ISP - Fall 2022 || Term Project || Anthony Lupica

Anthony Lupica's Web Page Generator 2.0

COMMANDS

Item	Description	Action
Heading	Add a heading to the page	addHeading
Text	Add paragraph content to the page	addText
Background Color	Change the background color of the page	
Text Color	Change the color of an element by id	
Text Alignment	Change the alignment of an element by id	-- Select Alignment --
Font Family	Change the font of an element by id	-- Select Font --
Font Size	Change the font size of an element by id	px(1 - 50)
Reset	Reset the design window, and return all styles to default settings	reset
See raw HTML	In a new window, view the raw contents of the page	raw
Close	Close the design window	close

OUTPUT

Thesis

My goal heading into work on this term project was to perform a perfective sprint on my web page generator from pa3.

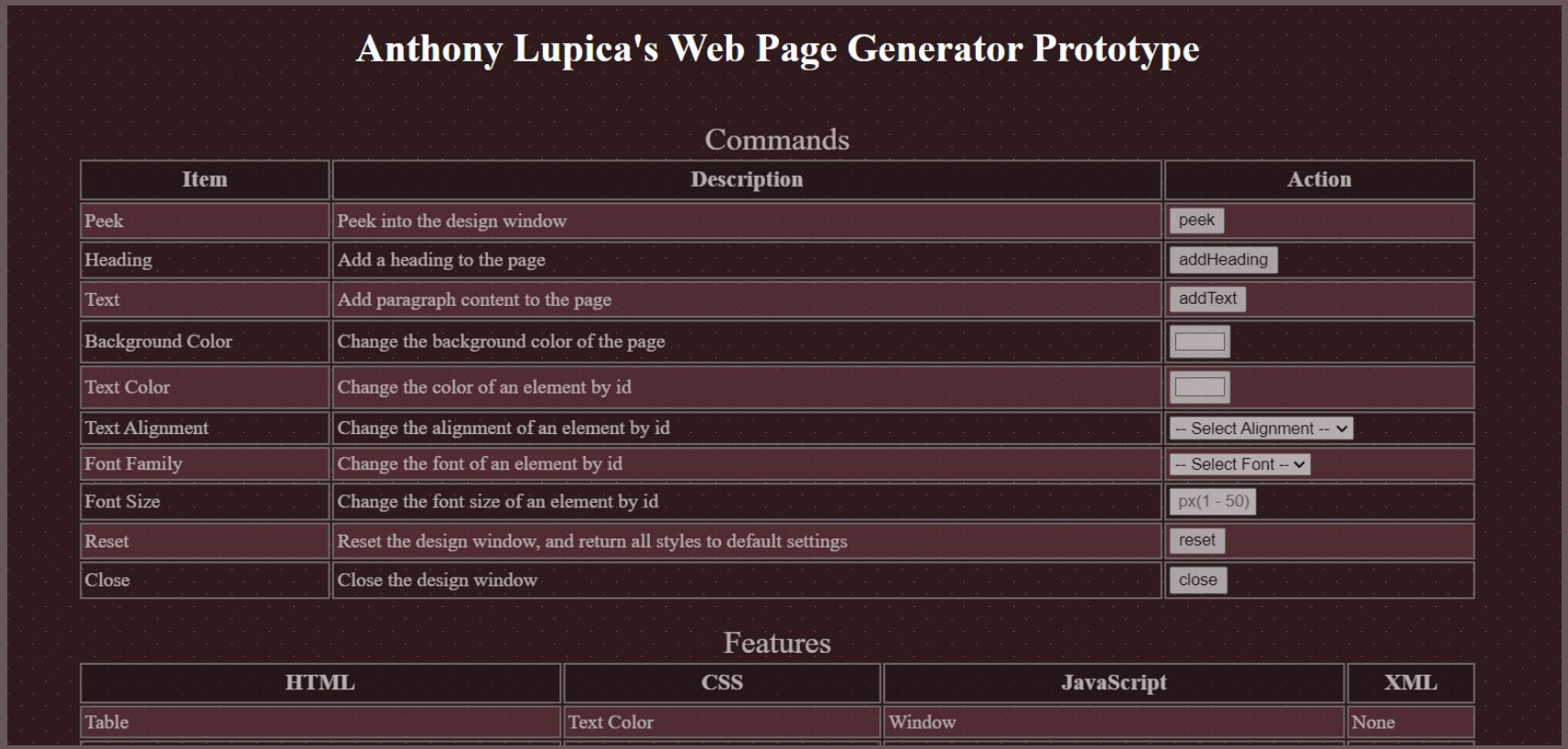
The original application was lacking in some key areas:

