

CHEM QUIZ GENERATOR

GGC Spring 2019 – ITEC3870



MAY 1, 2019
CATALYST QUARTET

Contents

Quick Setup Instructions.....	2
Features	3
Front-End.....	4
Front-End Code View	5
Testing	7
Catalyst Quartet.....	8
Client Info.....	9
Intellect Property Agreement	10
Flyer.....	11
Survey Result.....	12
Demo.....	14

Quick Setup Instructions

Local Deploy

1. Install the edition of Node.js appropriate to your device from [this archive link](#). If you have a more recent version of Node.js installed already, it may be worth reverting to the 8.11.4 version to ensure compatibility.
2. In the root directory /ChemQuizGen/ run the terminal command `npm link`. (If you receive an error, ensure `package-lock.json` does not previously exist.)
3. Going forward, you will be able to use the command `ng serve` to start up a live development server of this web application on your computer at <http://localhost:4200/>.

GitHub Pages

1. To update GitHub Pages within Master branch use Npm run `-script ghbuild`. this command will update the Doc folder which is associated with GitHub Pages.

Features

IMPLEMENTED FEATURES

Frontend

1. Ability to do other limiting reactant problems
 - a. Molarity questions
2. Ability to print out question without an answer
3. Fav Icon change
4. Home Page was added for better navigation
5. Navigation bar
6. Host to Online Server
 - a. GitHub Pages

MISSING FEATURES

Frontend

1. Restrict the output answer to a much accurate number

Front-End



Chemistry Quiz Key Generator

Home Page

Welcome to the Chemistry Quiz Generator. Here you can use the dropdown menu to access chemistry questions for stoichiometry or molarity questions.

Menu

Stoichiometry Equations
Molarity Equations

Copyright © 2019 Catalyst Quizzes



Chemistry Quiz Key Generator

Molarity Form

Menu

Input a conjugate base-acid pair, volumes, and molarity values of each, then press "Calculate".

SO₄ ▾ Volume of Base Molarity of Base
Na ▾ Volume of Acid Molarity of Acid
[Calculate](#)

Copyright © 2019 Catalyst Quizzes



Chemistry Quiz Key Generator

Stoichiometry Form

Menu

At the moment, this program calculates stoichiometry, to be used in limiting-reactant problems, involving monoatomic and polyatomic ions performing replacement reactions. Input the data into the form and click Calculate to generate the answer key. (Combustion and gas-evolution reactions using metals and acids may be added in a future update.)

Li ▾ Br ▾ Grams of First Compound
Na ▾ I ▾ Grams of Second Compound
[Calculate](#)

Answer Key

Copyright © 2019 Catalyst Quizzes

Front-End Code View



Chemistry Quiz Key Generator

Welcome to the Chemistry Quiz Generator. Here you can use this website to practice sample chemistry questions for stoichiometry and molarity questions.

Equations

Stoichiometry Equations

Molarity Equations

```
1 <h1 align="center">Chemistry Quiz Key Generator</h1>
2 <h2 align="center">Welcome to the Chemistry Quiz Generator. Here you can use this website
3   to practice sample chemistry questions for stoichiometry and molarity questions.
4 </h2>
5 <br>
6
7 <!--<div class="form-container">
8   <a routerLink="/stoichiometry"></a><button class="btn btn-success pull-right">Stoichiometry Eq
9   <a routerLink="/molarity"><button class="btn btn-success pull-right">Molarity Equations</button
10 </div-->
11
12 <div class="navbar">
13   <a href="form-container"Home></a>
14   <div class="dropdown">
15     <button class="dropbtn">Equations
16       <i class="fa fa-caret-down"></i>
17   </button>
18   <div class="dropdown-content">
19     <a routerLink= "/stoichiometry">Stoichiometry Equations</a>
20     <a routerLink= "/molarity">Molarity Equations</a>
21   </div>
22 </div>
23 </div>
24
```



Chemistry Quiz Key Generator

Stoichiometry Form

At the moment, this program calculates stoichiometry, to be used in limiting-reactant problems, involving monoatomic and polyatomic ions performing replacement reactions. Input the data into the form and click Calculate to generate the answer key. (Combustion and gas-evolution reactions using metals and acids may be added in a future update.)

Li ▾ Br ▾

2

Na ▾ I ▾

3

Calculate

Answer Key

$$\text{LiBr(aq)} + \text{NaI(aq)} \Rightarrow \text{LiI(aq)} + \text{NaBr(aq)}$$

mole ratio of LiI to LiBr is: 1.000
mole ratio of LiI to NaI is: 1.000
mole ratio of NaBr to LiBr is: 1.000
mole ratio of NaBr to NaI is: 1.000

Calculate the limiting reactant, excess reactant and theoretical yield in grams when: 2 g of LiBr reacts with 3 g of NaI according to the following equation:
$$\text{LiBr(aq)} + \text{NaI(aq)} \Rightarrow \text{LiI(aq)} + \text{NaBr(aq)}$$



Input a conjugate base-acid pair, volumes, and molarity values of each, then press "Calculate".

