CIT 155

Lab 1 - My First Website and Packet Analysis

Bryan A. Knowles

Nov 4, 2019–Nov 17, 2019

This assignment will be due Sunday, November 17th, midnight Eastern time.

Labs are worth 6/50 of your final grade.

You may work alone or with one or two other students from the class. You will only get credit for what *you* upload, so if you work in a team, everyone needs to submit separately to Blackboard.

This Lab spans two weeks' worth of reading (weeks 3 and 4). So, I recommend you go through it at least twice, once after the first reading where you answer everything that you can, then again after the second reading where you answer everything else.

Post any questions you have on the material or the assignment to Ask the Instructor on Blackboard.

Instructions

Skim the Trainings section below, which has tips/instructions on how to do particular tasks. Then, complete the Assignment questions that follow. Write everything in a Word/etc. document, then submit your final draft as a Word/PDF to Blackboard.

Grading will follow this rubric:

- 0 - Student does not submit on time or submits plagiarized or unacceptable work. Double check that you have submitted the right file, as usually students get zeros because they submitted a previous week's work by accident.
- 1 - Student completes less than half of the Assignment with sufficient work, or student leaves entire Parts of the Assignment blank. Make sure that you are leaving yourself enough time to complete each Packet, as usually students submit incomplete work because they were rushed at the last minute.
- 3 - Student completes most but not all of the Assignment. Or, student completes the entire Assignment, but only to the minimal letter of the prompt, not actually applying the concepts of the week's Readings, and/or Announcements in their work. If you encounter a problem or have a question about the instructions, be sure to post in Ask the Instructor well before 24 hours before the due date, then continue to attempt to resolve the issue on your own while you wait for a reply.
- 6 - Great job, maximum points! The student completes all parts of the Assignment accurately and sufficiently and honestly, demonstrating the best of their ability.

Trainings

How Do I Take a Screenshot?

In Windows: Open your Start menu and search for the Snipping Tool. Use that.

In Mac: Cmd+Ctl+Shift+4, then Cmd+V into your Word document

In Linux: If you have a GNOME shell, I recommend the **Screenshot Tool** extension.

On a Chromebook: I recommend the **Full Page Screen Capture** extension.

How Do I open my HTML code on my Computer without a Server?

Open your browser of choice, then press Ctrl+O, then find and select the HTML file you want to open. (Macs, use Cmd+O instead.)

How Do I Host Code on my own Server?

Let's go with the simplest and free option.

We'll use **Github Pages**.

Note, when you make changes to your site through Github Pages, the old version is cached, so it may take a few minutes to see the updates, meaning it is best to work locally and upload only when you've made significant changes.

Also, setting this up will require *exact* naming of things to be correct, so triple check your spellings before hitting any Submit buttons.

Simplified steps for using Github Pages (no install required):

1. Create a Github account. For consistency in our class, use your KCTCS username for your Github username. If you already have a Github account, feel free to use that one or create a new one. I suggest creating a new one to separate your school work from your personal work.
2. From the homepage of Github (after you've logged in), press the "plus" button in the top right and click "New Repository."
3. Name the repository like this: `bknowles0000.github.io`, except with your username and not mine. **You must match this pattern exactly and use your username exactly or it does not work.** This will also be the url for your website. For example, if that were my username, you could find mine at `http://bknowles0000.github.io/`.
4. Give the repository a brief description, mark it as public, initialize it with a README, and add the MIT license. Then, click Create Repository at the bottom.
5. To upload files to your website, first visit the repo page for your site, like at `https://github.com/bknowles0000/bknowles0000.github.io`, but with your username instead of mine.
6. Then, drag and drop your files/folders from your Desktop or File Explorer window onto the browser window into the middle of the repo page. I find it best to put Chrome and File Explorer side-by-side for this.
7. When you do this, it will take you to a page to create a "commit" with those files. A commit is like a save point that you can always roll back to if things mess up. To create a commit, you *must* write a description of what your changes were; this is a practice that helps you and anyone working with you to review changes made to a large project or to find the right save point to roll back to. Commit directly to the master branch, and press the Commit Changes button at the bottom.
8. **Note: You must have a file named index.html at the top level of your code for your website.** This will be the homepage that Github shows users when they view your page.
9. Then you should be able to view your website by visiting the URL `http://bknowles0000.github.io/` (but with your username instead of mine). If the changes you

have recently made do not appear, wait a few minutes and refresh again, since Github Pages caches your site for a brief period to make content delivery faster.

What do I use to Edit Code?

All you need is a text editor, like Notepad, not a word processor, like Microsoft Word.

There are *several* options.

I recommend:

- Notepad
- Caret (chrome extension)
- Sublime Text
- Visual Studio Code (not Visual Studio)

These will let you create and edit HTML files on your own computer, which you can then upload to your server.

Assignment

Part 1 - Review of the Reading

1. What is a server?
2. What is a client? Give a few different example.
3. What is an HTML tag? Is this the same as an HTML element?
4. What tags must every webpage have?
5. Open Notepad (or a similar text editor program) and write a bare minimum webpage that just has the required tags you listed above. Make sure they are all in the right places and in the right order. Save this file as "index.html" somewhere you can easily find it, like your desktop. (If you are using Notepad, *do* include the quotes around the filename when you save it, else it might save it as index.html.txt by mistake.) Embed a screenshot of your text editor showing your HTML code as your answer to this question.

Part 2 - Guided Hands-On Practice

1. Modify the HTML file you started above. Give it a `<title>` like, "Facts about my Pets" or "All about my Favorite TV Show" or so on. Save it, then open it in your browser. Embed screenshots showing your updated code and your browser tab showing the title of your webpage.
2. Add an image to your page of your pet/tv show/etc. Then add a paragraph introducing us to your pet/tv show/etc. And then add a list of facts about your pet/tv show/etc. At least one word in your paragraph and/or list of facts should be emphasized with bold and/or italic text. Save it, then refresh it in your browser, then embed screenshots showing your code and your page so far.
3. Next, add one more semantic element to your page as you see best fit that would be meaningful for the topic your page is about, and save the file. Then, create a Github Pages and upload this file as your homepage. Embed screenshots showing: your github repo list of your files; and your Github Page homepage (like `bknowles0000.github.io` but your username nad not mine) open in your browser.
4. View your Github Page homepage in Google Chrome. Right click / inspect element somewhere on the page. This should show the DOM as Chrome understands it. Embed a screenshot showing this DOM. Next, right click / view page source somewhere on the page. Embed a screenshot showing this source code. And finally, visit **https://onlinecurl.com/** and run it on the url for your Github Page homepage. Embed screenshots showing the Reponse Header and Response Body.

Part 3 - Thoughts for Discussion

1. What is a TCP/IP Packet? (Research as necessary, cite any sources that were helpful)
2. What is an HTTP Request header?
3. What is an HTTP Response header?
4. Compare and contrast, in good detail, what you saw when you looked at Inspect Element, View Source, and Curl in Part 2 above. Also run curl on **https://cit155.glitch.me/** and compare/contrast to the results you see there.
5. Share your results/answers for questions 2.4 and 3.1-3.4 to the Lab 1 discussion board on Blackboard. Briefly compare/contrast your results/answers with what others wrote, or briefly summarize the common themes you saw in everyone's answers.