- Content that the user produced could only have been seen by them if they pressed a button to do so, which opened a separate window. Intermittently, they would be given no real-time feedback on their designs.
- The UI was dull, and inefficiently arranged. It needed to be completely reworked.

And there were opportunities for growth in others:

- There was no way for the user to save their progress
- The user could not view their document in raw, unrendered, HTML

Old Design



Anthony Lupica's Web Page Generator Prototype

Commands

Item	Description	Action
Peek	Peek into the design window	peek
Heading	Add a heading to the page	addHeading
Text	Add paragraph content to the page	addText
Background Color	Change the background color of the page	
Text Color	Change the color of an element by id	
Text Alignment	Change the alignment of an element by id	-- Select Alignment --
Font Family	Change the font of an element by id	-- Select Font --
Font Size	Change the font size of an element by id	px(1 - 50)
Reset	Reset the design window, and return all styles to default settings	reset
Close	Close the design window	close

Features

HTML	CSS	JavaScript	XML
Table	Text Color	Window	None

New Design

The output window is an <iframe> element, and requires no button pushes. It also updates in real Time, and can be dynamically resized by dragging the bottom right corner.

New Feature

The ability to spawn a popup window to view the page is not gone completely. . .

The user can now click the button titled “raw” to open a separate page and view their web page in unrendered form.

- All content that has been added/changed since the user last performed this operation is overwritten to replace the out-of-date version.
- The window displays the HTML code in a well-formed, properly-nested style.

```
<!DOCTYPE html>
<html>
<head>
  <style>
    #h1
     {
       color : black;
       text-align : center;
       font-size : default;
       font-family : default;
     }

    #h2
     {
       color : black;
       text-align : center;
       font-size : default;
       font-family : default;
     }

    #h3
     {
       color : black;
       text-align : center;
       font-size : default;
       font-family : default;
     }

    #h4
     {
       color : black;
       text-align : center;
```

Some Implementation Details

As previously stated, the most obvious piece of the UI design – the embedded window –
Was accomplished using a single <iframe> element. These are specifically designed to embed one document inside of another. There were more difficult ways to do this, but <iframe> seemed to be the surest bet.

The ability to output a formatted, unrendered HTML document version of the user's current design
Was accomplished largely by leveraging string methods such as concat() and replace(). With these, it was possible to form a large string with specific characters I could later find and replace.

- Notably, the replace function looked for and converted characters as such . . .

```
'<'  : '&lt;',        // opening, closing and &, replaced by entity references
'>'  : '&gt;',
'&'  : '&amp;',
'\n' : '<br>',      // \n used to embed a break
'\t' : ' '   // \t used to embed a whitespace *
```

*A non-breaking space, also called NBSP, required space, hard space, or fixed space, is a space character that prevents an automatic line break at its position.

Issues and Concluding Remarks

My strategy of string replace was rather obtuse, and I know there is an XML solution out there that is simpler and much more extensible. However, I'm happy with where I landed for this project

For this sprint, I was also unable to include importing and exporting of HTML files, which was one of my initial goals. The skills honed here and lessons learned, though, would serve me well in the future to bring these features to the product.

Demo