

INFO/CS 1300: Lab 7 (10/6)

By Grant Storey

Due at the end of your lab section. Discuss your work in front of a TA to get your participation credit.

Overview

For the second half of the semester we'll learn how to create web pages using the programming language, PHP. In this lab, we will get PHP up and running and also fill out the mid-semester course evaluation

What You Need

Your laptop and a printed copy of this write-up (optional).

Part 1: Mid-Semester Course Evaluation

For the first part of lab, please complete the mid-semester course evaluation. This evaluation is very important to the course staff. Much like we try to provide useful feedback during your milestones, we would appreciate feedback on how we are doing in this course. Your feedback is an opportunity for us to improve and better serve your educational needs.

To complete the evaluation, check your Cornell email for a the mid-semester evaluation form for INFO/CS 1300. Follow the instructions in the email to complete the evaluation in class.

Credit

Once you finish the evaluation, **DO NOT CLOSE YOUR BROWSER WINDOW!** You will need to **show** your section TA that you have **submitted the evaluation** to get **credit for this lab**.

Part 2: Final Project Team Survey

The final project in this class is a **group** project. The final project is a chance for you to get some real-world experience in designing and building a website. Your job is to find your own client and create an innovative, interactive, and interesting website for them.

We will assign you to a project teams of 3 or 4 persons. In order to find a good team match, please complete this survey: https://cornell.qualtrics.com/jfe/form/SV_4OuwzuNE3zyZ3gh

If you do not complete this survey, we will still assign you to a group.

Group assignments are **final**. We will not reassign you to another group for **any reason**. This survey is your opportunity to voice any concerns or issues you may have with group assignment. We will do our best to accommodate your needs, but there are no guarantees.

Part 3: Install PHP

PHP enables us to create *dynamically generated* web pages as opposed to the file-based web pages we have created thus far in the class. PHP stands for **PHP: Hypertext Preprocessor** (it's a recursive acronym). We won't use PHP right away, but it's a good idea to get it installed on your computer before we start using it in class in the upcoming weeks.

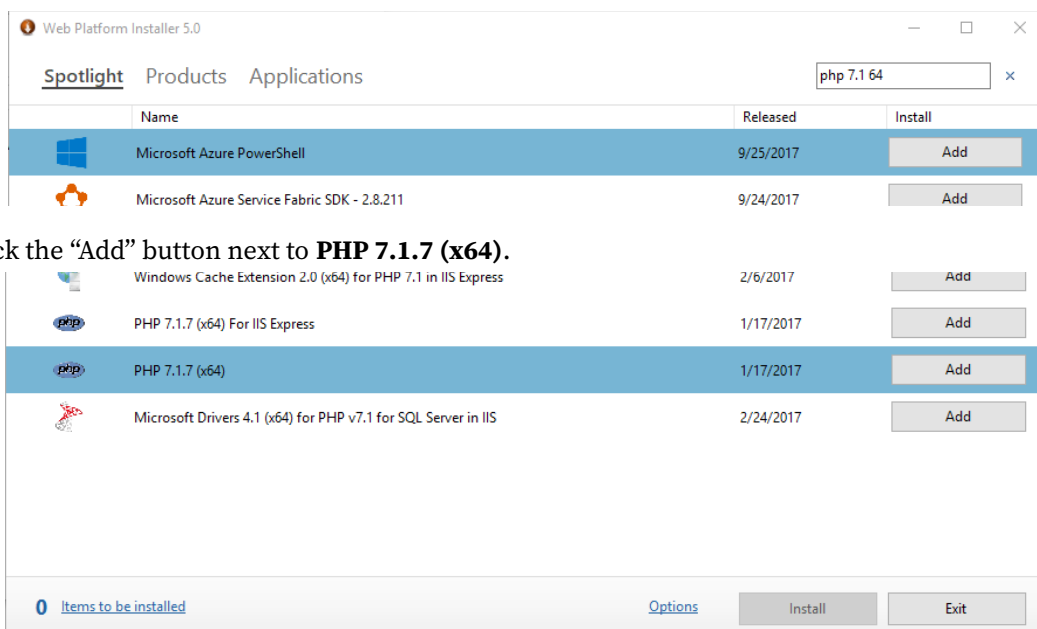
How you install PHP depends on your operating system. Read the section for your system (Windows, Mac, or Linux) and follow the instructions, then move on to Part 4.

Windows

1. Go to Microsoft's [Web Platform Install page](#).
2. Click the green "Free Download" button on the right:

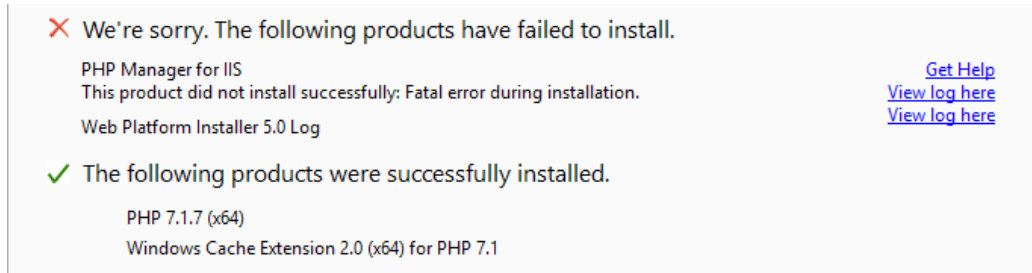


3. Once the program has finished downloading, open it up.
4. In the top right, search for "**php 7.1 64**":



5. Click the "Add" button next to **PHP 7.1.7 (x64)**.

6. Scroll to the bottom and click “Install.”
7. Click "I accept."
8. Wait for installation to complete.
9. Some platforms may fail to install; this is okay, as long as **PHP is installed successfully**.



