Project 8 – GitHub & Personal Profile Webpage

Learning objectives for this assignment are to demonstrate the following skills:

* Using git and github
  + Learn to use basic git commands
  + create a github.io repository, and upload a webpage.
* Learn to use UX Design tools like Figma to create a webpage that follows a design that has been outlined for you.
* Build a basic professional profile that you can use on your resume.

# (40 points) Github setup

Begin by creating a profile on github.

1. Go to <https://github.com>
2. Create an account.
3. Don’t create a repository yet. I’ve got instructions for that, in a bit

## Install Git

Install the software for git on your computer.

<https://git-scm.com/downloads>

Launch the installation wizard, and use all the default settings.

Once installation is complete, open up a command prompt or terminal window

* **Windows:** Click the start menu, type cmd, then enter.
* **Apple:** (two options)
  + Click the Launchpad icon in the Dock, type Terminal in the search field, then click Terminal.
  + OR: In the Finder, open the /Applications/Utilities folder, then double-click Terminal.

Type the following command to make sure that git has successfully installed:

git --version

You should see something like this:

C:\Users\tabbott>git --version

git version 2.17.1.windows.2

C:\Users\tabbott>

Now type the following two commands, supplying your name and email address. Your name must be in double-quotes:

git config --global user.name "Your Name"

git config --global user.email your-email@whatever.com

Git will run both of these commands and print out no response if it was successful.

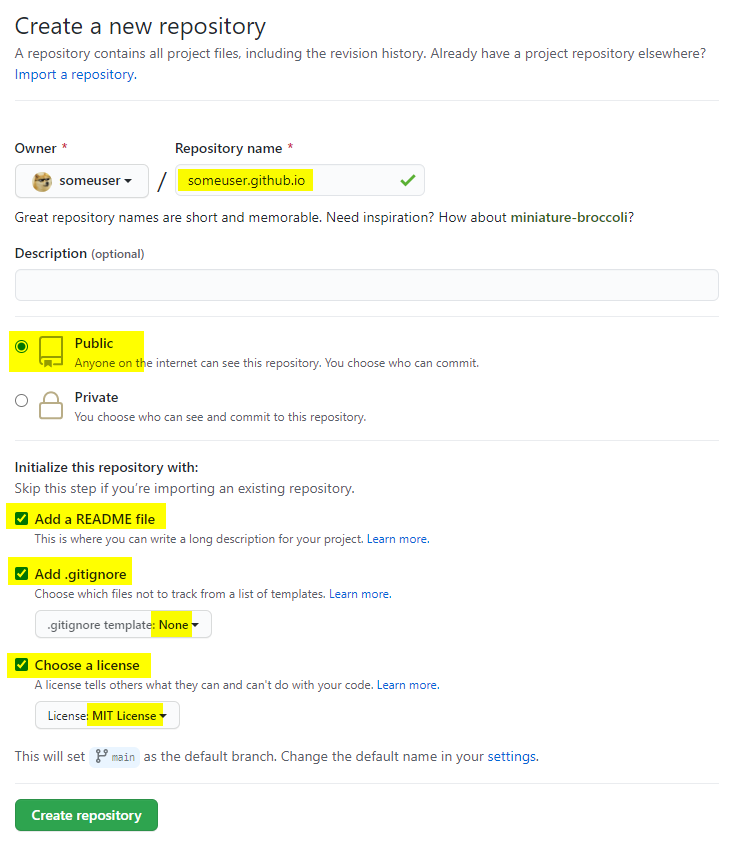
## Create a repository on GitHub

You have two options for this portion of the assignment

* (recommended) Create a github.io repository
* Use another repository, and have your netlify account pull from that repository. I will not help with this option.

There are only two hard requirements for this assignment. You must use Git, and your repo must be deployed to a website.

Log into GitHub and create a new repository. On your main homescreen, in the upper left, you’ll see a green button to create a new repository. Click it, and fill out the form, as follows:



Once you’ve created the repository, open up a command window

1. change directory to the folder where you keep all your projects on your hard drive,
2. type the git clone command
   1. supply your github username
   2. supply the name of the repo that you just created.

Here is an example:

cd <your-projects-folder>

git clone https://github.com/your-username/your-username.github.io.git

Here is an example of what I did for my repo. I’ve highlighted the two lines that I typed in yellow:

C:\>cd users\tabbott\projects

C:\Users\tabbott\projects>git clone https://github.com/twabbott/twabbott.github.io.git

Cloning into 'twabbott.github.io'...

remote: Enumerating objects: 23, done.

remote: Counting objects: 100% (23/23), done.

remote: Compressing objects: 100% (17/17), done.

remote: Total 23 (delta 6), reused 21 (delta 4), pack-reused 0

Unpacking objects: 100% (23/23), done.