NaOH is the limiting reactant giving 0.003 moles, 0.18015 grams of water, and 0.7101900000000001 grams of salt.

Calculate

The screenshot shows an IDE with the following components:

- File Explorer:** Shows the project structure with folders like 'src', 'test', and 'target'.
- Code Editor:** Displays the implementation of the `ReliabilityComponent` interface. The code includes:
 - Imports: `import (Component, OnExit, Import) from (Angular/core);`, `import {ForwardControl, Forwarding, Forwardable} from (Angular/forms);`, and `import {ElementInterface} from './elementInterface';`
 - Class Definition: `export class ReliabilityComponent implements OnExit {`
 - Attributes: `@Inject() private forwarder: Forwarder;`, `@Inject() private elements: ElementInterface[];`
 - Methods: `addElement(element: ElementInterface): void;`, `removeElement(element: ElementInterface): void;`, `getElements(): ElementInterface[];`, `onExit(): void;`
 - Implementation Details: The `addElement` method calls `forwarder.forward(element)` and adds the element to the `elements` array. The `removeElement` method removes the element from the array. The `getElements` method returns the array. The `onExit` method calls `forwarder.forwardAll(elements)`.
- Terminal:** Shows the command `mvn clean compile` and its output:
 - `[INFO] Compiled successfully.`
 - `[INFO] Compiling...`
 - `[INFO] wait until bundle finished: reliability`
 - `0 unchanged chunks`
 - `[INFO] Copying...`
 - `[INFO] Copying...`
 - `[INFO] wait until bundle finished: AngularJS`
 - `[INFO] wait until bundle finished: AngularJS.js`
 - `[INFO] wait until bundle finished: browser`
 - `[INFO] wait until bundle finished: vendor.js`
 - `[INFO] wait until bundle finished: vendor.js`



This is simple, just make sure none of the simple Test methods are returning false with the proper inputs.

Run the .java file ChemQuizTest, for the simplest experience use IntelliJ, but other Java IDEs should work as well. Make sure to run the file in an environment with both JUnit 4.12 and Selenium 3.141.59 at least. You will also need a chrome web driver, with can be found with Google. Once the file is open you will need to change the

"C:\\Users\\Wes
McMillen\\Documents\\2019C\\SoftwareQA\\chromedriver.exe"

Input in the `System.setProperty()` method found in the method `setUpSelenium` under the `@BeforeClass` to the location of the Chrome driver on your computer.

From there simply run the program and Selenium and JUnit will do the rest.

```
public void SO4Na111() {  
    driver.get("http://localhost:4200/molarity");  
  
    Select bases = new Select(driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/select[1]")));  
    bases.selectByVisibleText("SO4");  
  
    WebElement baseVol = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/input[1]"));  
    baseVol.sendKeys("1");  
  
    WebElement baseMol = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/input[2]"));  
    baseMol.sendKeys("1");  
  
    Select acids = new Select(driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/select[2]")));  
    acids.selectByVisibleText("Na");  
  
    WebElement acidVol = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/input[3]"));  
    acidVol.sendKeys("1");  
  
    WebElement acidMol = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/input[4]"));  
    acidMol.sendKeys("1");  
  
    WebElement calculate = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]/form/button"));  
    calculate.click();  
  
    WebElement textGetter = driver.findElement(By.xpath("/html/body/app-root/app-molarity/div[2]"));  
    String text = textGetter.getText();  
    text = text.substring(60);  
    Assert.assertEquals("NaOH is the limiting reactant giving 0.0005 moles, 0.018015 grams of water, and 0.0710190000000000
```


Catalyst Quartet



1. **Austin Lomax** - *Data Modeler & Client liaison* 🧠
 - ❖ Senior at Georgia Gwinnett College pursuing a bachelor's degree in Information Technology with a concentration in Software Development.
2. **Farzin Valizadeh** - *Team Manager / Documentation Lead* 🐱
 - ❖ Senior student at Georgia Gwinnet College studying Software Development Field, Completed it Projects in Java, Mobile Application, HTML.
3. **Wesley McMillen** - *UI/UX modeler / Testing Lead* 🌀
 - ❖ Software Developer major at Georgia Gwinnett College.
4. **Matthew Stiller** - *Architecture / Lead Programmer & Testing Lead* 🌀
 - ❖ Senior student studying in the Software Development field at Georgia Gwinnett College. Completed projects spanning Python, Java, R, JavaScript, and Angular applications, and intends to pursue graduate school.

