Software Design and Development: Major Work

Defining and Understanding The Problem

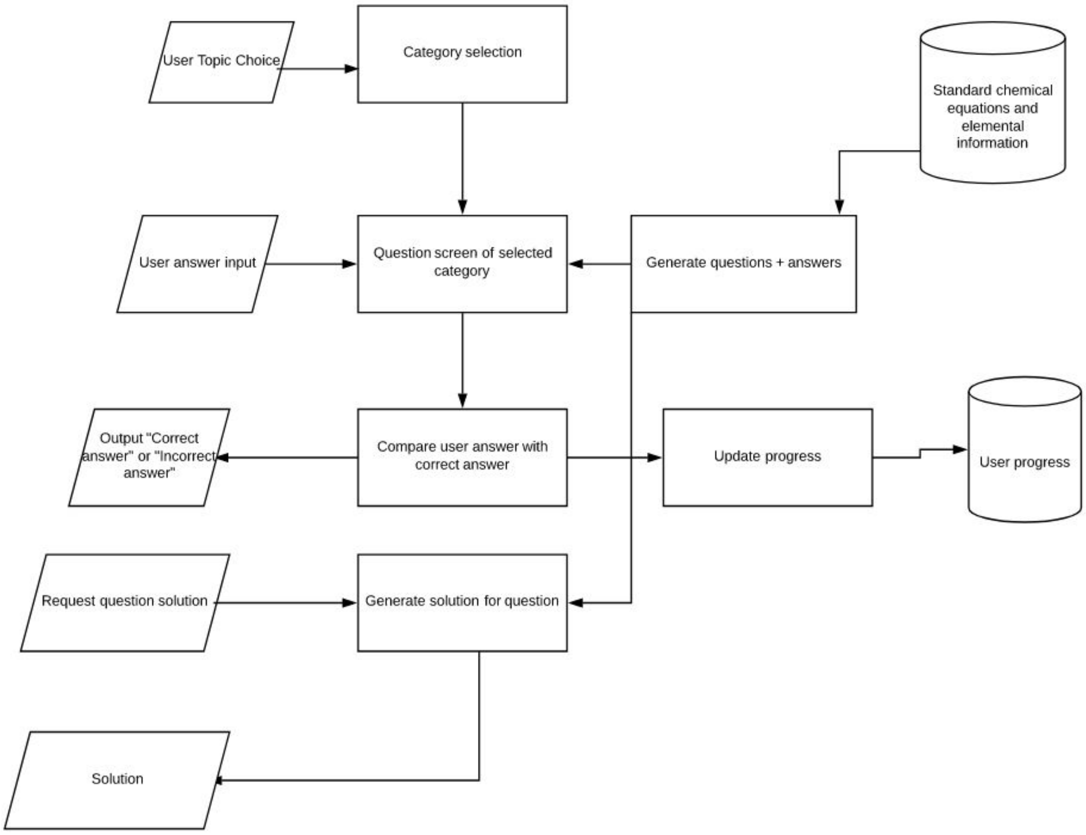
Requirements:

- A web-based chemistry learning game for the practicing of the mathematical aspects of chemistry.
- Automatic generation of questions relating to: stoichiometry, energy, and half life and pH.
- Automatic generation of solutions specifically for each question.
- Tracking of progression with successful answering of questions

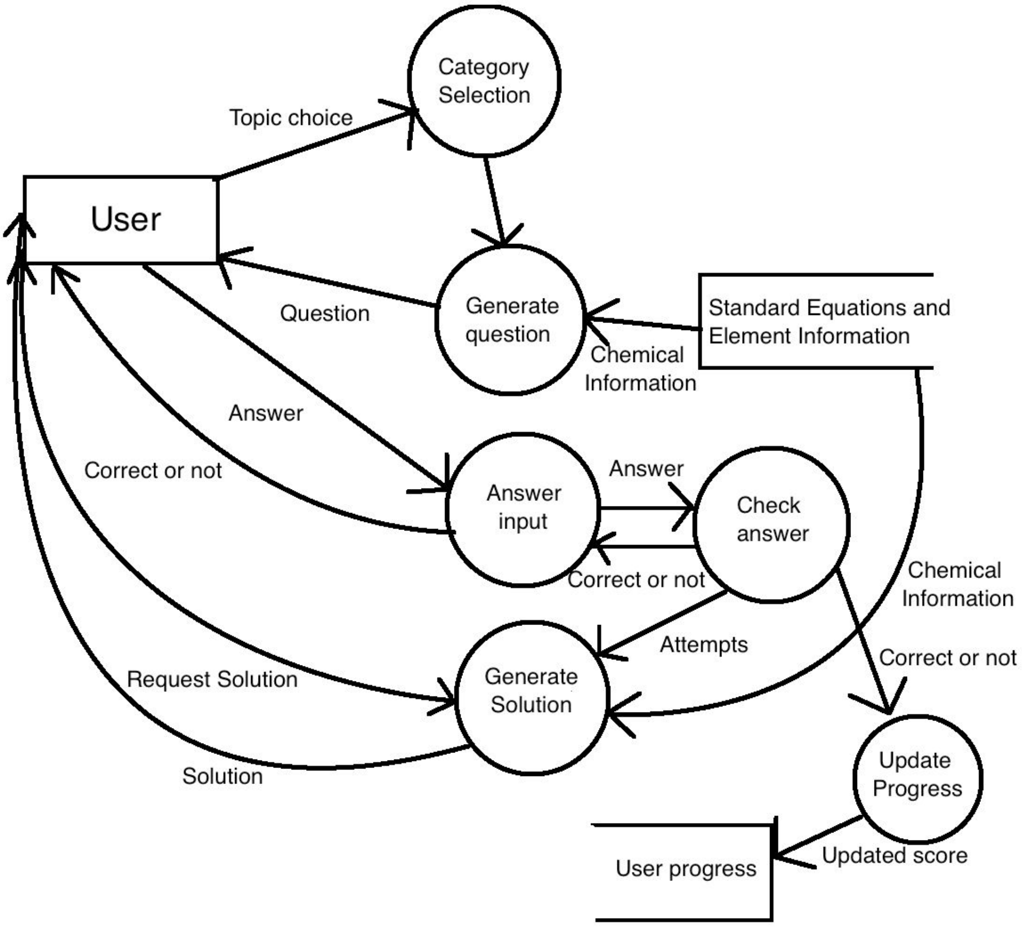
Specifications:

- An intuitive web based software with interface similar to industry standards
- Appropriate loading messages to give a sense of control of the software.
- Use of English as the language in all messages in order to be most inclusive
- Appropriate font size, and colour to highlight information to accommodate visual impairments
- Optimised software with reasonable performance on a wide range of hardware.

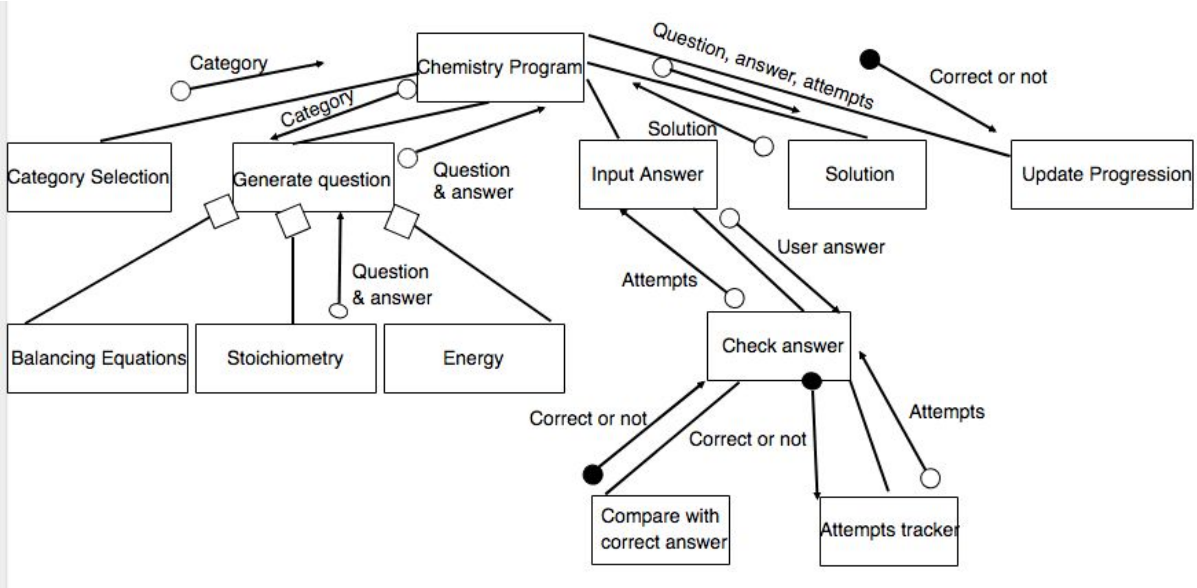
System flowchart:


User Topic Choice
Category selection
Standard chemical equations and elemental information
User answer input
Question screen of selected category
Generate questions + answers
Output "Correct answer" or "Incorrect answer"
Compare user answer with correct answer
Update progress
User progress
Request question solution
Generate solution for question
Solution

Data flow diagram:



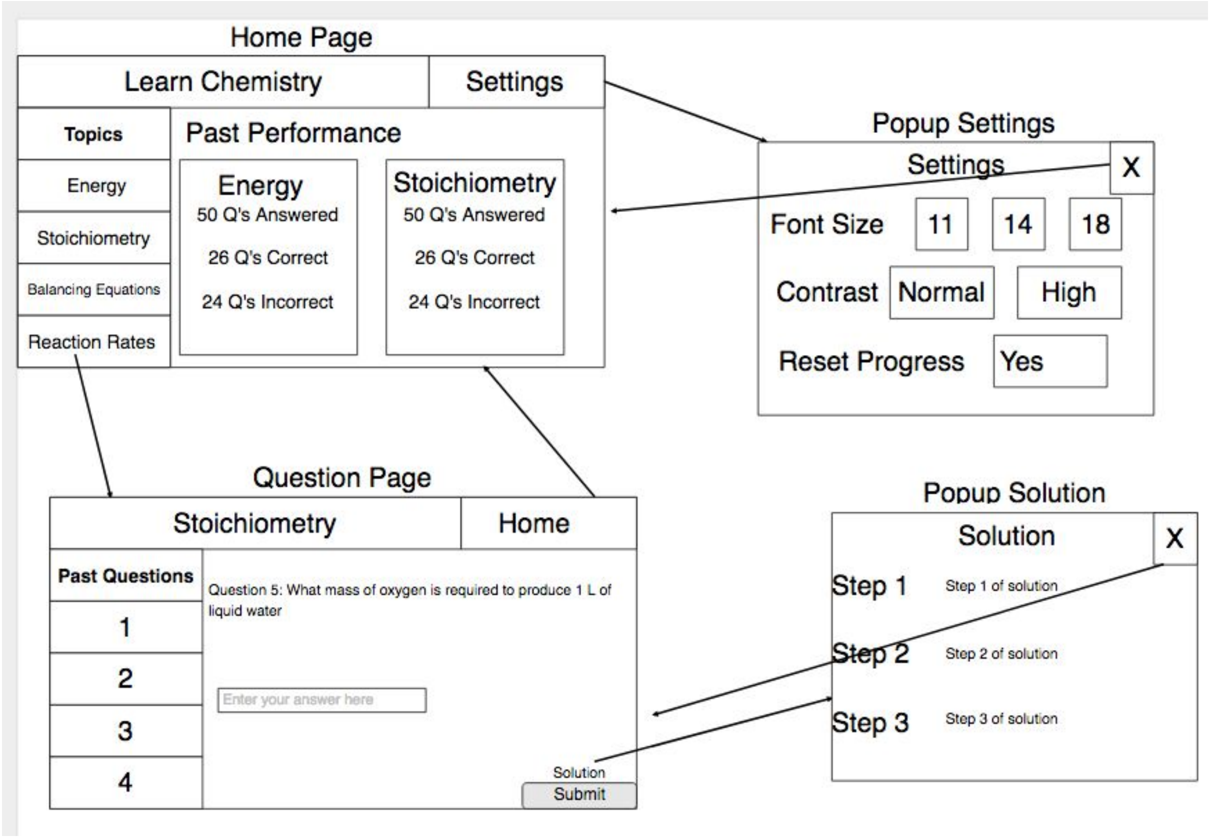
Structured chart:



IPO Diagrams:

Input	Process	Output
User makes category selection	Load appropriate category Generate question: - Using the predefined question structure within the category, replace placeholder information with generated values Generate answer: - Using the formula written into the section and information from datasheet, mathematically calculate the answer with the generated values	Specific category question
User inputs their answer	Compare user answer with generated answer.	If comparison returns true, display “Correct” If comparison returns false, display “False”
User requests answer solution	Take a pre created structured solution. Modify placeholder values with the values for this specific question.	Solution
User attempts inputted into attempts tracker	If question answered correctly on first go, update correct answers progression tracker.	Number of correct answers

Storyboard and screen design:



Planning and Designing

Research into program creation

Similar products

The most similar product to the program I will be developing is Khan Academy. However while Khan Academy places a heavy emphasis on actual content teaching, my program will purely focus on the practicing of the mathematical and quantitative aspects of chemistry. Just like Khan Academy, questions will be generated as well as answers for this particular question. Additionally their will be progress mechanics for the tracking of correct and incorrect answering of questions. This acts as a gamification of learning that aims to incentivise users to keep on practicing.

