# Emergency Calc

Developers: Dylan Engle, Eric Bunch, Weston Abner

The backend of the website has changed entirely and uses almost none of the tools that we discussed at the beginning of this project. The first thing is that we are using node.js instead of docker to host the server. While docker could potentially provide us with a better working environment, especially with a remote student, Node.js makes hosting a server not as difficult as docker. Along with the difficulty, many of the packages that Node offers makes the server hosting process much more efficient and it is much more appropriate for the scope of our project. The database has also changed. Instead of Microsoft SQL server we have went with an alternative, MySQL. We are using this database through and application called XAMPP that allows us to locally create and host our database on an Apache server. As stated above with Node, this allows us to create a local environment where we do not have to rely on Microsoft’s SQL Server and is more appropriate for the scope of our project. The server, website, and database will all be merged shortly, however Eric has opted to create a small website project on his own to ensure that all the functionality it all works before merging it into the rest of the group’s work. Along with this, the server being used is being hosted through a Node package called Express. Express allows us to generate templates that make the creation of the server much easier. The package works such that when we get a request to enter the website or go to a different page on the website, we will be able to dynamically render HTML pages to the user. With this we will also be using .EJS files instead of .HTML files to have imbedded JavaScript inside of the HTML. Express has many dependencies, so there are a lot of node modules working in the background to make it work. We will be opting to use body-parser to handle the HTML forms. This allows us to gather information from the HTML form and read it into the database. The content of the database has not changed at all, only the database that we are using has changed. As stated, the express package is a critical key in the server that we have chosen, in fact, express is what the server is being ran on, it does much of the work in the background and we can change the given information to correlate with our website. This of course also allows us to quickly create routes which will allow us to create many different pages that the users will be able to navigate (if we have many pages and something changes). The plan now is to try to bring the page count down to three. As far as the backend is concerned, these are all the changes that we have made. This is subject to change even further, but we feel that what we have now will be able to carry us through the project with as little difficulty as possible.

The graphing part of the GUI has changed significantly with the introduction of the p5.js library. This library allows for a multitude of tools that are not included in the basic html canvas or in basic JavaScript. It has its own canvas that will replace the default html canvas that was going to be used originally. This canvas allows anything made with in it to be used and edited with the functions created by the p5 library. It will allow changing the color of the lines to be easier and give more freedom to the user in designing their graphs and give more options of colors for the team to program into the graphical interface. It also makes the creation of all the lines from the grid lines to the graph itself easier to implement. It allows for the canvas to be redrawn at anytime with one simple command allowing for multiple graphs to easily be added to a singular canvas. This redraw function also makes it easier to reset the graph when the user is finished with it. Another reason this library was chosen is that it has a built-in function to monitor the use of a mouse which will make manipulation of the graph canvas by the user easier. The p5 library also includes functions that will make the exporting of the graph canvas as a file easier.

The front end of the project has been making slow but steady progress while going through some very different design choices then were there previously. The choice made with helpful advice to drop save states was a large factor in some of the over all changes. Due to this the collapsible part of the calculator has had major revisions in order to handle the new system for saving formulas to the profile of the user. Though this part isn’t fully up, there’s been a lot of progress and set up for this feature to be able to connect to the database in order to choose from the saved functions attached to ones profile. During this phase we will also be implementing the proper calculator functions with this menu and using a form to allow the user to choose to type out the formulas they care to use along with being able to write these functions using the calculators in-progress buttons. The hamburger menu containing the register, log in, log out, and save functions has been implemented and now Dylan and Eric are working on integrating these features together with the database.

Currently most of the project is partially complete. The server is currently running and interacting with the database, but it is a test server connecting to a test database. This server works with the p5.js library and is currently only useable through a Lan connection. The database has most of the tables that will be used in the final project but is not currently complete and none of the function calls from the website to the database are currently implemented. The basis of the Ui is also mostly complete. With some of the aforementioned features add in the last paragraph we are ready to move forward in adding a background layer that includes the work done for the graphing canvas. The graphing canvas itself is mostly complete with functions to create gridlines, along with remove them being completed and a function to draw the actual graph mostly completed just needs to be adjusted and set to accept user input. The function to adjust the range of the graph is currently not implemented but will be simple to adjust once the actual graphing function is finished.

The calculator and its functions still need to be implemented into the website. A lot of the basic functions were discovered to be creatable using the Math object that is built into JavaScript. This still leaves a lot of mathematical functions that need to be designed along with the calculator class itself. The zoom in/out function of the graph is not implemented yet. The exporting of the graphs from the graph canvas are also not implemented yet.

There are a couple of unresolved issues that are known to the design team at this time. One of these is the exporting of graphs from the graph canvas. This is a major concern as the exporting of the graphs is a large part of making this project useful to stem students. Even though this is a major concern to the overall project it should only require a minor amount of work to get it working correctly using the built-in functions from the p5 library. Another of the concerns is the zoom in/out functions for the graph as part of the GUI. This is a major concern as it is also a large part of what should make this project useful and is one of the reason that the p5 library is being used as it should make zooming in and out of the graph using the mouse easier to achieve.