Contents

[08-29-2020 Class Homework Help 2](#_Toc48057229)

[Document Goal 2](#_Toc48057230)

[Assignment 2](#_Toc48057231)

[CSS 2](#_Toc48057232)

[Styles.css 2](#_Toc48057233)

[Mazecontroller.js 6](#_Toc48057234)

[\_mazeForm.pug 6](#_Toc48057235)

[mazeWorker.js 7](#_Toc48057236)

# 08-29-2020 Class Homework Help

## Document Goal

In order to help you with the homework I am giving you some comments and notes over the next two weeks. This I hope keeps you engaged and gives you a way to be successful without too much frustration if you do not already know the Node language.

## Assignment

For this week’s homework you are to change the way the maze is displayed to use a table with borders set for the cells.

# CSS

## Styles.css

You need to add a style for your table. You can get the format from the class table.scss. This is how you do table formatting with CSS in the style.css class. You only need to really have place holders for .table, and tr. Then for TD you need some definitions. This is an example of mine. You can see I am using the keywords table, tr and td. I am adding right, left, top, bottom along with read versions of this. The reason is future use so I can build a red trail for what a user has traversed.

 table {

    font: 11px/24px Verdana, Arial, Helvetica, sans-serif;

    border-collapse: collapse;

    }

  tr {

    border-collapse: collapse;

     }

  td {

    border-bottom: 1px solid #FFFFFF;

    padding: 0 0.5em;

    }

  td.right {

    border-right: 2px solid #000000;

    }

  td.left {

    border-left: 2px solid #000000;

    }

  td.bottom {

    border-bottom: 2px solid #000000;

    }

  td.top {

      border-top: 2px solid #000000;

    }

  td.redTop {

    border-top: 2px solid #000000;

    }

  td.redRight {

    border-right: 2px solid #000000;

    }

  td.redLeft {

    border-left: 2px solid #000000;

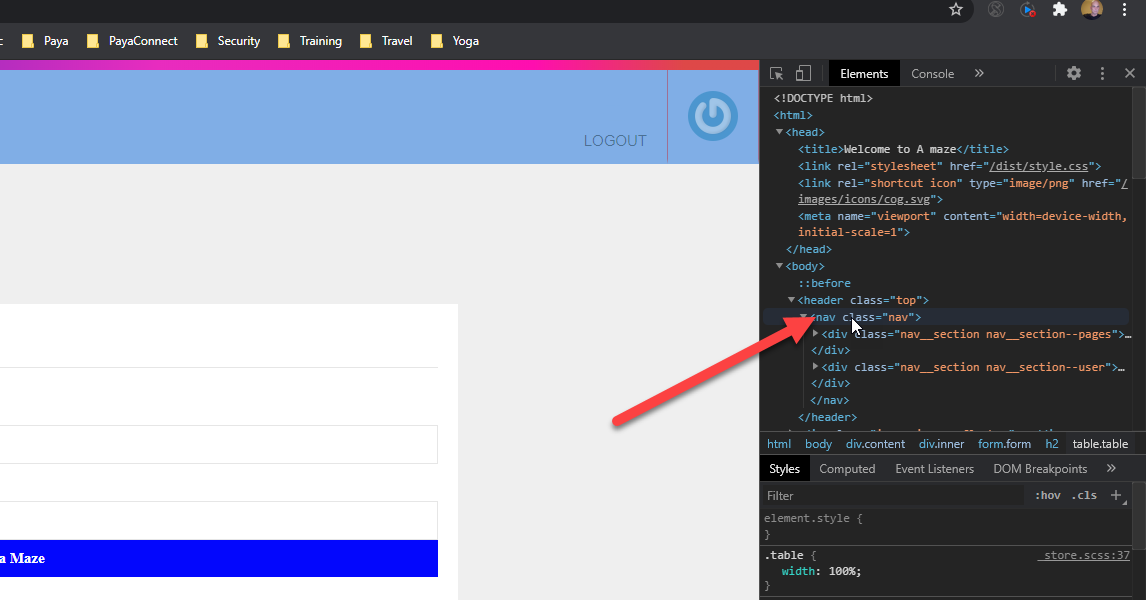
    }

  td.redBottom {

    border-bottom: 2px solid #000000;

