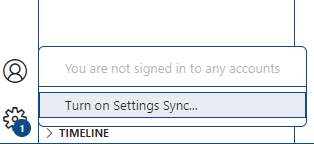
**VS Code Features**

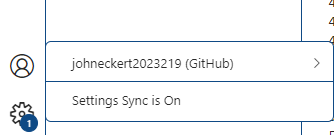
Since Atom is being shut down, we have had to migrate to a new app for our coding: VS Code. Some are sad about it, but there are ton of useful features that can make VS Code even better than Atom.

**Feature One: Git Integration**

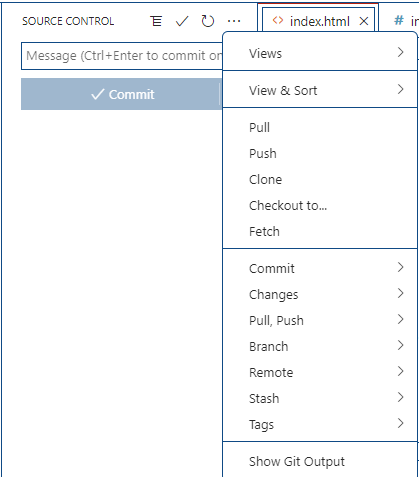
Visual Studio has an amazing feature called Git integration. Using this, you can do everything you would do in GitHub Desktop all through VS Code. To do this, open the source control tab on the side of your VS Code column. When you first open up VS Code, you should be prompted to sign in with your GitHub account. If you chose not to use your GitHub account, or you signed out of it, you can click on the bottom left profile icon. There will be an option called ‘turn on settings sync’ and it will allow you to sign in with GitHub. (This is not the only way, but Settings sync is useful to have in of itself, so it is better to do it through setting sync).



Open up the GitHub tab it prompts, sign in to your GitHub account, and wait for it to tell you you can close the tab. VS Code should then tell you that you are signed in with GitHub.



Once signed in, you enter the source control panel on the left-hand sign. Click on the 3-dot icon, and it will open a drop-down menu. You can pull, push, clone, commit, merge branches, stash changes, add or remove your remote, create tags, and more I might have forgotten to mention.

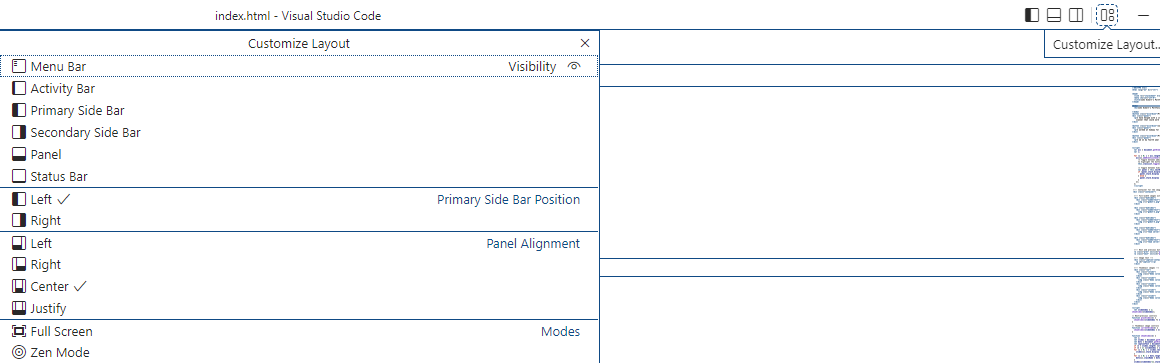


It is incredibly useful to use, as you won’t need to open GitHub desktop and can have less windows cluttering your computer screens. I recommend using GitHub desktop for changing your remote, merging branches, and all other highly important / visual issues, as it is very easy to mess up and GitHub desktop still makes these things easier, but doing commits and small things in the middle of your code on one project help you stay on task and feels good to do.