One aspect of the Khan Academy user interface I found appealing was the use of popup windows on the same page. I discovered these were known as modals, and there are various Javascript plugins that enabled them. I've opted to use Tingle modals developed by Robin Parisi for the modals in my software solution

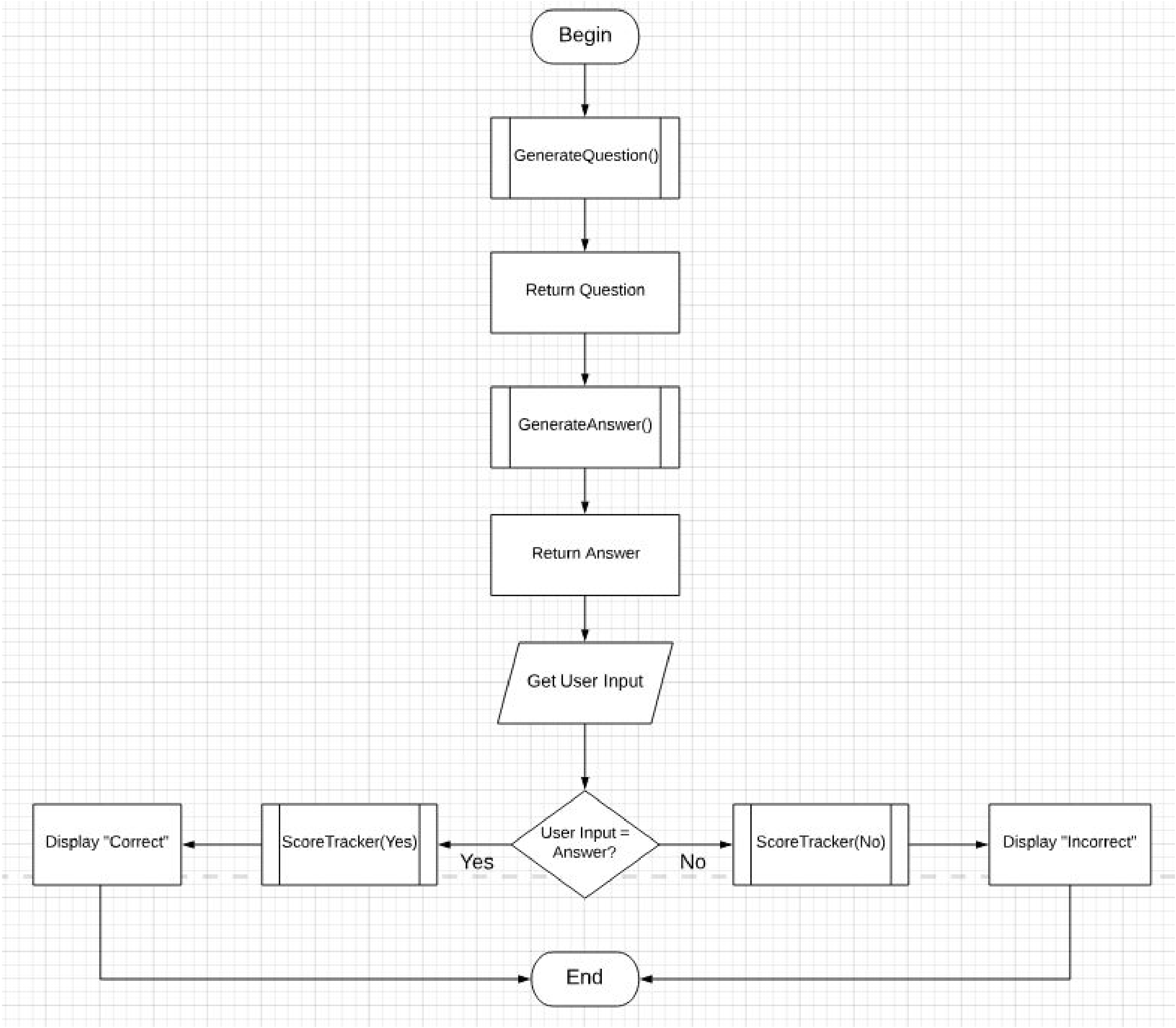
Standard Algorithms:

- Choosing between two options with equal probability
    - Essentially generating a boolean value. Achieved by using the built in random function that generates a number between 0 and 1. Then returning the result of checking if greater than 0.5 or not. This will either return true or false with theoretically equal probability.

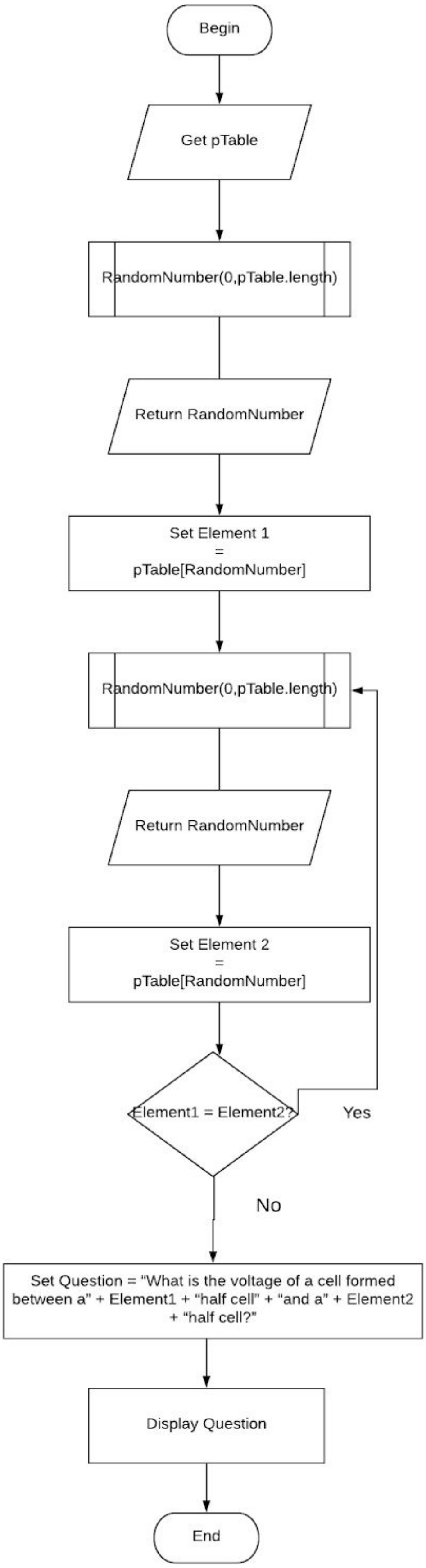
- String manipulation
    - As subscripts and superscripts are extremely common in chemical formulas and equations, it would be beneficial to use standard algorithms to manipulate each formula or equation to apply superscripts and subscripts. Strings are essentially arrays of characters therefore they can be iterated over. When a number is detected at some index, this character can be changed, and then reinserted at the same index.

Algorithm flowcharts

Choosing Answer:



ElectroChemistry Generate Question:


Begin
Get pTable
RandomNumber(0,pTable.length)
Return RandomNumber
Set Element 1
=
pTable[RandomNumber]
RandomNumber(0,pTable.length)
Return RandomNumber
Set Element 2
=
pTable[RandomNumber]
Element1 = Element2?
Yes
No
Set Question = "What is the voltage of a cell formed between a" + Element1 + "half cell" + "and a" + Element2 + "half cell?"
Display Question
End

Pseudocode

Choosing Answer:

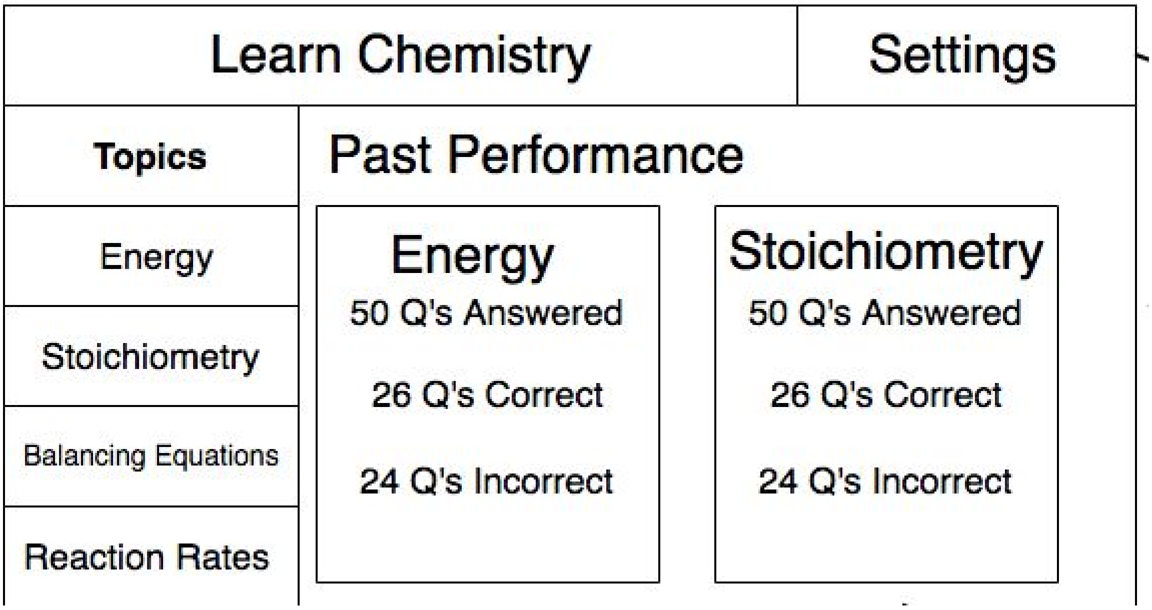
```
BEGIN ChooseAnswer
Question = GenerateQuestion()
Answer = GenerateAnswer()
get UserInput


IF UserInput = Answer THEN
    ScoreTracker(Correct)
    Display "Correct"

ELSE
    ScoreTracker(Incorrect)
    Display "Incorrect"
ENDIF
END ChooseAnswer
```

Electrochemistry Generate Question:

```
BEGIN Electrochemistry.GenQuestion

Let R(x,y) generate an integer between x and y

Class Elements{
    Private-
        Name: string
        Symbol: string
        Valency: integer
        EMF: integer

pTable = [Numerous Element objects….]
Element1 = pTable(R(0, pTable.length))

Flag = true
WHILE Flag == true:
    Element2 = pTable(R(0, pTable.length))
    IF Element2 != Element1:
        Flag == false
    EndIF
ENDWHILE

Question = "What is the voltage of a cell formed between a" + Element1 + "half cell" + "and a" + Element2 + "half cell?"
```

Display Question
END Electrochemistry.GenQuestion

User interface design



Learn Chemistry

Settings

Topics

Energy

Stoichiometry

Balancing Equations

Reaction Rates

Past Performance

Energy

50 Q's Answered

26 Q's Correct

24 Q's Incorrect

Stoichiometry

50 Q's Answered

26 Q's Correct

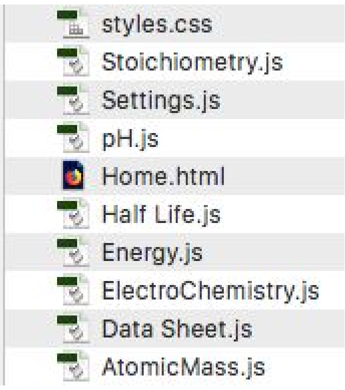
24 Q's Incorrect

Home page of site. Large text helps improve readability for a wide range of users. This allows the software to be inclusive towards possibly with mild visual impairments. Additionally use simple of English language helps comprehensibility to large groups of people allowing the software to be used by many.

Implementing

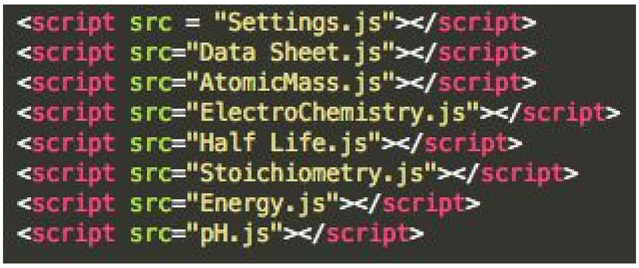
Creation and organisation of code and other elements

Due to the nature of my program featuring multiple separate and unrelated modules, it makes sense to keep them in separate files. This form of organisation allows for a clearer development workspace that avoids any unnecessary confusion.



styles.css
Stoichiometry.js
Settings.js
pH.js
Home.html
Half Life.js
Energy.js
ElectroChemistry.js
Data Sheet.js
AtomicMass.js

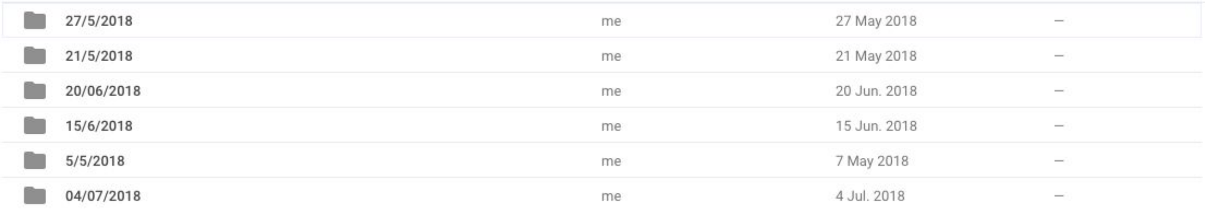
Because of the use of modals on my application, all modules operate within the same HTML document. This requires the linking of the separate Javascript files into one HTML file.



```
<script src = "Settings.js"></script>
<script src="Data Sheet.js"></script>
<script src="AtomicMass.js"></script>
<script src="ElectroChemistry.js"></script>
<script src="Half Life.js"></script>
<script src="Stoichiometry.js"></script>
<script src="Energy.js"></script>
<script src="pH.js"></script>
```

Version control

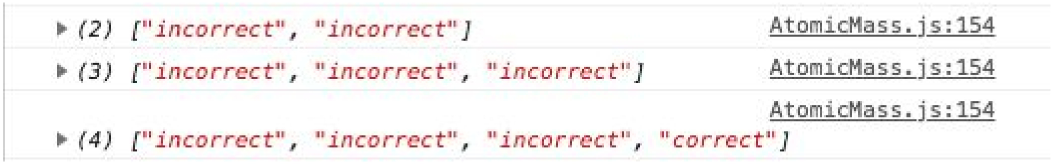
During the development progress, regular backup copies of source code were made both locally, and onto the cloud. This protects the source code against any factors that may cause loss of data, as well as allows for reverting to older versions if a modification made hinders the program in some way. The progress within each version is further tracked within the logbook to appropriately log each significant change.



27/5/2018	me	27 May 2018	–
21/5/2018	me	21 May 2018	–
20/06/2018	me	20 Jun. 2018	–
15/6/2018	me	15 Jun. 2018	–
5/5/2018	me	7 May 2018	–
04/07/2018	me	4 Jul. 2018	–

Description of testing strategies (stubs, flags, debugging output statements, desk checks, breakpoints, traces, variable watching, stepping, etc)

Variable watching: There are various variables in my program that are fundamental to its operation and are constantly manipulated to achieve this. One example of such is the use of the “scorearray”, an array that records the user’s order of choices for a given question. This contents of this array are vital to the successful updating of score, thus variable watching must be applied to it to ensure appropriate operation.

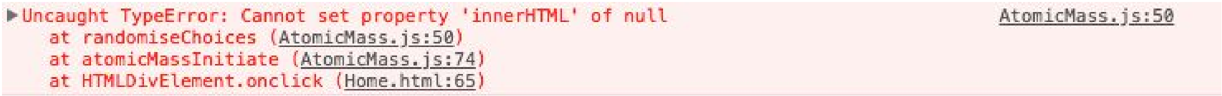


```
▸ (2) ["incorrect", "incorrect"]                              AtomicMass.js:154
▸ (3) ["incorrect", "incorrect", "incorrect"]                 AtomicMass.js:154
                                                              AtomicMass.js:154
▸ (4) ["incorrect", "incorrect", "incorrect", "correct"]
```

The “scorearray” being watched as questions are being answered.