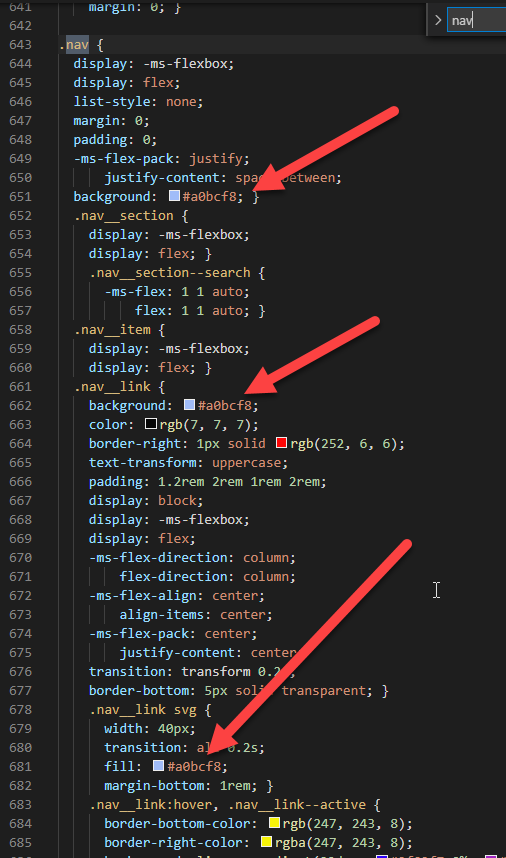
    }

I also changed my header for the page to have some background color. You can figure this out with the developer tools in Chrome.



You can see the class is called NAV. So I modify the code in the NAV section of the style.css class.

I need to change three places to make the color all the same.



# Mazecontroller.js

Not much to change here. Because I only really changed the function name I call.

    //console.log(req.body);

    //console.log("Found request for maze that is " + req.body.height + " by " + req.body.width + " in size!");

    // lets check the size and if ok send us the maze display

    if (req.body.height >= 5 && req.body.width >= 5 && req.body.height <= 70 && req.body.width <= 70) {

      maze.width = req.body.width;

      maze.height = req.body.height;

      mazeWorker.getTableMaze(maze.width, maze.height).then((result) => {

          maze.drawing = result;

          res.render('main', { title: 'Welcome to A maze',  maze });

      }).catch((error) => {

          console.log("Error", error);

          response.end("Error: " + error);

      })

# \_mazeForm.pug

I made quite a bit of change here. I setup a button for saving the maze to the mongo.db and I changed how the maze displays.

mixin mazeForm(maze = {})

  form.form(action='/' method="POST")

    h2 Size

    label(for="width") Width

    input(type="text" name="width" value=maze.width, maxlength='6', style="witdh:5px")

    label(for="height") Height

    input(type="text" name="height" value=maze.height, maxlength='6', style="witdh:5px" )

    input(type="submit" value="Get a Maze" class="button")

    - var x = 0

    - var y = 0

    h2 Maze

      p !{maze.drawing}

  form.form(action='/saveSave' method="POST")

    input(type="submit" value="Save To My Mazes" class="button")

Specifically I changed from a PRE to paragraph object. You can see this here on this line.

    h2 Maze

      p !{maze.drawing}

# mazeWorker.js

This class is the hardest to change of all of them for this work assignment. Here is some sample code you can see what your code should do in order to display a maze. This is just example code of what your code should output.

            let sb = new StringBuilder();

            let newColumn = true;

            // table borders

            sb.append("<table class=\"table\">");

            // uncomment for example table that is simple use of the css classes for

            sb.append("<td class=\"left bottom\">\_</td>");

            sb.append("<td class=\"top bottom\">\_</td>");

            sb.append("<td class=\"top right\">\_</td>");

            sb.append("</tr><tr class=\"noBorder\">");

            sb.append("<td class=\"top bottom right left\">\_</td>");

            sb.append("<td class=\"top bottom right left\">\_</td>");

            sb.append("<td class=\"right left\">\_</td>");

            sb.append("</tr>");

            sb.append("</table>");

Once you understand this code you can modify the function to build a table to display with only the correct columns highlighted.

We are using a string builder so you see the sb.append. This is a more efficient use of memory for all the many strings we need to append to build our output.