C:\Users\tabbott>\_

You should see something that looks like the output above. My new repo now lives on my hard drive at

c:\users\tabbott\twabbott.github.io

NOTE: When you clone a repo, git will always create a new folder that has the same name as the repo you’re cloning.

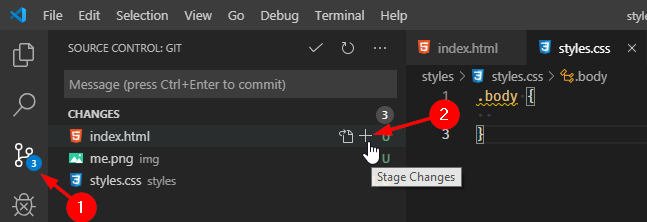
If all this is still confusing, I made a video. Watch the video and it should make sense.

## Commit often

As you work, remember to commit often. This is a good habit to get into, as it will allow you to roll back your changes if you mess up.

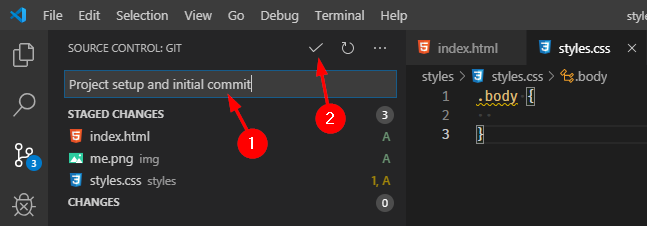
If you don’t like working with the command line, you can commit your work using VS Code. Click the source control icon on the left. You will see all the files that you’ve changed.

As you mouse over each file, a + icon will appear. Click the icon to stage each file that you want to commit.



Once you’ve staged all your files for commit, type a message. This is a note to your future self so that you can keep a journal of what you did in each commit. This can be as short or as long as you like, but it should describe what you did. Don’t write a novel; one sentence will suffice.

When you’re done, click the check-mark icon at the top of the panel. This will commit your changes locally. The source control panel will clear out and reset to a blank list.

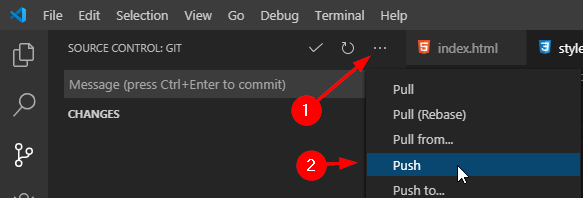


You should do this often while you work. You don’t need to commit every 5 minutes, but it’s definitely a good idea to commit every time you complete a section of work.

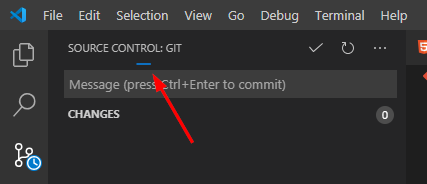
## Push your code to GitHub

Once your project is finished, you’ll need to perform one final step and push to github.

1. On the source control panel, click the elipses menu (The … icon).
2. From the menu, select Push.



You’ll see a busy-indicator slide along the top of the source control panel while VS Code does the push. When your code has been pushed up to GitHub, it will go away.



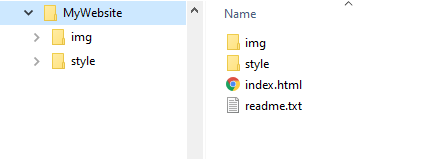
# (60 points) Create a professional profile webpage

## Project folder and file structure

Follow industry-standard conventions for creating a folder structure for your project and naming your files:

1. Your project’s main HTML page should be called index.html and it should live in your project’s root folder.
2. If you have any images, put them in a sub-folder, called images, or img, or pictures, or something along those lines.
3. If you have any external css files, put them in a folder called styles, or css, or anything along those lines.

When finished, your project’s folder and file makeup should look something like this:



## Related files

You need to have the following project related files. GitHub will create these files for you if you followed my instructions above

### .gitignore file

Your .gitignore file tells git which files or file-types it should ignore. This file should support multiple platforms (Windows and Mac).

Here is a basic template. You can add anything else you like, but it should include this:

# Windows

Thumbs.db

Desktop.ini

# Mac

.DS\_Store

# VS Code

.vs

.vscode

# Webstorm

.idea

### README.md

This file should contain your name and a general description of what styling items you decided to add to your project. See section 2.4 (Styling and appearance) for more info.