Javascript console: Google Chrome features a Javascript console intended to be a tool for developers. Any runtime errors that occur within the program will log the type of error and the corresponding line within the code. This is a very powerful debugging tool that simplifies development by pointing directly to the source of error.

Below is example of an error resulting from incorrect referencing of HTML element that was caught by Javascript console. The type of error is stated as well as the lines of code that caused this error.



```
▶Uncaught TypeError: Cannot set property 'innerHTML' of null                    AtomicMass.js:50
    at randomiseChoices (AtomicMass.js:50)
    at atomicMassInitiate (AtomicMass.js:74)
    at HTMLDivElement.onclick (Home.html:65)
```

Breakpoints: Breakpoints are used to temporarily halt the execution of the code. This allows for the examining of the values of variables

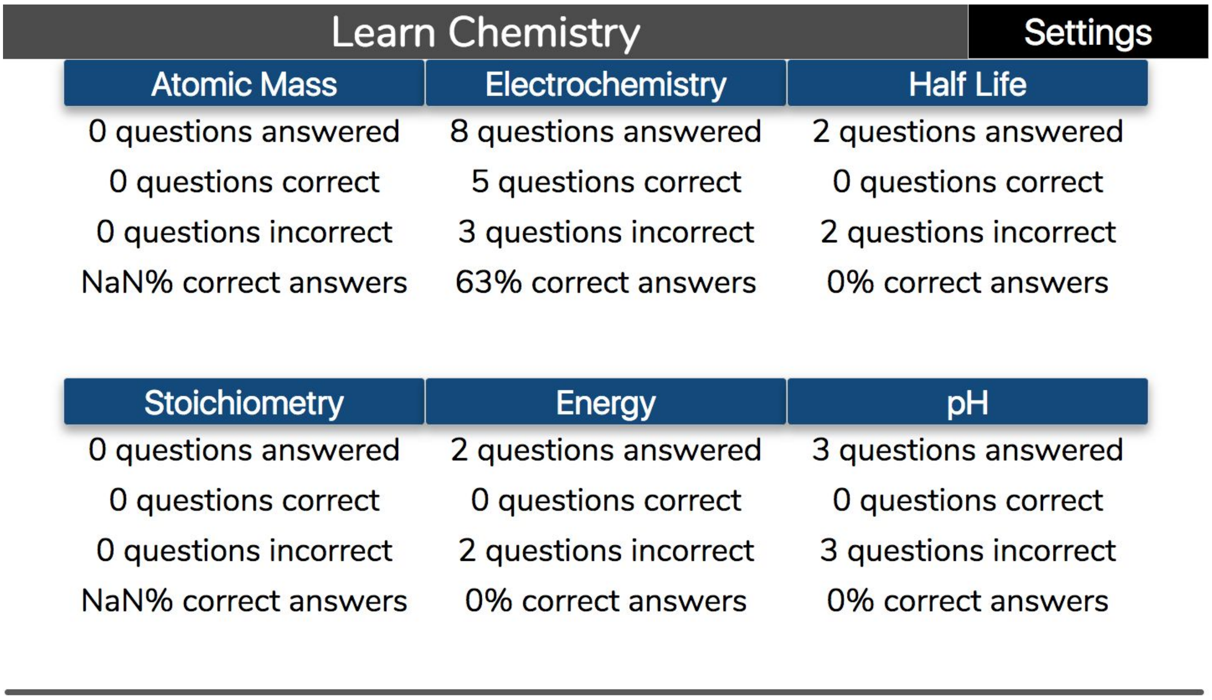
Table of errors

Error	Type	Solution
Uncaught TypeError: Cannot read property 'name' of undefined	Runtime	Caused by improper RNG, causing out of index problem. Readjusted RNG values.
For energy module, generation of answers not mathematically correct.	Logic	Caused by incorrect use of enthalpy formula. Used heat capacity of ethanol instead of water. Changed the formula to appropriately use the heat capacity of water.
Score counter not updating correctly answered questions	Logic	An array was used to determine if question was answered correctly first time. But this array was not reinitialised after each question.
Score counter still not updating correctly answered	Logic	The score array was reinitialised before the actual score update functions can work. Fixed by reordering the functions to make the score update run before reinitialisation.
Answers and choices all generated to be 0 in pH section.	Logic	pH generally works with really small values of which the rounding I used cannot properly handle. Instead of rounding to 2dp for each calculation, display 2dp to the user, but keep the actual variable in its full length.
Choice generation has bias with higher likelihood of generating values greater than the actual answer.	Logic	I was using Math.floor((Math.random() * Max) + Min) instead of Math.floor((Math.random() * Max - Min) + Min)

In Stoichiometry module the choice of reactant for the question was not randomising.	Syntax	The condition in the if/else statement was using “=” instead of “==” for comparison
In Stoichiometry module, the third choice option does nothing	Syntax	It was a spelling error where it was calling Stoichioemtry.chooseAnswer() instead of Stoichiometry.chooseAnswer()
“Uncaught RangeError: Maximum call stack size exceeded”	Syntax/runtime	A function was supposed to call a different function, but I mistyped and made it call itself, thus running into an infinite loop.

Social and ethical considerations

Ease of use: The user interface of the software solution is designed to be intuitive and easy to use for users. It is intended that users should be able to easily understand and operate this product upon first use. This has been achieved by implementing an interface that is both internally consistent within the application and also consistent with industry standards.



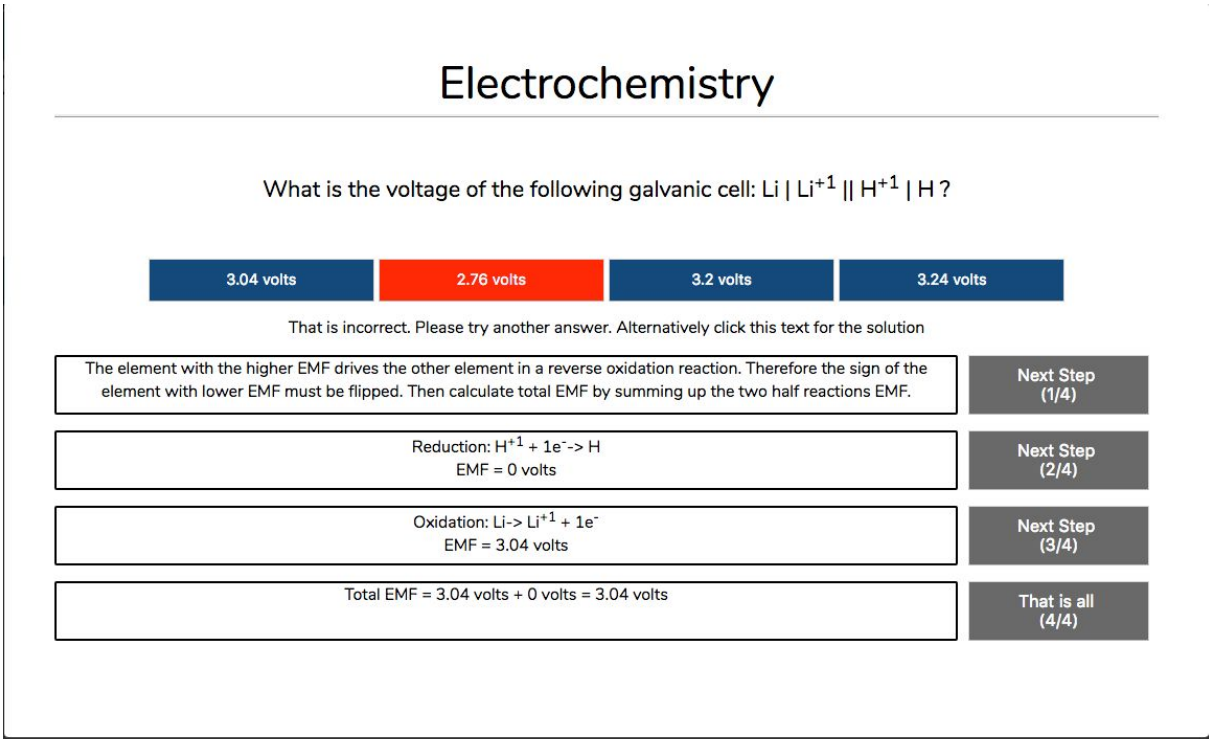
Learn Chemistry

Settings

Atomic Mass	Electrochemistry	Half Life
0 questions answered	8 questions answered	2 questions answered
0 questions correct	5 questions correct	0 questions correct
0 questions incorrect	3 questions incorrect	2 questions incorrect
NaN% correct answers	63% correct answers	0% correct answers

Stoichiometry	Energy	pH
0 questions answered	2 questions answered	3 questions answered
0 questions correct	0 questions correct	0 questions correct
0 questions incorrect	2 questions incorrect	3 questions incorrect
NaN% correct answers	0% correct answers	0% correct answers

For example, all the information regarding the modules on the homescreen and implemented with essentially the same design. This design is fairly conventional without the reinvention of industry standards. The buttons in particular have been designed to be as intuitive as possible by following standard button design principles. A combination of a shadow and hover effect has been used to give the buttons extra dimensionality, which is intended to allow users to know that they can be pressed. Then the actual pressing of the button features a short animation in order to give users feedback.



Electrochemistry

What is the voltage of the following galvanic cell: Li | Li^{+1} || H^{+1} | H ?

That is incorrect. Please try another answer. Alternatively click this text for the solution

The element with the higher EMF drives the other element in a reverse oxidation reaction. Therefore the sign of the element with lower EMF must be flipped. Then calculate total EMF by summing up the two half reactions EMF.

Reduction: H^{+1} + $1e^{-}$-> H
EMF = 0 volts

Oxidation: Li-> Li^{+1} + $1e^{-}$
EMF = 3.04 volts

Total EMF = 3.04 volts + 0 volts = 3.04 volts

This consistency in design is further applied within each section where all basic functionality is achieved using the same set of inputs. By using the same set of rules across the entire application, users quickly grow accustomed and build a mental model of the software. This allows users to quickly understand the program without the need to reexplain for each module. Once they learnt how to use the first section, they should be able to use all other sections with little issues.

Plagiarism: As a large emphasis of object oriented programming is placed on principles of reusability, I have opted to use existing code created by other developers. This is the case with the implementation of modals, which uses the Tingle modal plugin developed by Robin Parisi. The issue of plagiarism has been considered in this case as I have acknowledged the author of the work, and I have not claimed at any point for this work to be my own.

Accessibility: Like originally planned, and outlined in the specifications, my website will have features to cater towards those with visual impairments. This is achieved by allowing users to increase the font size as necessary to improve readability, and the use of colour to help convey information. However, to additionally accomodate for users with certain forms of colour blindness, colour is never the sole means of communicating information. It is always backed up with text.

Cultural inclusivity: The software solution is intended to be by anybody across any background. The main consideration that needed to be made was the means and method of communication. While utilisation of multiple languages would be the optimal option to ensure cultural inclusivity, this is not viable with a small scale project like this. Instead I had to compromise and use English as it is one of the most universal languages due to its large speaker base worldwide. All the messages written in my site then had to be clear, consistent, unambiguous and objective manner to increase comprehensibility for more users.

Testing, Evaluating and Maintaining

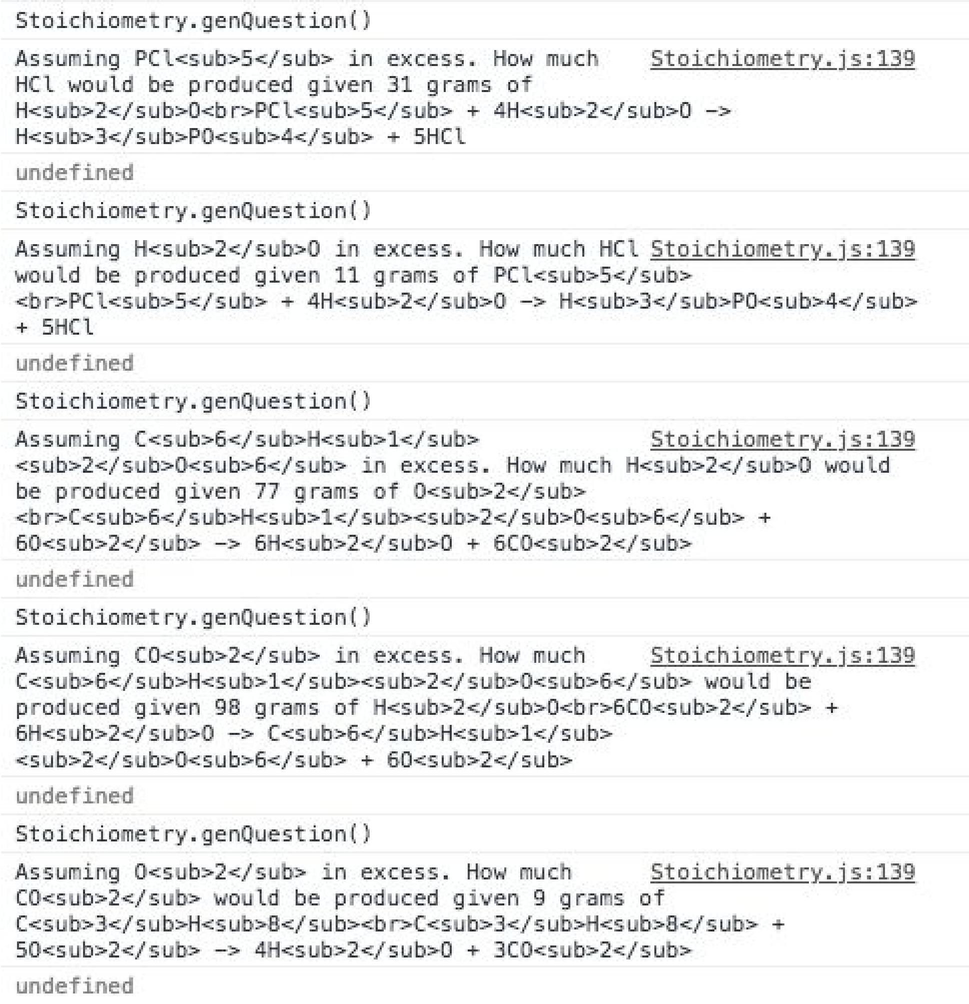
Test data summary and justification

As my software solution does not involve the user to manually type in or upload data, the main form of testing will involve accessing of functions.

Description	Justification
Mixed section use. Randomly use one section, then another.	It needs to be ensured that all modules in the program work together with no conflicts. By randomly moving from one section to another, we can check for any such conflicts. This also acts as a form of response time testing to see how fast one can move from one module to another

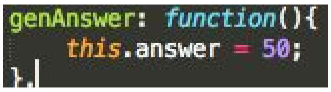
Module Testing:

Generation of questions: There were numerous times during the development process where the generation of questions led to runtime errors usually resulting from out of index errors, or referencing null element errors. Thus the question generation for each module needs to be tested to ensure that questions can be consistently generated with no errors. The image shown below shows the manual testing of the generating question module for the stoichiometry section by manually calling it in the Javascript console. There were no errors encountered during the testing of this module.

