplication

$**(**"iframe"**).**remove**();**

$**(**"canvas"**).**remove**();**

**if** **(**sum **<** 18**)** **{**

createPieChart**(**slide**,** i**,** json**[**i**].**yield**);**

userOption **=** "yield"**;**

**}** **else** **if** **(**sum **>=** 18 **&&** sum **<=** 30**)** **{**

createPieChart**(**slide**,** i**,** json**[**i**].**conservative**);**

userOption **=** "conservative"**;**

**}** **else** **if** **(**sum **>=** 31 **&&** sum **<=** 43**)** **{**

createPieChart**(**slide**,** i**,** json**[**i**].**balanced**);**

userOption **=** "balanced"**;**

**}** **else** **if** **(**sum **>=** 44 **&&** sum **<=** 55**)** **{**

createPieChart**(**slide**,** i**,** json**[**i**].**growth**);**

userOption **=** "growth"**;**

**}** **else** **if** **(**sum **>** 55**)** **{**

createPieChart**(**slide**,** i**,** json**[**i**].**global**);**

userOption **=** "global"**;**

**}**

$**(**"#report"**).**show**();** // display REPORT button

**});**

// change content on RESULTS slide by changing dropdown options

$**(**"#slide" **+** i **+**" .drop-down"**).**change**(function(){**

$**(this).**find**(**"option:selected"**).**each**(function(){**

**var** optionValue **=** $**(this).**attr**(**"value"**);**

userOption **=** optionValue**;**

**var** arg **=** eval**(**"json[i]."**+** optionValue**);**

// remove exitsing pieChart and iframe to avoid duplication

$**(**"iframe"**).**remove**();**

$**(**"canvas"**).**remove**();**

createPieChart**(**slide**,** i**,** arg**);**

**});**

**});**

# JSON

The JSON (JavaScript Object Notation) file is the file that feeds data into the JavaScript so that elements can be built. It allows the project to be dynamic, meaning “changing”, as none of the data is hard-coded into the original source code. Quiz elements like questions, answers, answer weightings, pie chart data, and even quiz slides, are all created with objects from the JSON file.

The JSON file is meant to be easy for humans to read and write, which is why it was used in this project. The goal was to make updating and maintaining the quiz simple and intuitive enough so that someone without coding experience could do it.

## Cover Page

**{**

"type" **:** "coverPage"**,**

"title"**:** "Find your Element - INVESTOR QUESTIONNAIRE"**,**

"intro"**:** "This questionnaire is designed to help you choose the right investment to meet your needs by matching your unique situation with the profile of one of the five AGF Elements portfolios. Your tolerance for risk, your time horizon and your expected rate of return over the life of the investment all play a part in finding the right Elements portfolio for you."**,**

"dropDown" **:** **[**

**{**"name" **:** "Jump directly to a portfolio profile"**,** "value" **:** ""**},**

**{**"name" **:** "AGF Elements Yield Portfolio"**,** "value" **:** "yield"**},**

**{**"name" **:** "AGF Elements Conservative Portfolio"**,** "value" **:** "conservative"**},**

**{**"name" **:** "AGF Elements Balanced Portfolio"**,** "value" **:** "balanced"**},**

**{**"name" **:** "AGF Elements Growth Portfolio"**,** "value" **:** "growth"**},**

**{**"name" **:** "AGF Elements Global Portfolio"**,** "value" **:** "global"**}**

**]**

**},**

Section Page

**{**

"type" **:** "section"**,**

"title" **:** "Investment Objectives & Time Horizon"**,**

"intro" **:** "Investment objectives are the goals or results you want to achieve through investing. Understanding your investment goals can help determine the types of investments best suited to meet your needs. One important factor to consider when selecting investments is the length of time you plan to have your money invested, your investment time horizon. Investors with a longer time horizon have a greater degree of flexibility when building a portfolio and may be able to take on more risk. If you have a very short time horizon, more conservative investments may be more suitable."

**},**

## Question Page

**{**

"type" **:** "question"**,**

"question" **:** "1. Which of the following statement best describes your primary investment goal?"**,**

"answers" **:** **[**

**{**"text" **:** "I require steady income"**,** "value" **:** "0"**},**

**{**"text" **:** "I want to protect the money I am investing with some growth potential"**,** "value" **:** "4"**},**

**{**"text" **:** "I expect my money to grow over the long term"**,** "value" **:** "8"**}**

**],**

"disclaimer" **:** "Please note that if your client has indicated they require income from their investments by answering 'l require steady income' to Question 1, a monthly distributing investment or purchase option such as AGF Elements Yield Portfolio or a Series T or V purchase option may be prudent for your client depending on their time horizon and risk tolerance."**,**

"img" **:** ".\/q1.png"

**},**