(If you are having trouble with committing, all you have to do is type what you want the commit to be called below all the commented-out text. Once you have it named, click the check mark at the top right of your screen, and it should be committed. You can open your GitHub to check your commits if you still do not trust that it went through).

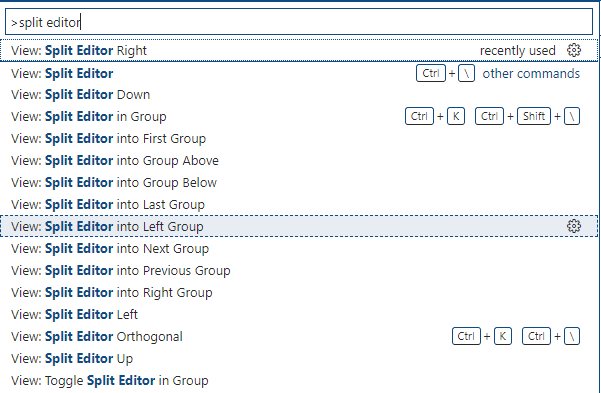
**Feature Two: Zen Mode / Split View / Beautify**

If you’re coding but you find the side bar distracting or just too space consuming, Visual Studios allows you to hide it from site. Although it is not some game changer on it’s own, it can help de-clutter your screen and allows for another feature to be paired with it really well. To turn zen mode on, click on the ‘customize layout’ button on the top right (to the left of the minimize button)



Near the bottom of the list, there is a button called ‘Zen Mode’. It will hide everything from site except for the current coding window and the customize layout drop down. If you close the drop-down window and then wish to end Zen mode, click ‘esc’ twice and Zen mode will close.

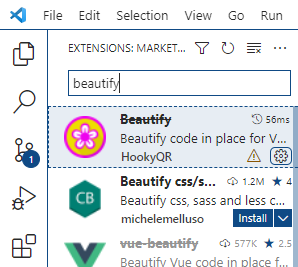
The other feature that I think pairs really well with Zen mode is the split view. Using split view, you can have multiple files open at the same time, so you would be able to code or use one as a reference whilst having Zen mode activated. It helps with constantly swapping through files to use two at once, and for projects like the formbar in which there are multiple files to sift through, you can two visible at once. This was also a feature on Atom, and one I used an uncountable amount of times. To activate split view mode, just press Ctrl-Shift-P and either search for ‘Split editor’. It will give you multiple options, so choose the one you like best, however I recommend split right / left.



There are two other, quicker ways to split your screen, however these will always be split to the right. On the top right side of VS Code, there is two buttons below the window closing button. The one on the left allows you to quickly split your screen, or you can use the keyboard shortcut ‘Ctrl-\’. Make sure to use the backslash, as the forward slash comments out the current line you are clicked on.



Another useful, built-in feature that is in VS Code would be Beautify. Beautify is a feature that was also in Atom, and it makes your code more legible without messing anything up. If you have weird indents or no breaks in your code, it will correct these issues and allow your code to be read by someone unfamiliar in a way easier manner. In order to use beautify, you first need to install an extension. Click on the four blocks button on the left-hand side of your VS tab, or press Ctrl-Shift-K, and search for ‘Beautify’. Press the blue install button and wait a couple seconds.



Once installed, press Ctrl-Shift-P and search for Beautify. Click on ‘Beautify File’ and watch as it rearranges your code.

**Feature Three: Keyboard Shortcuts Reference**

With all these features and shortcuts you can use, there may be some features overshadowed or hard to memorize. VS Code has already thought of this situation, and has made a PDF file for you to look at in case you forget something. In the PDF, it shows a large number of keyboard shortcuts you can use. It tells you the button input, what it does, and what category they believe it would be categorized under. It’s highly recommended for beginning with VS Code and all their shortcuts, and even if you don’t think you’ll use 90% of these, I still recommend glossing over it and seeing if you find any useful shortcuts you didn’t know about.