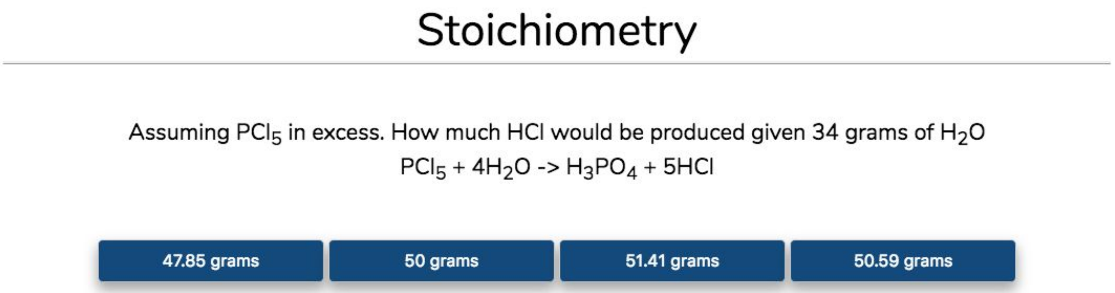


```
Stoichiometry.genQuestion()
Assuming PCl<sub>5</sub> in excess. How much    Stoichiometry.js:139
HCl would be produced given 31 grams of
H<sub>2</sub>O<br>PCl<sub>5</sub> + 4H<sub>2</sub>O ->
H<sub>3</sub>PO<sub>4</sub> + 5HCl
undefined
Stoichiometry.genQuestion()
Assuming H<sub>2</sub>O in excess. How much HCl Stoichiometry.js:139
would be produced given 11 grams of PCl<sub>5</sub>
<br>PCl<sub>5</sub> + 4H<sub>2</sub>O -> H<sub>3</sub>PO<sub>4</sub>
+ 5HCl
undefined
Stoichiometry.genQuestion()
Assuming C<sub>6</sub>H<sub>1</sub>             Stoichiometry.js:139
<sub>2</sub>O<sub>6</sub> in excess. How much H<sub>2</sub>O would
be produced given 77 grams of O<sub>2</sub>
<br>C<sub>6</sub>H<sub>1</sub><sub>2</sub>O<sub>6</sub> +
6O<sub>2</sub> -> 6H<sub>2</sub>O + 6CO<sub>2</sub>
undefined
Stoichiometry.genQuestion()
Assuming CO<sub>2</sub> in excess. How much     Stoichiometry.js:139
C<sub>6</sub>H<sub>1</sub><sub>2</sub>O<sub>6</sub> would be
produced given 98 grams of H<sub>2</sub>O<br>6CO<sub>2</sub> +
6H<sub>2</sub>O -> C<sub>6</sub>H<sub>1</sub>
<sub>2</sub>O<sub>6</sub> + 6O<sub>2</sub>
undefined
Stoichiometry.genQuestion()
Assuming O<sub>2</sub> in excess. How much      Stoichiometry.js:139
CO<sub>2</sub> would be produced given 9 grams of
C<sub>3</sub>H<sub>8</sub><br>C<sub>3</sub>H<sub>8</sub> +
5O<sub>2</sub> -> 4H<sub>2</sub>O + 3CO<sub>2</sub>
undefined
```

Randomising choices: For the multiple choices, the software is intended to randomly assign one of the four boxes to be the correct answers and the rest will be a randomly generated value that is between 90% and 110% of the true answer. To test this module in isolation, a stub was written for the generate answer function such that the answer is always 50. This allows us to see if the randomising choices module functions as intended.



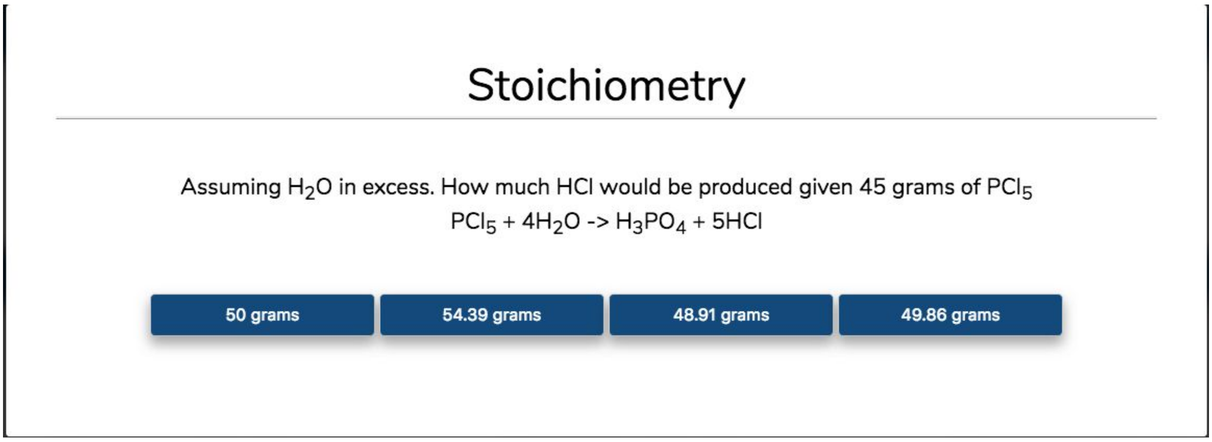
```
genAnswer: function(){
    this.answer = 50;
},
```



Stoichiometry

Assuming PCl_5 in excess. How much HCl would be produced given 34 grams of H_2O

$PCl_5 + 4H_2O \text{ -> } H_3PO_4 + 5HCl$



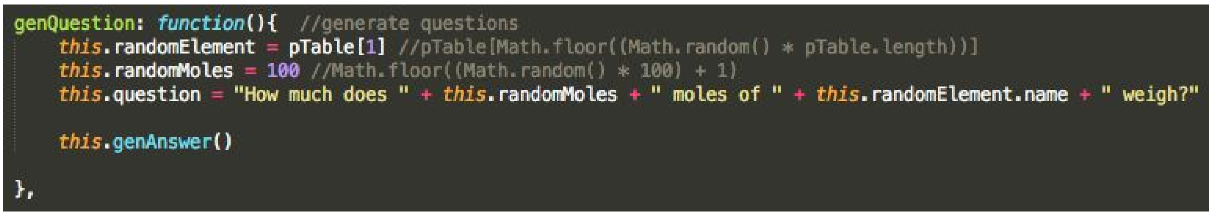
Stoichiometry

Assuming H_2O in excess. How much HCl would be produced given 45 grams of PCl_5

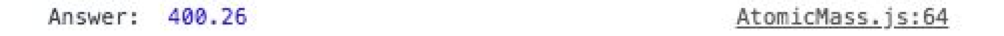
$PCl_5 + 4H_2O$ -> $H_3PO_4 + 5HCl$

In the two runs of testing for this module, it can be seen that the module functions properly in that it randomly assigns one of the boxes to be the correct answer. The other choices are also correctly within range from 90% (45) to 110% (55) of 50.

Generation of answers: Testing needs to be done to ensure that the answers generated are actually correct for the questions they are intended for. The program would be of little educational use if these were to be wrong. The generate question module was stubbed with numerous values to appropriately test the generate answer module's ability to generate answers.

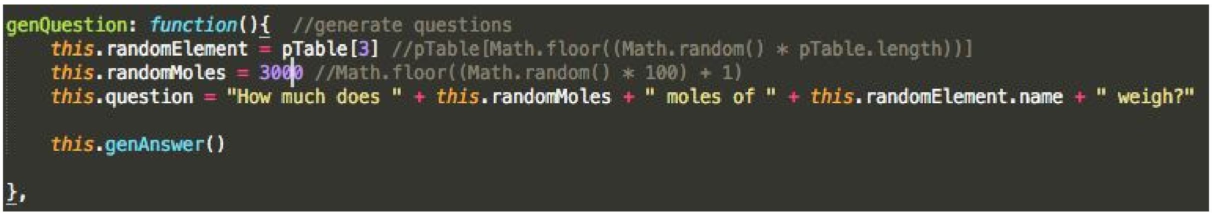


```
genQuestion: function(){  //generate questions
    this.randomElement = pTable[1] //pTable[Math.floor((Math.random() * pTable.length))]
    this.randomMoles = 100 //Math.floor((Math.random() * 100) + 1)
    this.question = "How much does " + this.randomMoles + " moles of " + this.randomElement.name + " weigh?"

    this.genAnswer()

},
```

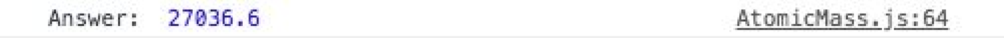


```
Answer:  400.26                                        AtomicMass.js:64
```

In this images above, the generate question module is stubbed such that the generated question is always "How much does 100 moles of helium weigh?". The expected answer for this should be 400.26 which is what the generate answer module outputted.



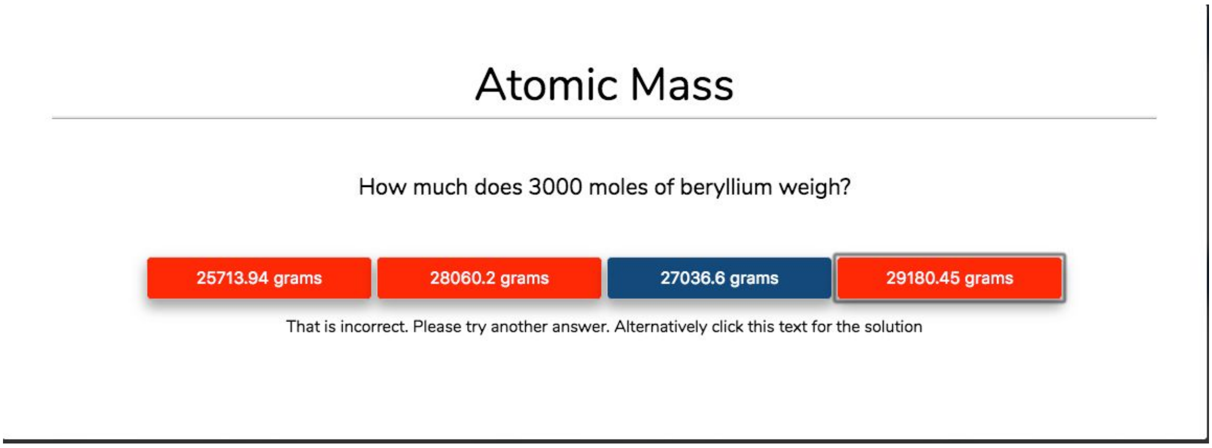
```
genQuestion: function(){  //generate questions
    this.randomElement = pTable[3] //pTable[Math.floor((Math.random() * pTable.length))]
    this.randomMoles = 3000 //Math.floor((Math.random() * 100) + 1)
    this.question = "How much does " + this.randomMoles + " moles of " + this.randomElement.name + " weigh?"

    this.genAnswer()

},
```



```
Answer:  27036.6                                       AtomicMass.js:64
```

The module was tested again, this time with the generating question module stubbed so the question is always "How much does 3000 moles of beryllium weigh?". The expected answer was 27036.6 which was what was outputted.

Choosing answers: Using the same stubbing approach from above, whereby the question is set and the answer is known. The choosing answers module can be tested by choosing both the correct and incorrect choices.

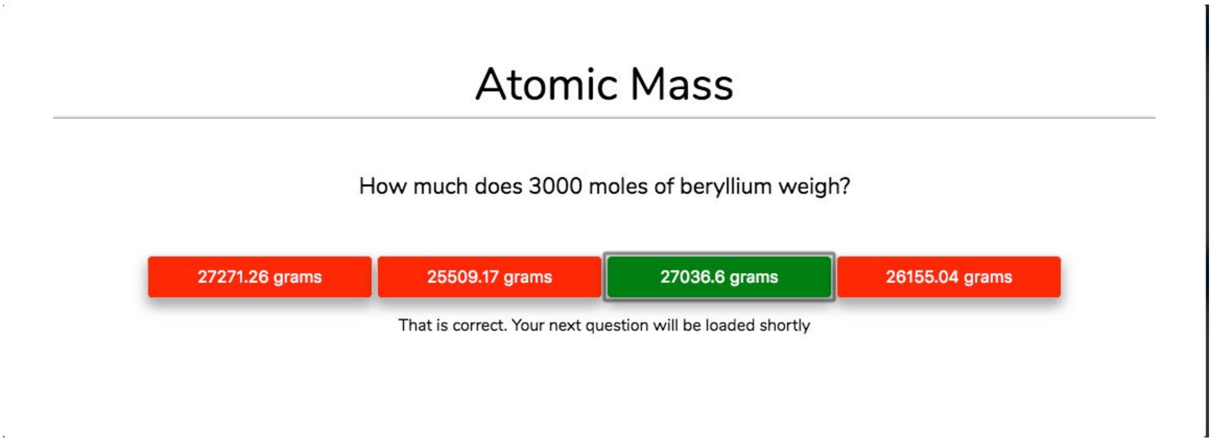


Atomic Mass

How much does 3000 moles of beryllium weigh?

That is incorrect. Please try another answer. Alternatively click this text for the solution

27036.6 is the correct answer here, and as seen from the image above, choosing the other choices results in their buttons turning red and the incorrect message to pop up.



Atomic Mass

How much does 3000 moles of beryllium weigh?

That is correct. Your next question will be loaded shortly

Clicking on the correct answers then results in the button to turn green, with the correct message being displayed. After a certain wait time, the next question is then loaded. This module works as intended with no errors.

System Testing

Testing of the combined operation of modules: While during module testing, the generating questions and generating answers modules were tested in isolation, here it is to be tested as a combined process. No stubs will be used, instead the generation of questions is to be randomised as originally intended, and then the generation of answers will generate the answer for the given question. The expected answers will be calculated manually. The actual answer is the option that turns green upon clicking.