DO NOT mention this course name or course number (CS 2550)

## Content (40 points total)

For this portion of the project, you will use Figma to follow a design layout from a UI/UX designer. As a developer, you will rarely be in charge of the actual look and theme of your project. Instead, you will be given a set of set of high-fidelity mocks. Usually these are delivered via one of the following professional online tools:

* Figma
* Adobe XD
* Zepplin
* inVision

For this assignment, we will use Figma. Go to this link, and you can see the design you will use:

<https://www.figma.com/file/IEJFxrlbk8SdqjYkZpzIrZ/2550-project?node-id=0%3A1>

The background color for your page should be a light grey. Use #F0F0F0, or rgb(240,240,240). The font color should be black.

All four sections are meant to follow one after the other in one continuous flow (no borders) as the user scrolls down the page. Here is a thumbnail screenshot of one possible configuration:



The figma mock gives you several options for each section, so you have a lot of flexibility in what kind of look you want to go for. Your profile must have the following four sections:

* A header that has your name, a menu bar along the top, and a splash image.
* A brief about-me section.
* A projects section that lists some of the projects that you would like to show off to a hiring manager.
* A footer that links to your social media profiles.

### (10 points) Page Header

The page header should be 100% witdth, and 600px high.

The page header will have a menu along the top with the following links:

* About
* Projects
* Contact

Clicking the menu will scroll the page down to whichever section that the user clicked. It will not take you to a separate page.

Your name should be displayed in a large font size (36pt, 48pt, etc). Use flexbox to center it vertically and horizontally (Look up on Google: “how to center a div”)

You will pick one of the two header designs.

* Design 1: **Hero image**.
  + You may use a nice background photo that you took yourself, or you may go to unsplash.com and find something there.
  + If your photo is light, then use black for your name. If your photo is dark then use white.
  + The image should cover the header area regardless of how wide the browser’s window is.
  + The image should not be distorted or stretched.
  + The image should not be tiled.
* Design 2: S**imple**: this design is more conservative. It will have an image of you on the left, and your name on the right.
  + Use CSS to give your portrait a circular frame.
  + Use a professional photo, or a photo that would give a potential employer a good impression of you.
  + To space out your photo and your name, use justify-content: space-between;

### (10 points) About me (two or more sections)

Each About Me section should be 100% wide and as tall as it needs to be. Set the padding (not the margin) to 48px;

Pick one of the designs for an About Me section

* Design 1: with photo
  + Choose this design if your page header uses a splash image, or if you want to show two photos of yourself.
  + The section will have the title “About Me”, then a photo, then a one to three paragraph blurb about yourself. Use flexbox to do a column layout, and center everything horizontally.
  + Your photo should be 400px by 400px square, framed and cropped neatly,
* Design 2: no photo
  + Choose this design if you already have a photo in the header section.
  + List 3 to 5 things about yourself in a left-justified bulleted list. Use a 64px margin on the left and right sides.

You can have multiple about-me sections, but you should keep to the following guidelines:

* Each one should follow a theme and have a single subject: personal info, work history, travel, outdoors, sports, etc.
* The first one should be about you, personally.
* Each section should have an item on the nav-bar. Clicking on the nav-bar should take the user to that section.
* You can have as many as you like, but you only get 10 points total. Two or three sections is ideal.

When telling about yourself, pick topics that a team of programmers might like to know about you. List your hobbies, your personal interests, places that you’ve lived or travelled to, books and movies you like, etc. Keep this portion neutral and businesslike.

Include your degree and where you went to school.

You might also list any work history that you have, as long as it’s tech-related. List the company, your position title, a short description of the main project your team owned, then a comma-separated list of any languages / frameworks / tools you used:

* Micro Creations, junior front end engineer. Tax assessor audit and review website. PHP, MySQL, jQuery, Python.

You have a lot of flexibility in this area. The mocks are more to give you guidance on how to lay things out. Feel free to fill up a page if you like, but it does not need to be as long or as complete as an actual resume.

Do not put any personal contact info here. Save that for the footer.

### (15 points) Projects

Pick one of the designs, and list any projects that you can show off to a hiring manager. This is where you will showcase your best work. You may use projects from this class, or from any other class, or any personal projects that you would like to show off.

Each project should have its own card. Each card should have the following

* Some type of splash image. This can be a screenshot, or any kind of relevant image from unsplash.com.
* A project title
* One or two sentences that describe what the project is, what kind of problem you were trying to solve, and what languages, frameworks, or tools you used to solve it.
* If your project is a web page, then you can include a link that takes the user to the page. If you need to, you can use netlify.com to host your website.

This section can be as tall as you like but it should have top, bottom, left, and right padding of 48px.

