# Emergency Calculator

Developers: Dylan Engle, Eric Bunch, Weston Abner

## Abstract

Emergency calc will produce accurate calculations for basic math problems, as well as more rigorous calculations. The calculator will also be capable of creating two dimensional graphs using complex equations. It will include built in functions and variables for compressing commonly used math functions to help simplify equations.

## Technology

* HTML5
* CSS
* Typescript
* JavaScript
* VSCode
* GitHub
* Docker
* Node.js
* Microsoft SQL server
* SQL

## Requirements

1. Front-End:
   1. User Interface
      1. Implemented with HTML5 and CSS, scripting with typescript.
      2. Canvas for two-dimensional graphing.
         1. Canvas will be a 2d grid with points for position graphing.
         2. Will display as a grid with grid lines
            1. Will have an option to disable gridlines
         3. Will be created from a given function.
         4. Will feature default range for graph
            1. Allow zoom for graph features. Allows users to adjust range.

Zoom into the graph relative to the mouse using mouse button 3. (Scroll Wheel)

Drag zoom with mouse based on mouse position

Buttons that allow the user to zoom in and out from the graph. (Plus, and Minus)

* + 1. Upon hitting the exit button, confirmation box will appear
       1. Box will prompt the user asking them if they want to save
          1. If no exit the website without saving
          2. If yes, ask if they want to save graph data, equations, or both

Both buttons will call both the file saver and the image exporter (3, 4)

Individual buttons will call whichever application is selected (image exporter or file saver)

* + 1. CUI (Calculator user Interface)
       1. Will feature a collapsible menu in order to bring up calculator buttons
          1. The button will input into the calculation form
          2. When menu button is clicked, the menu will expand showing many new buttons for user, and can be collapsed

Buttons will highlight on mouseover

Buttons will be input into calculation form (1.1.5.

* + - * 1. Menu will contain buttons for most math functions

Basic math operators: addition, subtraction, multiplication, division

All Numbers: [0-9]

Trigonometric functions: sin, cos, tan

Exponential functions and roots

Exponents: x, y, z

Logarithmic functions: log, ln, log base

Important variables: e, pi

Constant Physics variables: Newtonian constant of gravitation, speed of light in vacuum, planck constant

Moving back and forth in an equation (similar to hitting the left and right buttons on a keyboard), for moving the cursor left and right

Clear button to remove current calculations from screen

Changing number from positive to negative and vice versa

* + - 1. Button to pull up table menu
         1. User can input specific locations into the graph

Will display a table for x and y values

Will only have two columns, for x and y

Will feature unlimited rows so the user can have as many lines on the graph as they want

Will place a dot at specific x and y value on the graph

Delete all button that will remove all points from the table

* + 1. Calculator form
       1. The calculator form will take input from both the CUI buttons and the user’s keyboard
       2. Hitting enter in the calculator form or hitting the finished button beside the form will bring up 2 buttons

The calculator form will bring up an enter button to run the calculations in the form

Hitting this button will display the answer to the calculation in a text box

The calculator form will bring up a graph button

Hitting this button will bring up a menu with options for colors, styles, and an ok button

The styles will show a dropdown menu letting you choose from points, single line, or dotted line.

The colors will also be a drop-down menu allowing the basic colors as options (yellow, green, blue, violet, red, orange, and black)

The default for the graph button will be black and single line

Hitting the ok button will graph on the canvas (1.1.2.)

Hitting this button with a graph already on the canvas will graph onto the same canvas that the graph is already on

The form will contain a clear graph button that will clear all graphs from the canvas

This will bring up a text box with a yes and no

Hitting yes will remove graphs hitting no will remove the buttons from the screen

* + - * 1. If there is an error that is entered into either the graph or enter button a text box will appear

The text box will contain the error along with 2 buttons

It will have a clear calculator form button

It will have a keep calculator form button

* + 1. Hamburger menu button for saving and loading
       1. Save Calculations
          1. Save user calculations as addressed in 3.1

Will be saved from given user login

* + - 1. Load Calculations
         1. Load user calculations as address in 3.2

Will be loaded from given user login

* + - 1. Save Graph
         1. Save user-made graphs as addressed in 3.3
      2. Load Graph
         1. Load user-made graph using functions from 3.3

1. Back-end:
   1. Google Chrome will be the supported browser
   2. Using Docker to run the server
      1. Server language will be JavaScript
      2. Files from user will be saved to database
         1. Calculations saved in .csv files
      3. Files from user will be loaded from database
   3. Database using Microsoft SQL server
      1. 3 Tables
         1. User information
         2. Calculation saving information
         3. Image saving information
   4. Typescript Application for buttons
      1. Each button mentioned in 1.1.3 will be implemented
   5. Application for saving files
      1. Will feature parts that will allow the user to save different types of files (continued in 3)
   6. Application for Exporting images
      1. Will implement simple image exporter (Continued in 4)
   7. Application for sharing
      1. User can send calculations using a given username
         1. Once another valid username is added, the other user will be given access to the first users save file
         2. Will allow loading from the receiving user's calculator, not a file.
            1. Can be accessed in their load file menu.
2. File Saver:
   1. Will allow user to save calculations to a .txt
      1. When the user chooses to save, a box will appear to ask for a save location. This will occur for every save instances for all functions that deal with saving.
   2. Will allow user to load calculations from a .txt
   3. Will save graphing functions to a .txt
      1. Allow user to quickly pull up an already made graph
3. Image Exporter:
   1. Export user created graph to PNG, JPG
      1. Give user option to save as PNG or JPG
         1. User interface to ask where to save image
         2. Message notifying user graph has been saved
         3. Returns to User interface noted in 1.1

## Timeline:

|  |  |  |  |
| --- | --- | --- | --- |
| Week: | Eric | Weston | Dylan |
| 9/12 | Design Main Page | Design Main Page | Design Main Page |
| 9/19 | Design Main Page | Design Main Page | Design Main Page |
| 9/26 | Get docker running | Implement graph canvas | Implement main buttons |
| 10/3 | Push the website onto docker platform | Implement graph canvas features | Implement main buttons |
| 10/10 | Database | Export graphs as png | Graph buttons |
| 10/17 | Database | Export graph as png | Calculator Form |
| 10/24 | Finish up File Menu: Design on website | Export graph as jpeg | Calculator Form |
| 10/31 | Graphing Table Menu | HT | Graphing Table Menu |
| 11/7 | Perfect front-end design | HT | Perfect front-end design |
| 11/14 | Perfect front-end design | HT | Perfect front-end design |
| 11/21 | Testing/Debugging | Testing/Debugging | Testing/Debugging |
| 11/28 | Testing/Debugging | Testing/Debugging | Testing/Debugging |

### Guide:

**HT = Helping Teammates**

**If a person’s job is finished early, work with others to finish their jobs.**