Atomic Mass

Question	Expected Answer	Actual Answer
How much does 73 moles of oxygen weigh?	1167.93	1167.93

Electrochemistry

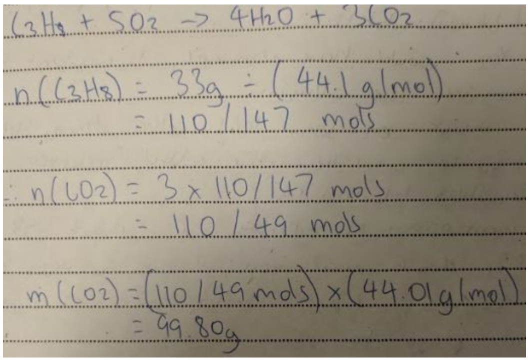
Question	Expected Answer	Actual Answer

What is the voltage of the following galvanic cell: Na | Na^{+1} || H^{+1} | H ?	2.71	2.71

Half Life

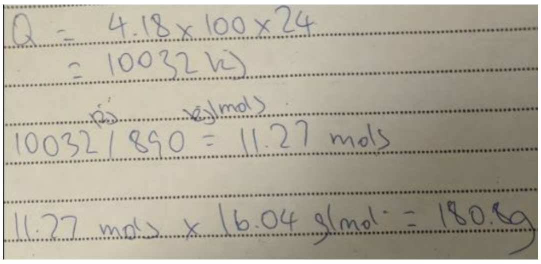
Question	Expected Answer	Actual Answer
The half-life of 79 grams of a radioisotope is 93 years, how many grams will remain in 27 years?	64.6	64.6

Stoichiometry

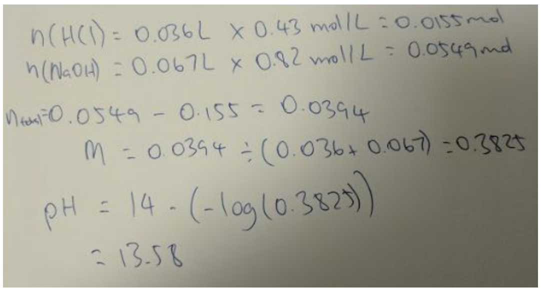
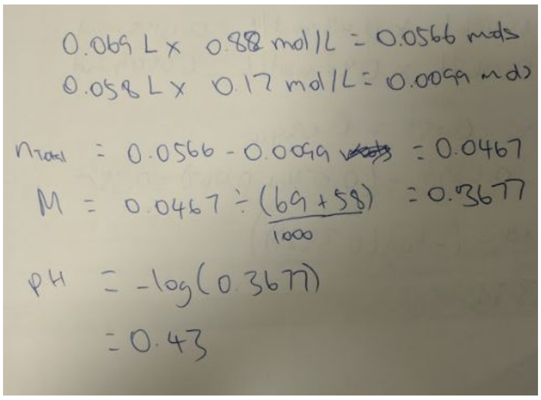
Question	Expected Answer	Actual Answer
Assuming O_2 in excess. How much CO_2 would be produced given 33 grams of C_3H_8 $C_3H_8 + 5O_2 \rightarrow 4H_2O + 3CO_2$	99.80  $C_3H_8 + 5O_2 \rightarrow 4H_2O + 3CO_2$ $n(C_3H_8) = 33g \div (44.1 g/mol)$ $= 110/147$ mols $\therefore n(CO_2) = 3 \times 110/147$ mols $= 110/49$ mols $m(CO_2) = (110/49 \text{ mols}) \times (44.01 g/mol)$ $= 99.80g$	99.02

*There are errors countered with precision here. The rounding methods used have since been updated to circumvent this.

Energy

Question	Expected Answer	Actual Answer
Given methane has a heat of combustion of 890 kj/mol. What mass is required to heat up 100 L of water by 24K?	180.8  $Q = 4.18 \times 100 \times 24$ $= 10032$ kJ $\overset{kJ}{10032} / \overset{kJ/mols}{890} = 11.27$ mols 11.27 mols $\times 16.04$ g/mol $= 180.8g$	180.8

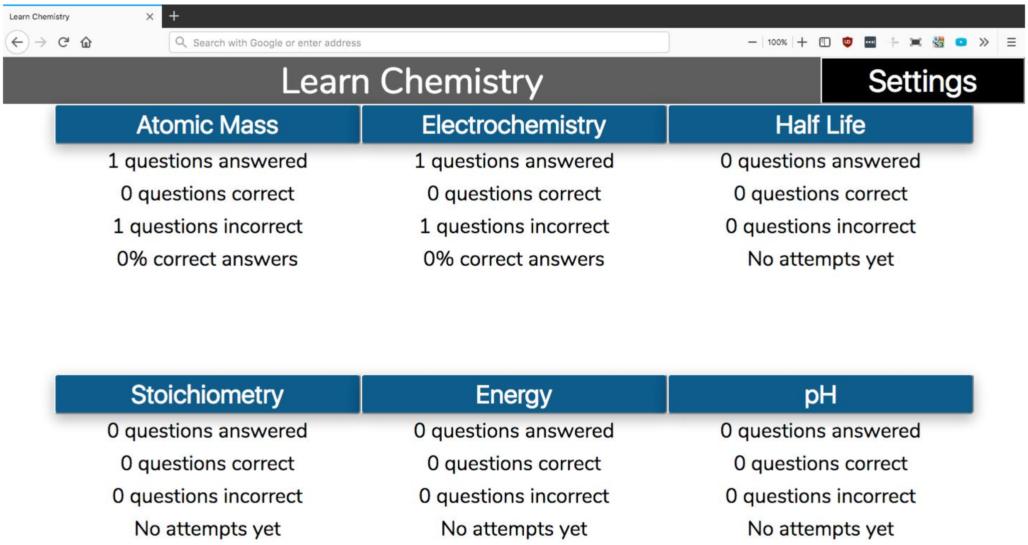
pH

Question	Expected Answer	Actual Answer
36 mL of 0.43 mol/L HCl is mixed with 67 mL of 0.82 mol/L NaOH. What is the resulting pH?	13.58  n(HCl) = 0.036L × 0.43 mol/L = 0.0155 mol n(NaOH) = 0.067L × 0.82 mol/L = 0.0549 mol n_{total} = 0.0549 − 0.155 = 0.0394 M = 0.0394 ÷ (0.036 + 0.067) = 0.3825 pH = 14 − (−log(0.3825)) = 13.58	12.75
69 mL of 0.82 mol/L HCl is mixed with 58 mL of 0.17 mol/L NaOH. What is the resulting pH?	0.43  0.069 L × 0.88 mol/L = 0.0566 mols 0.058 L × 0.17 mol/L = 0.0099 mols n_{Total} = 0.0566 − 0.0099 = 0.0467 M = 0.0467 ÷ $\frac{(69 + 58)}{1000}$ = 0.3677 pH = −log(0.3677) = 0.43	0.43

*The inaccuracy here was cause by the incorrect conversion of units within my program. This has since been fixed as evident from the second test.

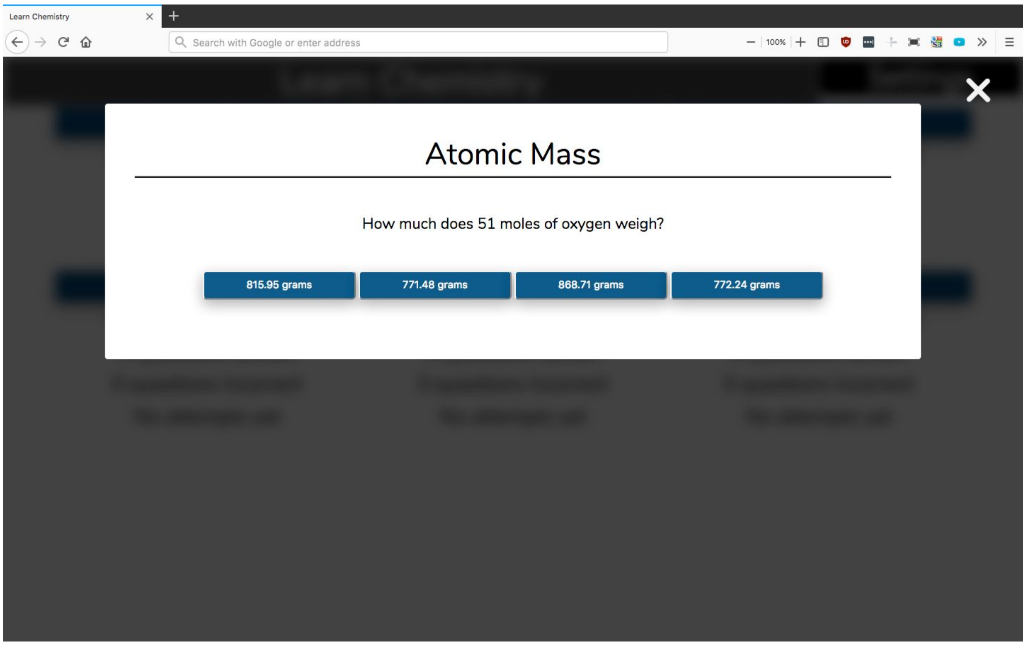
Testing on different browsers: Due to the entire software solution being developed on Google Chrome, it must be tested on other browsers to ensure compatibility for as many users as possible.

The two images below show my software solution working within Mozilla Firefox. While the user interface and design seems to differ compared to when displayed on Google Chrome, all functionality of my website still works as intended



Atomic Mass	Electrochemistry	Half Life
1 questions answered	1 questions answered	0 questions answered
0 questions correct	0 questions correct	0 questions correct
1 questions incorrect	1 questions incorrect	0 questions incorrect
0% correct answers	0% correct answers	No attempts yet

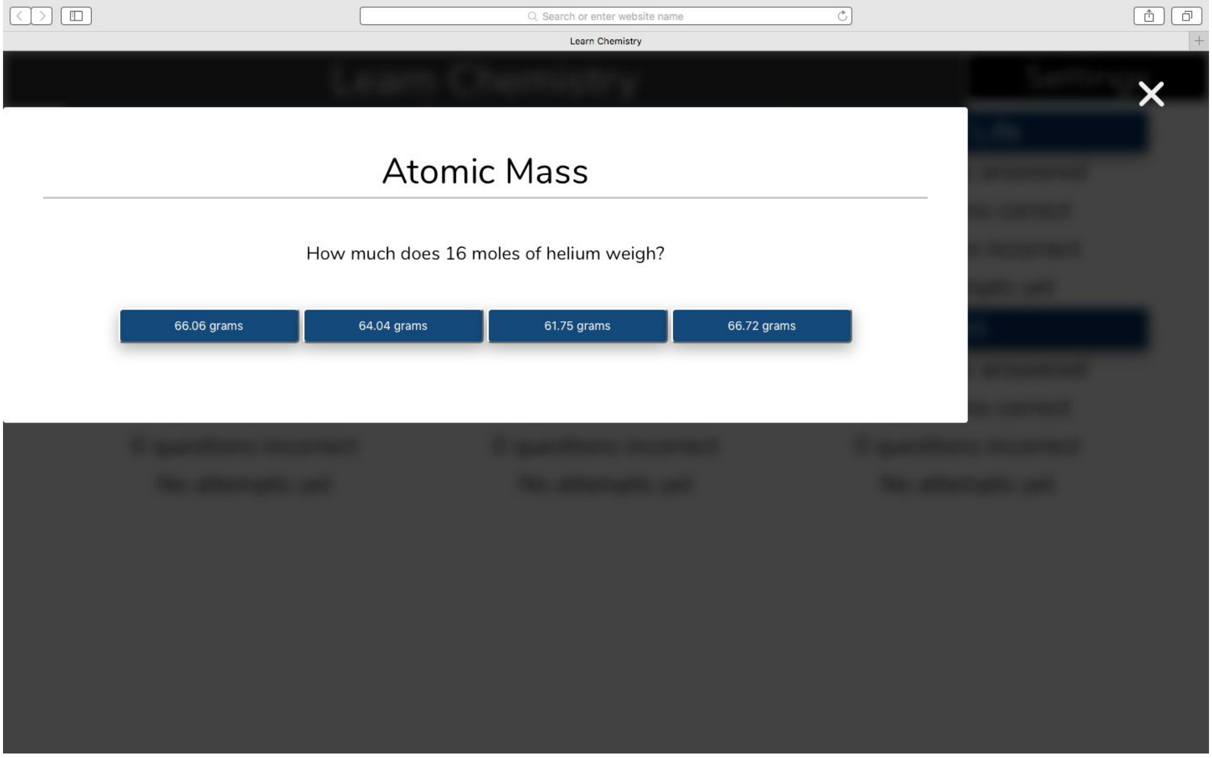
Stoichiometry	Energy	pH
0 questions answered	0 questions answered	0 questions answered
0 questions correct	0 questions correct	0 questions correct
0 questions incorrect	0 questions incorrect	0 questions incorrect
No attempts yet	No attempts yet	No attempts yet



Atomic Mass

How much does 51 moles of oxygen weigh?

While on Safari all basic functionality works, it has significant issues with the alignment of the modals. Furthermore many of the styling effects such as shadows don't apply. This was intended to be fixed, but ultimately the source of the error cannot be pinned down.



Atomic Mass

How much does 16 moles of helium weigh?

Modifications in response to testing

Text input for answers: While text input can prevent users from randomly guessing answers, it also introduces issues with usability and ergonomics. The software solution is designed to work on a variety of hardware, having text input can be highly inconvenient for those on touch devices.

Online progression tracking: It would be useful to have the progress within each section saved within some form of online account system. However this system is difficult is simply too difficult to implement this late into development. Such a system would require the entire software to be designed to work with it from the start.

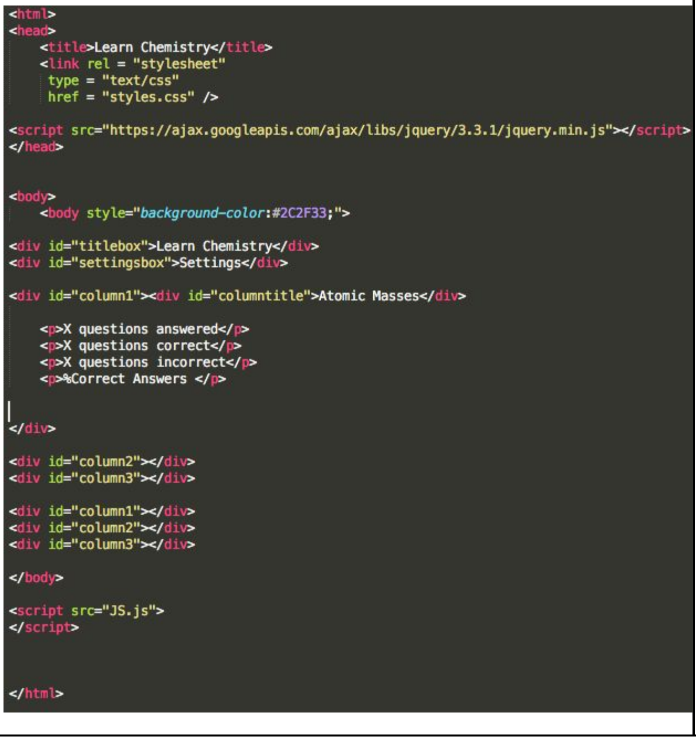
Evaluation of final solution

Overall I am pleased with the software solution I have created. All the requirements and specifications outlined at the start of the development process have been achieved. The software solution succeeds in the generation of questions relating to the mathematical aspects of chemistry. It also succeeds in the generation of solutions specific to each question.