Warning: do not mention UVU or the course names or course numbers, as this information is searchable and will be used by cheaters. If it is discovered while you are a student that other students are using your work for their own projects, you will be held accountable.

### (5 points) footer

Your footer should have icons for your contact info. You can get your icons from fontawesome.com (figure out how to do this; that’s what programmers do). This section should have 48px of padding on all four sides.

At the very least, you should have the following links:

* Email: use a mailto link (look up how to do this in HTML)
* LinkedIn
* Github

Include any other social media platform you want an employer to know about (they will look you up, regardless)

* Instagram
* Twitter
* Facebook

## (20 Points) For additional credit

### (5 points) Attractive color scheme

Go to Google and look for a website color scheme picker, and pick a color scheme.

* Your website’s main color will be used for the background.
* Pick a high-contrast color for your font. You’re going for readability.
* Pick a secondary color that compliments the main color, and use it on your menu, and elsewhere in your page when you want a “color-pop”.

### (5 points) Attractive use of fonts

Pick an attractive font to use for your headers. This can be something fancy. Use the same font for all section headers.

Pick a readable font to use as the text of your page. You don’t have to use a sans-serif font, but to get full credit you can’t use the browser’s default font.

### (5 points) Banded website

Build a website that uses banded sections. Each section will be enclosed in its own container element. You can use <section> or <div>, then apply a background color.

The sections should alternate between your site’s primary and secondary colors, or a parallax image

Here is an example:

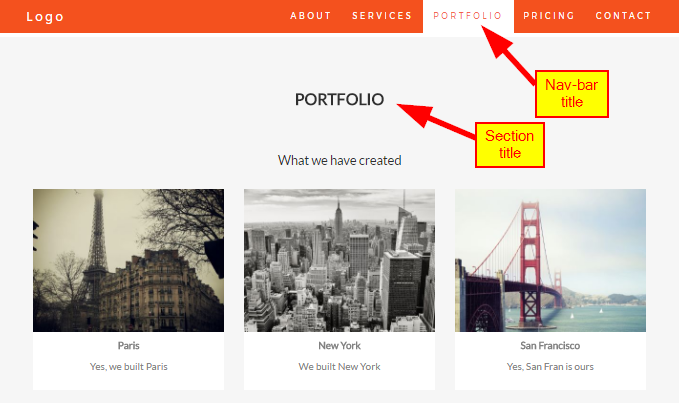
<https://www.w3schools.com/howto/howto_css_parallax.asp>

### (5…20 points) Floating sticky nav bar

Use javascript and CSS to enhance the nav menu for your site

* (5 points) **make the menu sticky:** The menu should remain attached to the top of the page as the user scrolls down.
* (5 points) **smooth scrolling:** go to google and look up how to do smooth scrolling in CSS, so that when the user clicks on your page’s menu, the page will smoothly scroll to the section instead of snapping instantly to the section.
* (5 points) **highlighted menu:** When you click an item on your menu, change its background to be a lighter version of your secondary color, and un-highlight the previously highlighted menu item.
* (10 points) **auto-highlight menu**: The menu should highlight each item as the user scrolls through that section, and as a section scrolls up past the top of the viewport, the menu item for that section should become highlighted,

Here is [an example on w3schools](https://www.w3schools.com/bootstrap/bootstrap_theme_company.asp) what I’m looking for:



### (5 points) Parallax background

Do a parallax background on one of the sections in your profile. The header is a good choice, but you could probably pull this off. Don’t overuse this effect. It’s like cooking with garlic; a little goes a long way.

### (20 points) Rotating image carousel

Add a rotating photo carousel. You should put this in its own section. You can title the section anything you like.

* All the photos should follow a consistent theme. There needs to be a reason for it to be on your page.
* The photos should auto-scroll every 10 to 15 seconds.
* The carousel should have navigation breadcrumbs at the bottom so that the user can click to navigate to a particular page.
* There should be left and right arrows along the edge of the carousel that the user can click to navigate backwards or forwards.

# Submitting your work

When you’re ready to submit, go to Canvas and enter the following:

* Your GitHub username
* URL to the repository for your code on GitHub.
* URL to your website.
* A bulleted list of items you did for additional credit (section 2.4).

## What you may not do

You may not use any kind of ready-made website, or website generator. The purpose of this assignment is to learn how to code using HTML, CSS, and some light JavaScript.

This includes the following:

* Wordpress
* Wix / Weebly / Squarespace, etc.
* Blogspot.com / blogger.com
* Medium.com
* Any kind of social media page

## Grade breakdown

This project will be graded as follows

* 40 points: your GitHub account with a repository where this code lives.
* 60 points: your webpage hosted on a server. This can be github.io or netlify.