Client Info



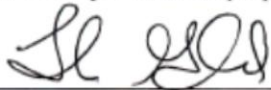
Dr. Thomas Gluick is a chemistry professor at Georgia Gwinnett College. He teaches classes such as Survey of Chemistry, Principles of Chemistry and Biochemistry. He currently holds a bachelor's degree in Chemistry from Princeton University as well as a Doctorate in Chemistry from the University of Montana.

Intellect Property Agreement

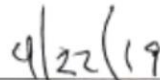
Intellect Property Agreement

This intellectual Property Contribution and Assignment Agreement (the "Agreement") is made as of 22 April 2019 by and between the members of the Catalyst Quartet (the "Company"), Austin Lomax, Farzin Valizadeh, Matthew Stiller, Wesley McMillen and Dr. Thomas Gluick (the "Client"), and Dr. Cengiz Gunay (the "Sponsor").

All parties have agreed members Austin Lomax, Farzin Valizadeh, Matthew Stiller, Wesley McMillen of Company will receive an equal share of 16%, totaling 64% each of intellectual claim Partner Dr. Thomas Gluick will receive 20%, totaling 84%, and Sponsor would be assigned 16% of intellectual claim. (Intellectual claim includes entire rights, title and interest in and to any and all the properties that exist as of the date hereof.



Signature



Date

Thomas Gluick

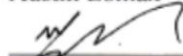


Signature



Date

Austin Lomax



Signature

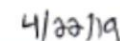


Date

Farzin Valizadeh



Signature



Date

Matthew Stiller

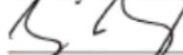


Signature



Date

Wesley McMillen



Signature



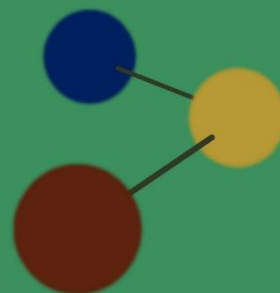
Date

Cengiz Gunay

Catalyst Quartet Presents:

Chemistry Quiz Generator

**Farzin Valizadeh | Matthew Stiller |
Austin Lomax | Wesley McMillen**



Survey Result

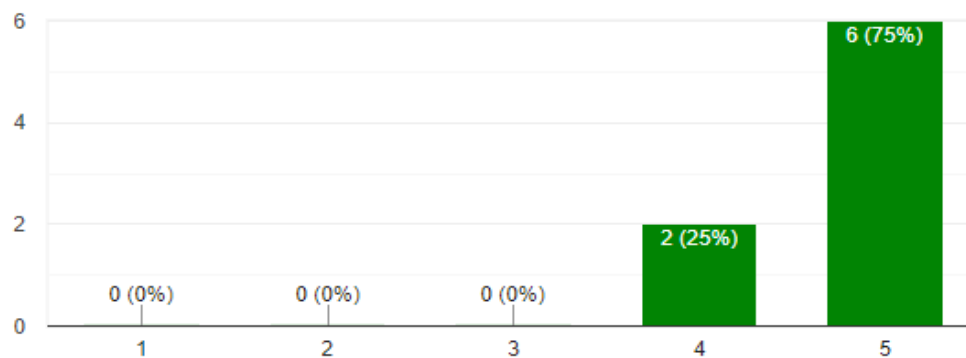
Chem Quiz Gen Feedback

8 responses

[Publish analytics](#)

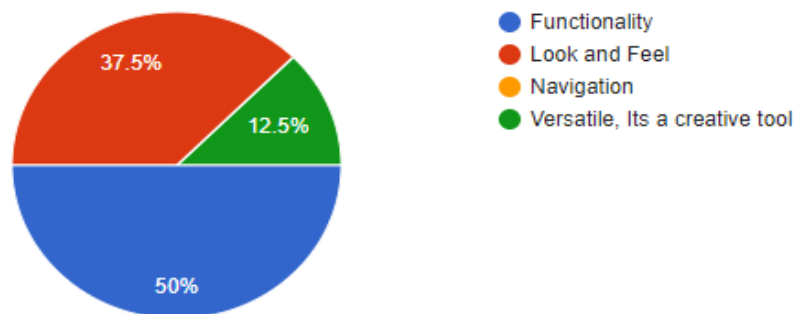
Was Chem Quiz Gen Website Useful?

8 responses



What did you like about Chem Quiz Gen Website ?

8 responses



Bug Reports

5 responses

None
negative values and strings aren't filtered out
none
Needs a menu or nav bar to understand the user's position, but great otherwise!
n/a

Feature Request

5 responses

None
Restrict the output to the number of accuracy
bolded answers
Maybe different types of questions? Like it wouldn't apply to chem equations, but maybe for more textbook-info type stuff
n/a

Demo

[Google Drive Link](#)