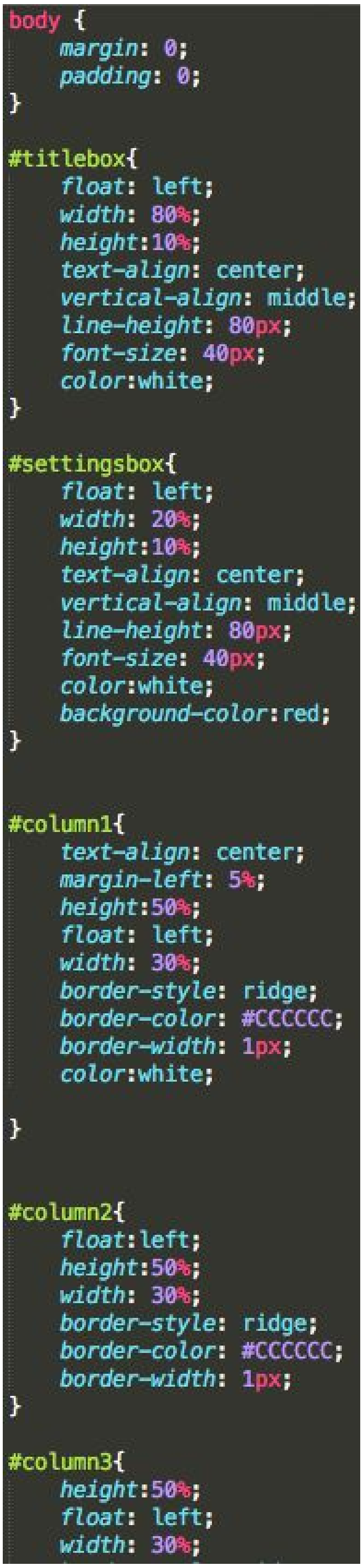
While achieving everything I originally set out to do, I feel like certain aspects could have been made better. For example, in the stoichiometry section, while programmed to generate questions, it still requires chemical equations be provided in. A much more powerful solution would involve the generation of chemical equations from elements on the periodic table. However this requires a much deeper understanding of chemistry and mathematics that I currently do not possess. There are billions of possible combinations of elements, and it is highly difficult to predict which elements can actually react together in a chemical reaction. Furthermore even after a possible equation has been generated, it still needs to be balanced. Chemical equations are linear systems of equation, and while generally possible to balance using human intuition, the algorithms a computer requires to balance them often involve high level mathematics.

Another aspect I felt like could have been improved on is the user interface across a variety of different screen sizes and resolutions. While the software itself is efficient enough to perform optimally on most modern hardware, it is ultimately limited by the user interface. User interface is severely hindered on smaller screened devices such as mobile phones, and certain tablets. Had extra time been given, I would have liked to implement a responsive user interface that changes according to user devices.

Logbook

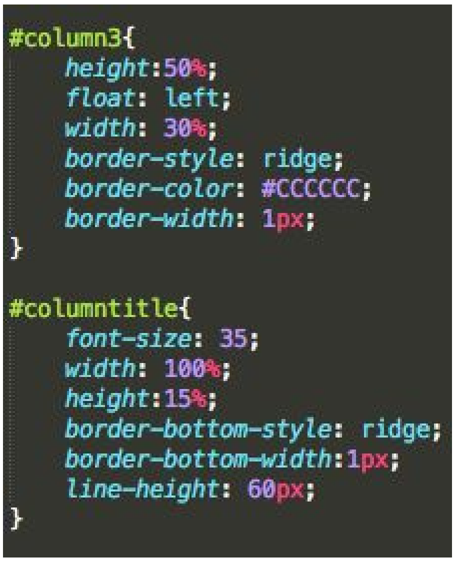
Date	Image/Screenshot	Accomplished/Problems
6/5		Created basic structuring of the overall design of the web application.

```html
<html>
<head>
    <title>Learn Chemistry</title>
    <link rel = "stylesheet"
     type = "text/css"
     href = "styles.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
</head>


<body>
    <body style="background-color:#2C2F33;">

<div id="titlebox">Learn Chemistry</div>
<div id="settingsbox">Settings</div>

<div id="column1"><div id="columntitle">Atomic Masses</div>

    <p>X questions answered</p>
    <p>X questions correct</p>
    <p>X questions incorrect</p>
    <p>%Correct Answers </p>


</div>

<div id="column2"></div>
<div id="column3"></div>

<div id="column1"></div>
<div id="column2"></div>
<div id="column3"></div>

</body>

<script src="JS.js">
</script>



</html>
```



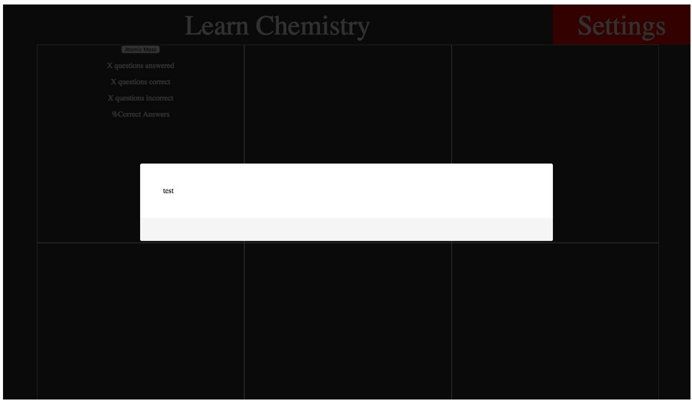
```
body {
    margin: 0;
    padding: 0;
}

#titlebox{
    float: left;
    width: 80%;
    height:10%;
    text-align: center;
    vertical-align: middle;
    line-height: 80px;
    font-size: 40px;
    color:white;
}

#settingsbox{
    float: left;
    width: 20%;
    height:10%;
    text-align: center;
    vertical-align: middle;
    line-height: 80px;
    font-size: 40px;
    color:white;
    background-color:red;
}


#column1{
    text-align: center;
    margin-left: 5%;
    height:50%;
    float: left;
    width: 30%;
    border-style: ridge;
    border-color: #CCCCCC;
    border-width: 1px;
    color:white;

}


#column2{
    float:left;
    height:50%;
    width: 30%;
    border-style: ridge;
    border-color: #CCCCCC;
    border-width: 1px;
}

#column3{
    height:50%;
    float: left;
    width: 30%;
```

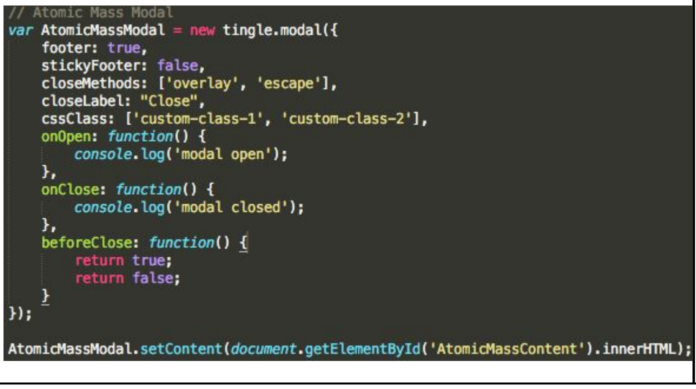
<table>
<tr>
<td></td>
<td>



```css
#column3{
    height:50%;
    float: left;
    width: 30%;
    border-style: ridge;
    border-color: #CCCCCC;
    border-width: 1px;
}

#columntitle{
    font-size: 35;
    width: 100%;
    height:15%;
    border-bottom-style: ridge;
    border-bottom-width:1px;
    line-height: 60px;
}
```

</td>
<td></td>
</tr>
<tr>
<td>8/5</td>
<td></td>
<td>Issues with UI on different screen sizes and when resizing window. Caused by usage of % for CSS styling. Changed to use Viewpoint values instead.</td>
</tr>
<tr>
<td>12/5</td>
<td>



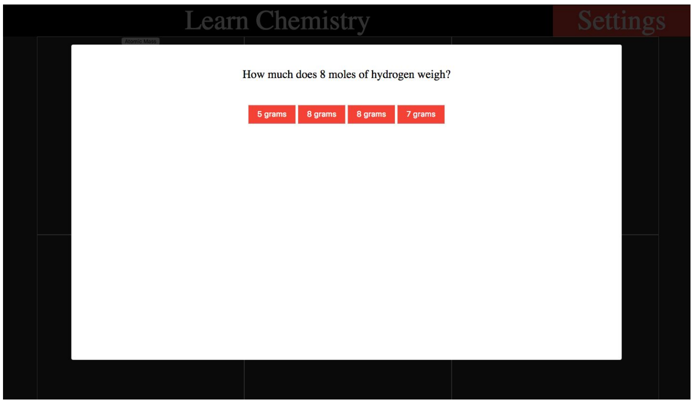


```javascript
// Atomic Mass Modal
var AtomicMassModal = new tingle.modal({
    footer: true,
    stickyFooter: false,
    closeMethods: ['overlay', 'escape'],
    closeLabel: "Close",
    cssClass: ['custom-class-1', 'custom-class-2'],
    onOpen: function() {
        console.log('modal open');
    },
    onClose: function() {
        console.log('modal closed');
    },
    beforeClose: function() {
        return true;
        return false;
    }
});

AtomicMassModal.setContent(document.getElementById('AtomicMassContent').innerHTML);
```

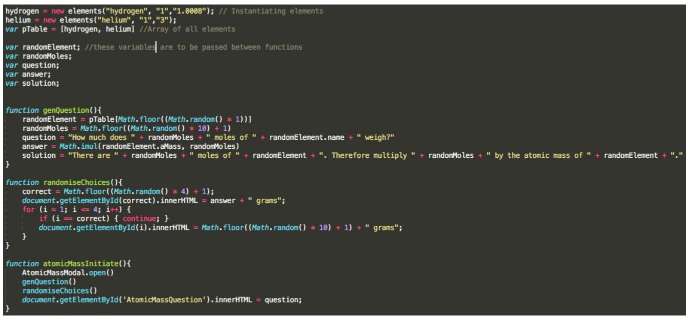
</td>
<td>Through the use of the Tingle plugin, implemented modals. Currently only using the default layout through the stock code they provided, to be updated and stylised after functionality done.

Tingle sets the content of the modal through Javascript. Initially ran into a problem where I didn't know how to implement HTML elements into this modal such as with buttons or text input. Realised can set content with document.getElementById</td>
</tr>
</table>

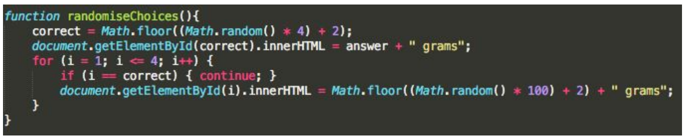
915/5



How much does 8 moles of hydrogen weigh?



```
hydrogen = new elements("hydrogen", "1","1.0008"); // Instantiating elements
helium = new elements("helium", "1","3");
var pTable = [hydrogen, helium] //Array of all elements

var randomElement; //these variables are to be passed between functions
var randomMoles;
var question;
var answer;
var solution;


function genQuestion(){
    randomElement = pTable[Math.floor((Math.random() * 1))]
    randomMoles = Math.floor((Math.random() * 10) + 1)
    question = "How much does " + randomMoles + " moles of " + randomElement.name + " weigh?"
    answer = Math.imul(randomElement.aMass, randomMoles)
    solution = "There are " + randomMoles + " moles of " + randomElement + ". Therefore multiply " + randomMoles + " by the atomic mass of " + randomElement + "."
}

function randomiseChoices(){
    correct = Math.floor((Math.random() * 4) + 1);
    document.getElementById(correct).innerHTML = answer + " grams";
    for (i = 1; i <= 4; i++) {
        if (i == correct) { continue; }
        document.getElementById(i).innerHTML = Math.floor((Math.random() * 10) + 1) + " grams";
    }
}

function atomicMassInitiate(){
    AtomicMassModal.open()
    genQuestion()
    randomiseChoices()
    document.getElementById('AtomicMassQuestion').innerHTML = question;
}
```



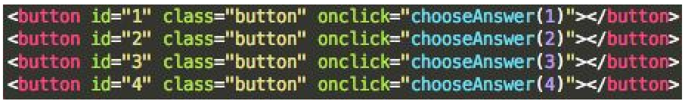
```
function randomiseChoices(){
    correct = Math.floor((Math.random() * 4) + 2);
    document.getElementById(correct).innerHTML = answer + " grams";
    for (i = 1; i <= 4; i++) {
        if (i == correct) { continue; }
        document.getElementById(i).innerHTML = Math.floor((Math.random() * 100) + 2) + " grams";
    }
}
```

Implemented the generation of random questions for the “Atomic Mass” section. Also setup a function to randomly select one answer choice as correct, and then randomly generate the others.

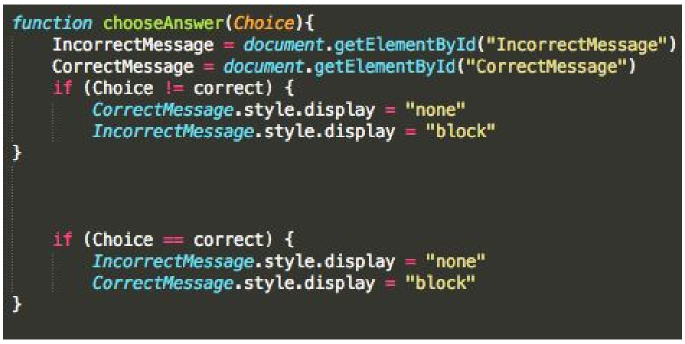
Created the AtomicMassInitiate function that will load the said content above into the modal.

Currently there exists an issue where duplicate choices are generated. Will use the standard algorithm to generate unique random numbers in order to avoid this issue

16/5



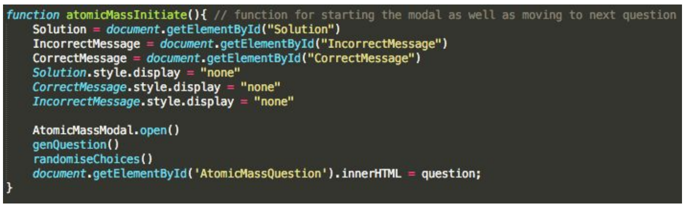
```
<button id="1" class="button" onclick="chooseAnswer(1)"></button>
<button id="2" class="button" onclick="chooseAnswer(2)"></button>
<button id="3" class="button" onclick="chooseAnswer(3)"></button>
<button id="4" class="button" onclick="chooseAnswer(4)"></button>
```



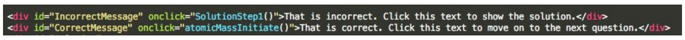
```
function chooseAnswer(Choice){
    IncorrectMessage = document.getElementById("IncorrectMessage")
    CorrectMessage = document.getElementById("CorrectMessage")
    if (Choice != correct) {
        CorrectMessage.style.display = "none"
        IncorrectMessage.style.display = "block"
}



    if (Choice == correct) {
        IncorrectMessage.style.display = "none"
        CorrectMessage.style.display = "block"
}
```

Made each of the buttons on the question screen initiate the chooseAnswer(Choice) function, while passing a different variable each. This allows the checking of the question to see if it matches the correct answer determined by the randomise choices function. Also displays a message on the modal to show if answered correctly or not.

