CSS Documentation Report for Atkinson Baker

General Code. Line#

* General Code
  + Open All Button 197
  + Animated Arrows 243
  + Wrapper (Background gradient Styling) 118
  + Accordions 131
    - Accordion Header 132,151,163
    - Accordion Container 209
    - Accordion Tables 228
* General Modal Close button 262
* Dashboard 282
  + Accordion 285
  + Accordion Container 312
  + Upcoming title with openall 317
  + Upcoming Title 322
  + Accordion Headers 327
  + All cells 381
  + Upcoming Table Data 402
  + Sub Accordions 420
  + Sub Accordion Content 437
  + Dash Board Images 445
* Upcoming Jobs 458
  + Open All Button 453
  + Main Headers 458, 463
  + Accordion Content Header 469
  + Accordion Content Border styles
* Job List 782
* Job Details 943
  + Transcripts 1004
  + Supporting files 1019
* Files 1680
* Schedules 1546
* Reports 1937
* Invoices 1986
* Profile 2205
* Login 2354
* Menu 2514

**Dashboard: Details**

(line# 287)

The Accordions in Dahsboard are basically the same except they don’t have much code for the header titles in the HTML page.

In the styles (line# 292) the accordion has a min-width: 200px to keep it from getting smushed after that its set at 98%.

Within the AccordionContent there are the sub accordions which have the same basic properties as its parents.

The Entire table is set as an HTML table; the first cell has a colSpan and rowSpan of 2 to give its larger size.

The other cells have a set width of 10%. And a min-with: 150px so they too don’t look squished.

**Upcoming Job: Details**

(line# 458)

Headers above in and above the accordions are made up of divs displayed as tables

Example:

Html:

<div class=”table”>

<div class=”row”>

<div class=”cell”>

*(cell content)*

</div>

</div>

</div>

CSS:

.table { display: table;}

.row {display: table-row;}

.cell{ display: table-cell;}

Inside the Accordion Content

Tables: Are traditional HTML tables

The Right and Left Cells have an overflow-y: scroll. So the whole thing isn’t stretched out if there’s too much data. That is in line 543 & 544.

The pdf Icon is surrounded by an Anchor tag (<a>) which is the trigger for opening the modals

Note: originally the tags were set to position: absolute and placed over the pdf icon. So the accordion wouldn’t open when the modal did. It worked but the other accordions moved and the tags stayed in placed losing its purpose.

The map is set to 100% width so it stretches with the page, and has a min-width of 175px.

**Upcoming Modals**

The Upcoming modals start on line 597. Created using an html table within the modalContent

Currently to open an individual modal each anchor tag needs a unique ID. Since I added multiple accordions as examples, they all open the same modal. To open a modal synced to a specific accordion it will need its unique modal Id. I think that could be generated with each new accordion simultaneously.

**Jobs List: Details**

*(line# 755)*

Jobs list is very simple. It’s a CSS table created with divs. In the stylesheet I created the borders individually with the ‘border:’ property. The first row is using the ‘:first-child’ selector since there will always be a first row. And the last row uses the ‘:last-child’ selector since there will always be a last row.

The width of the table is 98% and each cell is set at 10%.

**Job Details: Details**

The Content in the Accordions are all Div CSS tables, certain row are surrounded by a div with the class border.

Table cells are at a set width of of 600px. As a standard and set width to keep it from squishing.

The Table its self is set at a width of 70%.

Since these tables are all created with Divs the map needed to take up two spaces. I accomplished this by surrounding the cell and giving it a class med ‘colSpan’. And the row with a class ‘map’.

The CSS has the display as flex with flex-flow: wrap, and colSpan with flex: 2;

Ex.

//html

<div class=”row map”>

<div class=”cell colSpan”></div>

</div>

//CSS

.map

{

display: flex;

flex-flow: wrap;

}

.colSpan

{

Flex: 2;

}

This will give the map the length of two cells.

Supporting Files Shipping & Job History

Tables: Css Tables with surrounding borders

The bottom row for the arrows and pagnation is separated from the table.

It is a separate div stretched to match the other rows.

I had difficulty creating a colSpan to take up multiple rows using CSS tables. I successfully did this above for the map. But for some reason I couldn’t repeat it here. So this was my work around.

**Calendar**

I left the Calendar alone just added the modal once a user clicks on the trigger.

Calendar Modal

The Modal Content has a table like the previous ones created with divs and CSS.

The last and cells has an imbedded Upcoming Jobs table. The outside cells are set to overflow-y: scroll to keep the modal from stretching.

**Schedules**

Schedules is very similar to Job Details. Some necessary information has a red circle next to it. I added the red circle dimensions to all the form Text to keep them aligned.

The inputs mostly consist of type: text or radio button, except for

General Information > preceding type : drop down

Case > Trial Date : calendar

Case> Case Type :dropdown

Case> Case Caption :text Area

Billing > Stated :dropdown

Date of Loss > :Calendar

RealTime>live disposition :dropdown

Delivery>special Instruction :textArea

The notes added in there are not within the table cells so it stays centered in the page.

**Files**

Is very similar to Upcoming jobs in the way it is set up.

It has a header on the very top and within the accordion is the information related to the header.

In its contents everything is inside an html table. Mostly consists of text headers, image icons and buttons.

The exhibits button opens up the Exhibits modal similar to that of upcoming jobs. Except here its located within the accordion content. The Modal is not much different than the one from Upcoming jobs its made with a basic table. And also includes a Check all button powered with JavaScript

**Reports**

Reports page is just basic CSS displayed table with images within. I used the CSS table to style the border easier. There are four rows and five columns. The third row down is hidden to give that extra space.

**Invoices**

Is a also a basic CSS styled table. The headers and table text are in ‘vw’ units so they shrink down with the page. Prior to this the headers would look to squished. First and last rows are selected in css using :fist-child and :last-child selectors.

**Profile**

Has the same setup as Job Details. But I gave it all a min width because there was too much information in the last two accordions.

**Login Page**

Login Page Page links are buttons and Anchor tags. To get back into the site just click on the request account link. A temporary link.

The background is set to cover but I know this will change later. To remove the background look for

“.BG”

**Menu**

From the Menu bar up is all located in the Menu section of the CSS file. There are four layers to it. Background, ul,li,a. the drop down animations are all located in the menu section as well. The Text is set to ‘vw’ to shrink with the page and avoid the text over lapping itself.

I made the ‘li’ smaller than the BG and the anchor tags to give it the floating separator look.