Application Project

# General Description

The application is a What You See Is What You Get text editor which runs in the browser. On the left hand side of the page is the actual editor with a document view. On the right hand side is a pane to present export options to the user

*Example Layout:*



# 

# Front-end Specification

The front end of the application should have the following qualities:

* The top bar should show all available formatting options, each of which the user should be able to set/toggle
* In the main editor window, user keystrokes should be inserted into to the underlying *RichText* in the current formatting style
* The editor window should display where the user is currently typing with a standard cursor bar
* When the user selects text, it should be highlighted, and they should be able to delete text, replace text, or replace formatting.
* When the user changes position in the document, the formatting options on the top bar should change (possibly the options on the bar are controlled by the back end)
* The output menu should display a dropdown menu of all supported output formats
* The output menu should have a button to download the current document in whatever output format is currently selected
* Possible preview for some formats

# 

# Back-end Specification

The back-end of the application should do the following:

* Provide the front end with a list of formatting options when the program starts
* Provide the front end with a list of output options when the program starts
* Rendering the text (and the text input cursor) - either the back end should render to html and we should have a javascript interface to replace the current document, or we should have a [canvas](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D) which we render to
* Render the output and export it as something the front end can present for download (like a byte buffer or [Blob](https://developer.mozilla.org/en-US/docs/Web/API/Blob))
* Store the current formatting options that the user is entering, and update them on the page

# Expected Challenges

* Syncing the front end and the back end views of the text. We could have actual HTML nodes showing the text and JavaScript or C++ which calculates the needed changes each time the user makes a change (though this seems hard)
  + We could also have a canvas element like how Google Docs does things, though this also poses its own challenges such as which side should actually be doing the rendering (and if that is C++, how to pass it the canvas context)
* User interface design since we are all programmers :P
* Downloading the file will require some more interfacing between JS and C++ depending on the file type. We will likely need to get a byte buffer of the file’s contents and store it in a Blob object which can then be downloaded via a <button> or <a> element
* Input handling: do we want to have shortcuts such as Ctrl+I for italics or Ctrl+B for bold?
* Devising an input library in JS/CSS which can handle generating good looking settings or widgets for the top bar and output view out of whatever the back-end hands us
* Highlighting text making sure it knows where in the RichText the start and end of that highlight occurs.