17/5



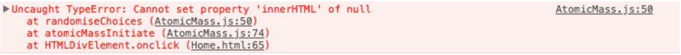
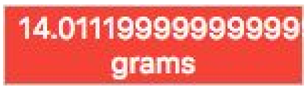
```
function atomicMassInitiate(){ // function for starting the modal as well as moving to next question
    Solution = document.getElementById("Solution")
    IncorrectMessage = document.getElementById("IncorrectMessage")
    CorrectMessage = document.getElementById("CorrectMessage")
    Solution.style.display = "none"
    CorrectMessage.style.display = "none"
    IncorrectMessage.style.display = "none"

    AtomicMassModal.open()
    genQuestion()
    randomiseChoices()
    document.getElementById('AtomicMassQuestion').innerHTML = question;
}
```

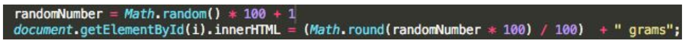
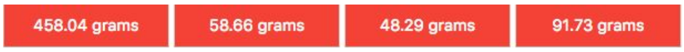
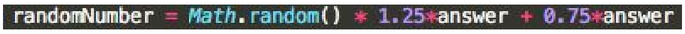


```
<div id="IncorrectMessage" onclick="SolutionStep1()">That is incorrect. Click this text to show the solution.</div>
<div id="CorrectMessage" onclick="atomicMassInitiate()">That is correct. Click this text to move on to the next question.</div>
```

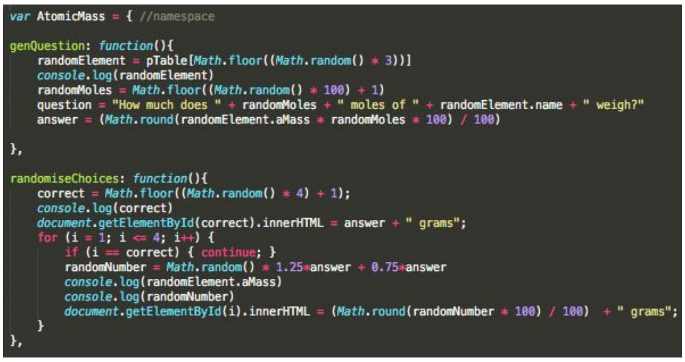
Updated the AtomicMassInitiate function to also update the modal with progression from question to question.

Progression from question to question is implemented, with buttons that trigger the AtomicMassInitiate function.

19/5	
	`▶Uncaught TypeError: Cannot set property 'innerHTML' of null` `AtomicMass.js:50`
	`    at randomiseChoices (AtomicMass.js:50)`
	`    at atomicMassInitiate (AtomicMass.js:74)`
	`    at HTMLDivElement.onclick (Home.html:65)`
	
	14.0111999999999 grams
		Ran into problem where after a certain number of questions answered, functions to generate next question cease to work. Discovered this was an error in the RNG within the randomiseChoices function. The HTML buttons for the choices were numbered 1 to 4, but the RNG was generating numbers greater than this range causing it to access a null HTML element.
		
		Originally used integers for atomic masses to simplify the problem. Now converted to floats, however this causes errors with trailing digits due to the nature of floating point arithmetic. Solved this through rounding method: Math.round(num * 100) / 100. Multiply the number by 100, then round the result to nearest integer, then divide by 100.

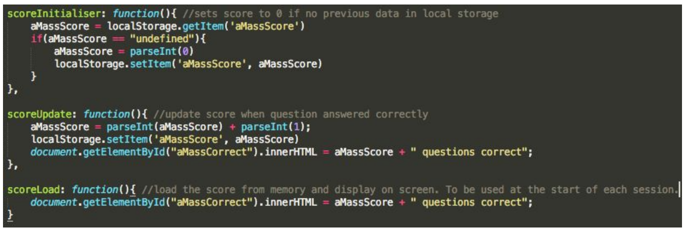
20/5	
	`randomNumber = Math.random() * 100 + 1`
	`document.getElementById(i).innerHTML = (Math.round(randomNumber * 100) / 100)  + " grams";`
	
	458.04 grams 58.66 grams 48.29 grams 91.73 grams
	
	`randomNumber = Math.random() * 1.25*answer + 0.75*answer`
		The incorrect other choices are generated purely by a random number within a set range. This creates choices that are unreasonable, and the correct choice is obviously apparent. Adjusted the random number generation to be between 75% and 125% of the actual answer to produce reasonable choices.

<table>
<tr>
<td>23/5</td>
<td>



```
var AtomicMass = { //namespace

genQuestion: function(){
    randomElement = pTable[Math.floor((Math.random() * 3))]
    console.log(randomElement)
    randomMoles = Math.floor((Math.random() * 100) + 1)
    question = "How much does " + randomMoles + " moles of " + randomElement.name + " weigh?"
    answer = (Math.round(randomElement.aMass * randomMoles * 100) / 100)

},

randomiseChoices: function(){
    correct = Math.floor((Math.random() * 4) + 1);
    console.log(correct)
    document.getElementById(correct).innerHTML = answer + " grams";
    for (i = 1; i <= 4; i++) {
        if (i == correct) { continue; }
        randomNumber = Math.random() * 1.25*answer + 0.75*answer
        console.log(randomElement.aMass)
        console.log(randomNumber)
        document.getElementById(i).innerHTML = (Math.round(randomNumber * 100) / 100)  + " grams";
    }
},
```

</td>
<td>Previously used variables and functions for the “AtomicMass” section in the global scope. Realised however this may cause issues such as variable or function conflicts when developing the other sections. Created a dedicated namespace for the section as a means to encapsulate related data and functions. Will also intend to do this for all other sections.</td>
</tr>
<tr>
<td>25/5</td>
<td>

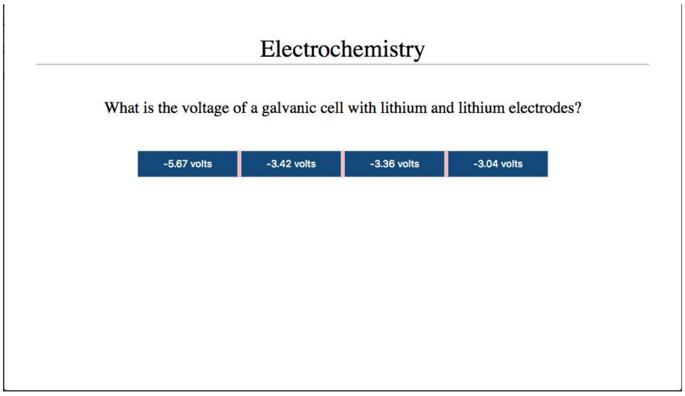


```
scoreInitialiser: function(){ //sets score to 0 if no previous data in local storage
    aMassScore = localStorage.getItem('aMassScore')
    if(aMassScore == "undefined"){
        aMassScore = parseInt(0)
        localStorage.setItem('aMassScore', aMassScore)
    }
},

scoreUpdate: function(){ //update score when question answered correctly
    aMassScore = parseInt(aMassScore) + parseInt(1);
    localStorage.setItem('aMassScore', aMassScore)
    document.getElementById("aMassCorrect").innerHTML = aMassScore + " questions correct";
},

scoreLoad: function(){ //load the score from memory and display on screen. To be used at the start of each session.
    document.getElementById("aMassCorrect").innerHTML = aMassScore + " questions correct";
}
```

</td>
<td>Implemented basic score tracking by using the browser’s localstorage.

Used a temporary boolean value that determines if the question was answered on the first go or not to accurately calculate the score.

There is currently a problem where continually pressing the answer choices continually update the scores when it is not supposed to.</td>
</tr>
<tr>
<td>27/5</td>
<td>

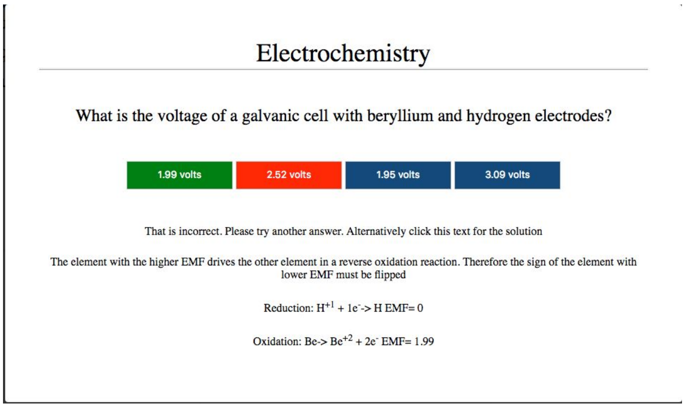


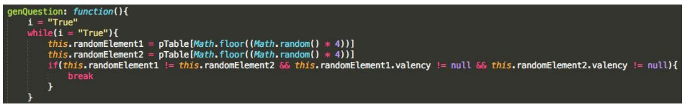
Electrochemistry

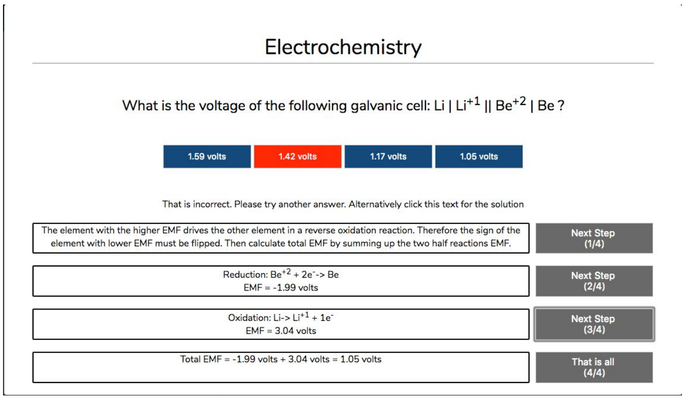
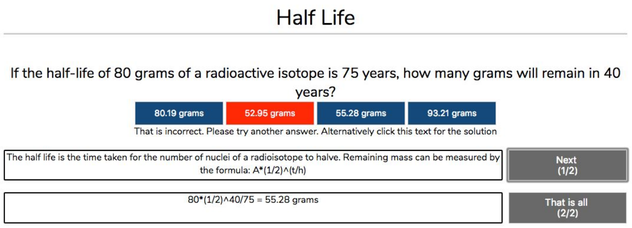
What is the voltage of a galvanic cell with lithium and lithium electrodes?

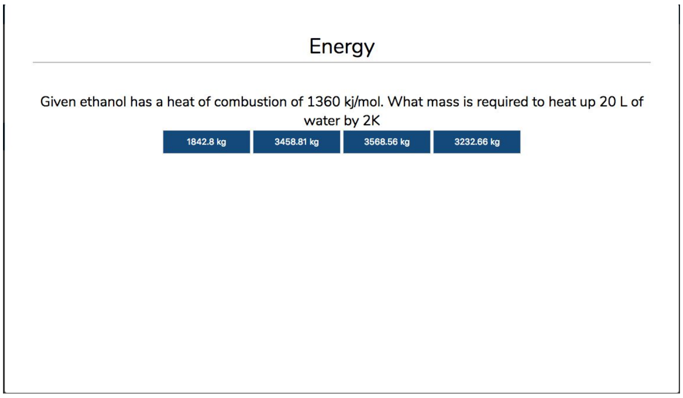
</td>
<td>Focused on creating other modules, to ensure the program runs as a whole with no conflicts between different sections.

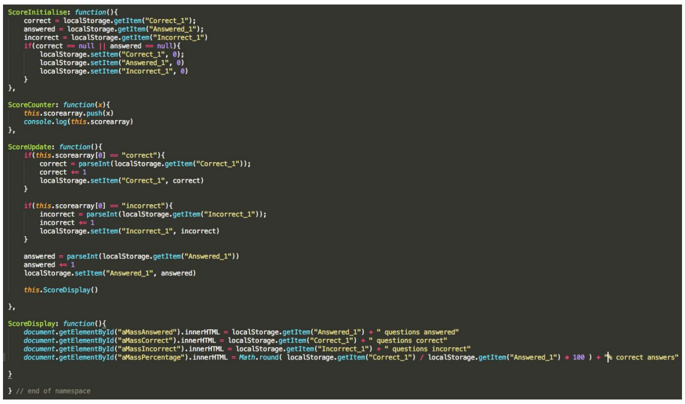
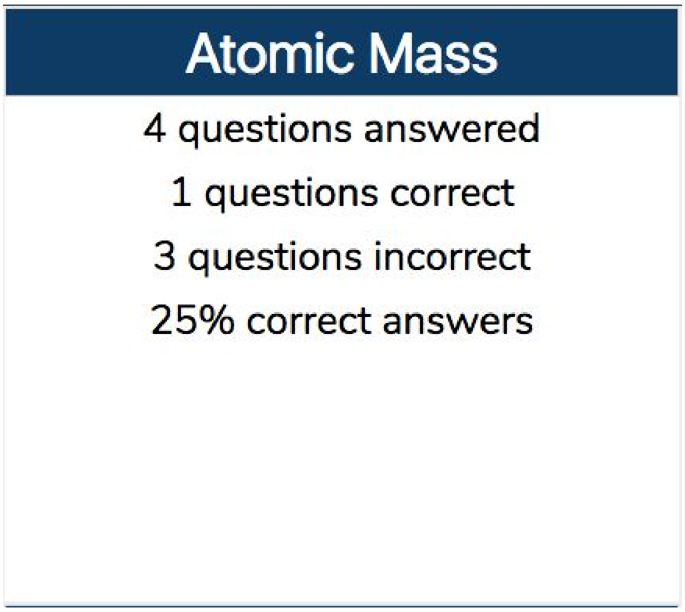
Developed the basic generation of electrochemistry solutions as well as solutions</td>
</tr>
</table>

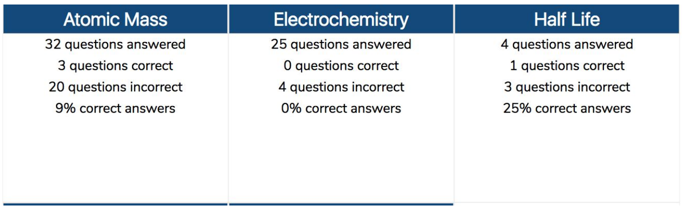
	 Electrochemistry What is the voltage of a galvanic cell with beryllium and hydrogen electrodes? 1.99 volts 2.52 volts 1.95 volts 3.09 volts That is incorrect. Please try another answer. Alternatively click this text for the solution The element with the higher EMF drives the other element in a reverse oxidation reaction. Therefore the sign of the element with lower EMF must be flipped Reduction: H^{+1} + $1e^{-}$-> H EMF= 0 Oxidation: Be-> Be^{+2} + $2e^{-}$ EMF= 1.99	

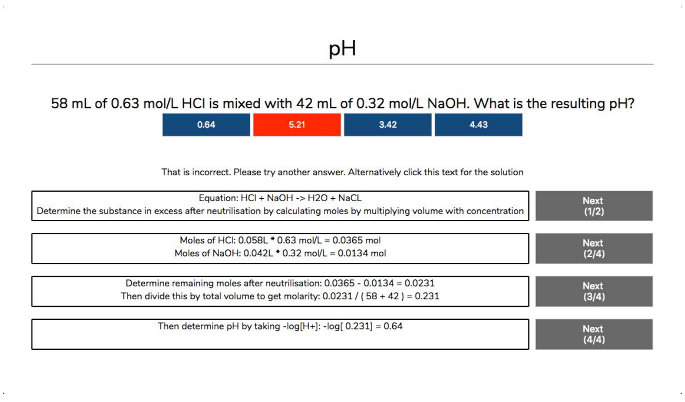
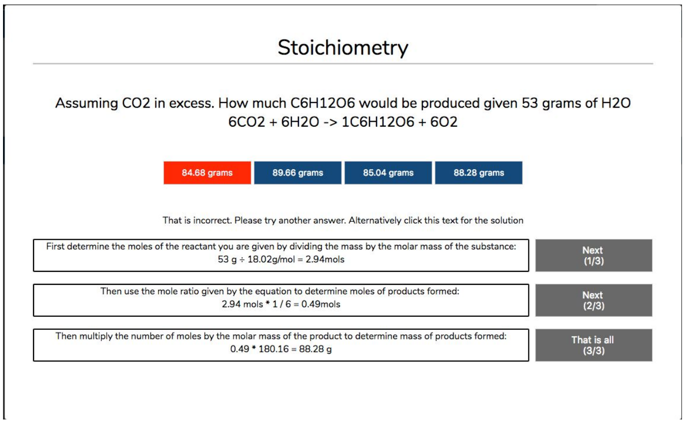
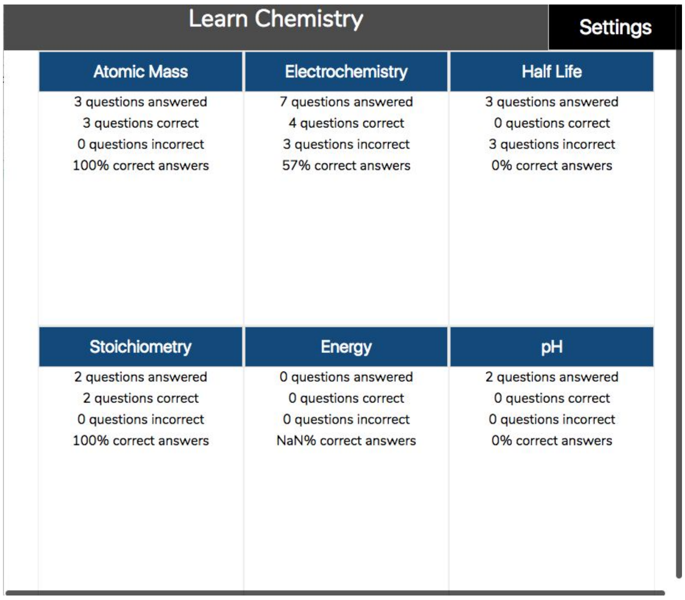
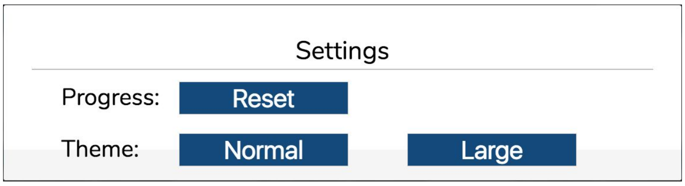
28/5	
	`genQuestion: function(){`
	`    i = "True"`
	`    while(i = "True"){`
	`        this.randomElement1 = pTable[Math.floor((Math.random() * 4))]`
	`        this.randomElement2 = pTable[Math.floor((Math.random() * 4))]`
	`        if(this.randomElement1 != this.randomElement2 && this.randomElement1.valency != null && this.randomElement2.valency != null){`
	`            break`
	`        }`
	`    }`
		Noticed a problem with the generation of questions with same electrodes. Set extra conditions to avoid this.

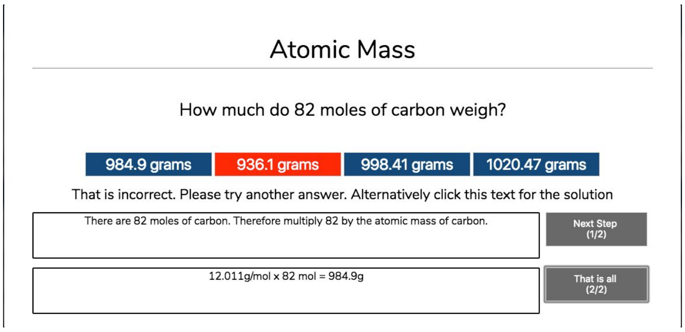
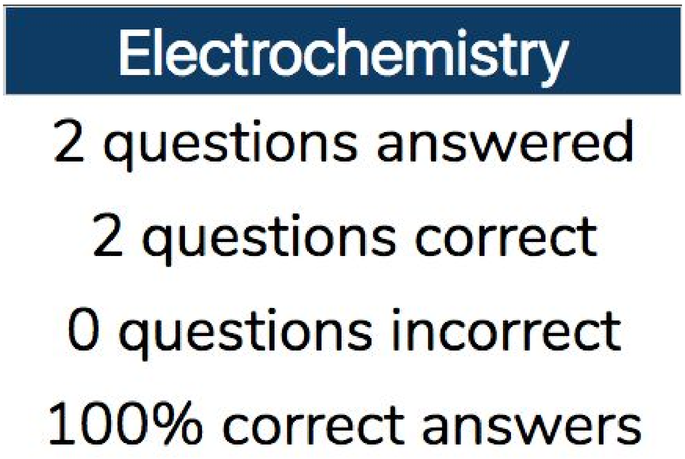
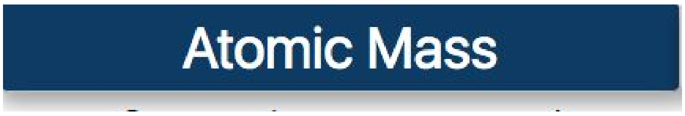
30/5	 Electrochemistry What is the voltage of the following galvanic cell: Li | Li^{+1} || Be^{+2} | Be ? 1.59 volts 1.42 volts 1.17 volts 1.05 volts That is incorrect. Please try another answer. Alternatively click this text for the solution The element with the higher EMF drives the other element in a reverse oxidation reaction. Therefore the sign of the element with lower EMF must be flipped. Then calculate total EMF by summing up the two half reactions EMF. Next Step (1/4) Reduction: Be^{+2} + $2e^{-}$-> Be EMF = -1.99 volts Next Step (2/4) Oxidation: Li-> Li^{+1} + $1e^{-}$ EMF = 3.04 volts Next Step (3/4) Total EMF = -1.99 volts + 3.04 volts = 1.05 volts That is all (4/4)	Created a more intuitive user interface for moving from step to step of solution
1/6	 Half Life If the half-life of 80 grams of a radioactive isotope is 75 years, how many grams will remain in 40 years? 80.19 grams 52.95 grams 55.28 grams 93.21 grams That is incorrect. Please try another answer. Alternatively click this text for the solution The half life is the time taken for the number of nuclei of a radioisotope to halve. Remaining mass can be measured by the formula: A*(1/2)^(t/h) Next (1/2) 80*(1/2)^40/75 = 55.28 grams That is all (2/2)	Developed half life section. Due to preexisting code from the other sections, was able to easily create this new section.

3/6	 Energy Given ethanol has a heat of combustion of 1360 kj/mol. What mass is required to heat up 20 L of water by 2K 1842.8 kg 3458.81 kg 3568.56 kg 3232.66 kg	Started developing energy section. Encountered issues with incorrect generation of answers.
4/6		Finished energy section

6/6	
	`ScoreInitialise: function(){`
	`    correct = localStorage.getItem("Correct_1");`
	`    answered = localStorage.getItem("Answered_1");`
	`    incorrect = localStorage.getItem("Incorrect_1")`
	`    if(correct == null || answered == null){`
	`        localStorage.setItem("Correct_1", 0);`
	`        localStorage.setItem("Answered_1", 0)`
	`        localStorage.setItem("Incorrect_1", 0)`
	`    }`
	`},`
	
	`ScoreCounter: function(x){`
	`    this.scorearray.push(x)`
	`    console.log(this.scorearray)`
	`},`
	
	`ScoreUpdate: function(){`
	`    if(this.scorearray[0] == "correct"){`
	`        correct = parseInt(localStorage.getItem("Correct_1"));`
	`        correct += 1`
	`        localStorage.setItem("Correct_1", correct)`
	`    }`
	
	`    if(this.scorearray[0] == "incorrect"){`
	`        incorrect = parseInt(localStorage.getItem("Incorrect_1"));`
	`        incorrect += 1`
	`        localStorage.setItem("Incorrect_1", incorrect)`
	`    }`
	
	`    answered = parseInt(localStorage.getItem("Answered_1"))`
	`    answered += 1`
	`    localStorage.setItem("Answered_1", answered)`
	
	`    this.ScoreDisplay()`
	
	`},`
	
	`ScoreDisplay: function(){`
	`    document.getElementById("aMassAnswered").innerHTML = localStorage.getItem("Answered_1") + " questions answered"`
	`    document.getElementById("aMassCorrect").innerHTML = localStorage.getItem("Correct_1") + " questions correct"`
	`    document.getElementById("aMassIncorrect").innerHTML = localStorage.getItem("Incorrect_1") + " questions incorrect"`
	`    document.getElementById("aMassPercentage").innerHTML = Math.round( localStorage.getItem("Correct_1") / localStorage.getItem("Answered_1") * 100 ) + "% correct answers"`
	
	`}`
	
	`} // end of namespace`
	
	Atomic Mass
	4 questions answered
	1 questions correct
	3 questions incorrect
	25% correct answers
		Further development on score tracking functions.

7/6	 Atomic Mass 32 questions answered 3 questions correct 20 questions incorrect 9% correct answers Electrochemistry 25 questions answered 0 questions correct 4 questions incorrect 0% correct answers Half Life 4 questions answered 1 questions correct 3 questions incorrect 25% correct answers	Finished score tracking functions for all modules

10/6	 pH 58 mL of 0.63 mol/L HCl is mixed with 42 mL of 0.32 mol/L NaOH. What is the resulting pH? 0.64 · 5.21 · 3.42 · 4.43 That is incorrect. Please try another answer. Alternatively click this text for the solution Equation: HCl + NaOH -> H2O + NaCL Determine the substance in excess after neutrilisation by calculating moles by multiplying volume with concentration — Next (1/2) Moles of HCl: 0.058L * 0.63 mol/L = 0.0365 mol Moles of NaOH: 0.042L * 0.32 mol/L = 0.0134 mol — Next (2/4) Determine remaining moles after neutrilisation: 0.0365 - 0.0134 = 0.0231 Then divide this by total volume to get molarity: 0.0231 / (58 + 42) = 0.231 — Next (3/4) Then determine pH by taking -log[H+]: -log[0.231] = 0.64 — Next (4/4)	Developed pH module
12/6	 Stoichiometry Assuming CO2 in excess. How much C6H12O6 would be produced given 53 grams of H2O 6CO2 + 6H2O -> 1C6H12O6 + 6O2 84.68 grams · 89.66 grams · 85.04 grams · 88.28 grams That is incorrect. Please try another answer. Alternatively click this text for the solution First determine the moles of the reactant you are given by dividing the mass by the molar mass of the substance: 53 g ÷ 18.02g/mol = 2.94mols — Next (1/3) Then use the mole ratio given by the equation to determine moles of products formed: 2.94 mols * 1 / 6 = 0.49mols — Next (2/3) Then multiply the number of moles by the molar mass of the product to determine mass of products formed: 0.49 * 180.16 = 88.28 g — That is all (3/3)	Developed Stoichiometry module
13/6	 Learn Chemistry — Settings Atomic Mass: 3 questions answered, 3 questions correct, 0 questions incorrect, 100% correct answers Electrochemistry: 7 questions answered, 4 questions correct, 3 questions incorrect, 57% correct answers Half Life: 3 questions answered, 0 questions correct, 3 questions incorrect, 0% correct answers Stoichiometry: 2 questions answered, 2 questions correct, 0 questions incorrect, 100% correct answers Energy: 0 questions answered, 0 questions correct, 0 questions incorrect, NaN% correct answers pH: 2 questions answered, 0 questions correct, 0 questions incorrect, 0% correct answers	Reworked the CSS to make the website handle resizing of window size better.
14/6	 Settings Progress: Reset Theme: Normal · Large	Implemented the settings page. Allows for the resetting of progress and change of font size. Encountered a problem where while Javascript can manipulate CSS elements, it cannot manipulate CSS

	 Atomic Mass How much do 82 moles of carbon weigh? That is incorrect. Please try another answer. Alternatively click this text for the solution There are 82 moles of carbon. Therefore multiply 82 by the atomic mass of carbon. 12.011g/mol x 82 mol = 984.9g  Electrochemistry 2 questions answered 2 questions correct 0 questions incorrect 100% correct answers	classes. Worked around this by using a separate function that returns all the elements of a given class, and then iterating over this, changing the CSS of every element.
17/6	 	Essentially finished with all necessary functionality of my code. Now focusing on improving the interface and user experience. Made buttons more intuitive by adding hover effects and an extra layer of dimensionality with shadows

Annotated Bibliography

Robin Parisi, (Unknown year), Modal Plugin Written In Pure Javascript
https://robinparisi.github.io/tingle/

- The developer documentation for the tingle modal

Nick Babich, (2018), 7 Basic Rules for Button Designs
https://uxplanet.org/7-basic-rules-for-button-design-63dcdf5676b4

- The guide used to implement intuitive and user friendly buttons

W3schools, (Unknown year), Javascript Tutorial
https://www.w3schools.com/jS/default.asp

- An unofficial Javascript developer documentation. Used as a reference for syntax and other Javascript specific information.