Mac

PHP is bundled with your Mac.

1. Open Terminal (Applications/Utilities/Terminal)
2. Type “php --version” into the command line.
 1. If you see something like PHP 7.1, you are good to go.
 2. If you instead see something like, command not found..., seek help from your TAs or on Piazza.

Linux

PHP should be bundled with your Linux distribution. Open Terminal and type “which php” into the command line. If you see something like /usr/bin/php, you are good to go. If not, then you’ll need to install it via your distribution’s package management system.

Part 4: Configure Atom to Run PHP

Now that you have PHP installed on your machine, we are going to configure atom to run PHP web sites.

In this class, the teaching staff will only support Atom. You are free to use any other text editor (Sublime, Brackets, etc.) and mechanism to run PHP (MAMP, etc.) however, **we will not help you if run into issues.**

See the **lab7-demo.mp4** video on CMS for a demo of the following installation steps.

Download the Lab 7 Activity Files

On CMS, download the lab7-activity.zip and unzip it to your machine.

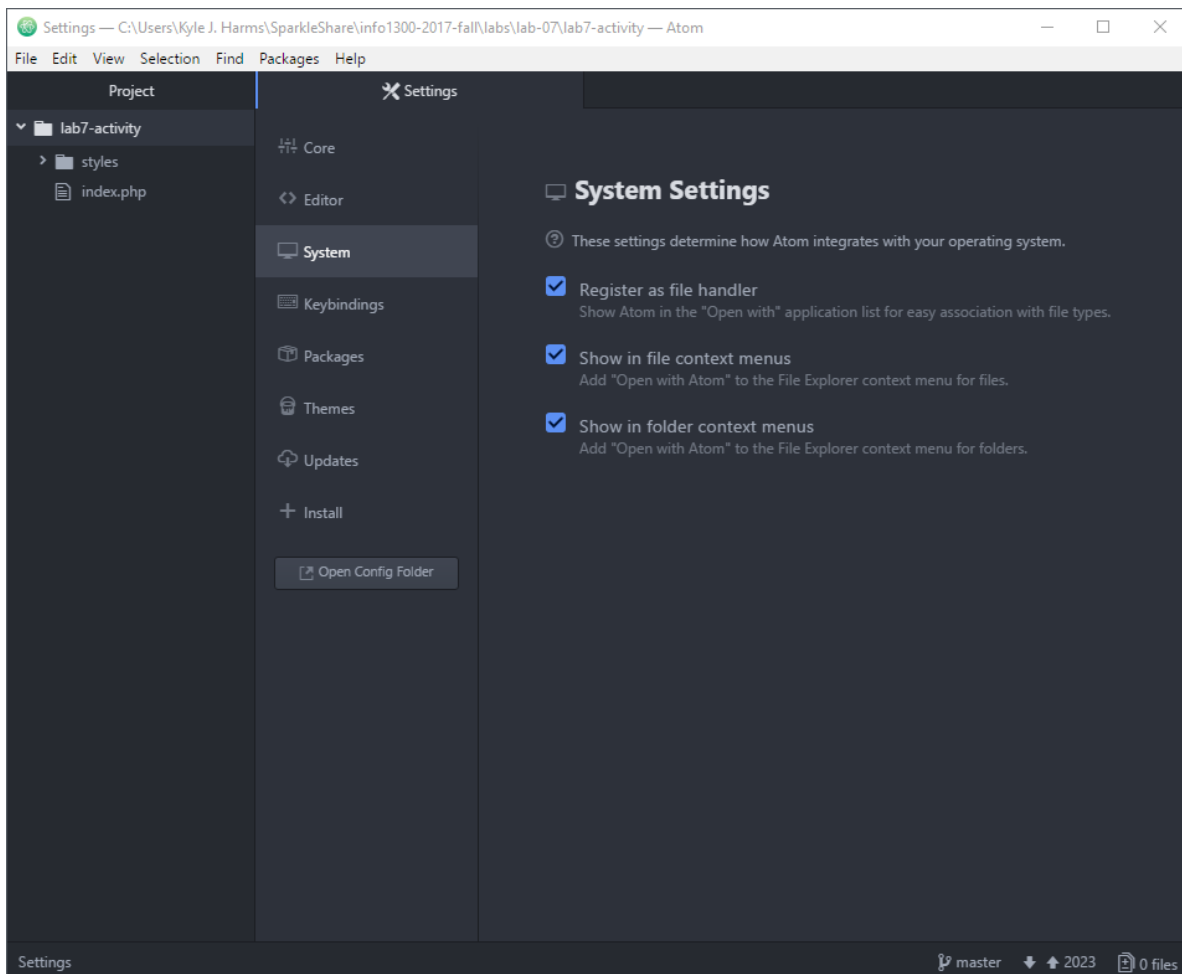
Open Project in Atom

Atom works best when we work on *projects* rather than *just opening files* in Atom. You learned how to do this in your first lab. Your instructor has also demonstrated this in class. However, many of you seem to have forgotten these lessons. You will want to follow this advice while working with PHP otherwise things may not work as expected.

Keep in mind that a **folder** on your computer (in your file manager) is a **project** in Atom.

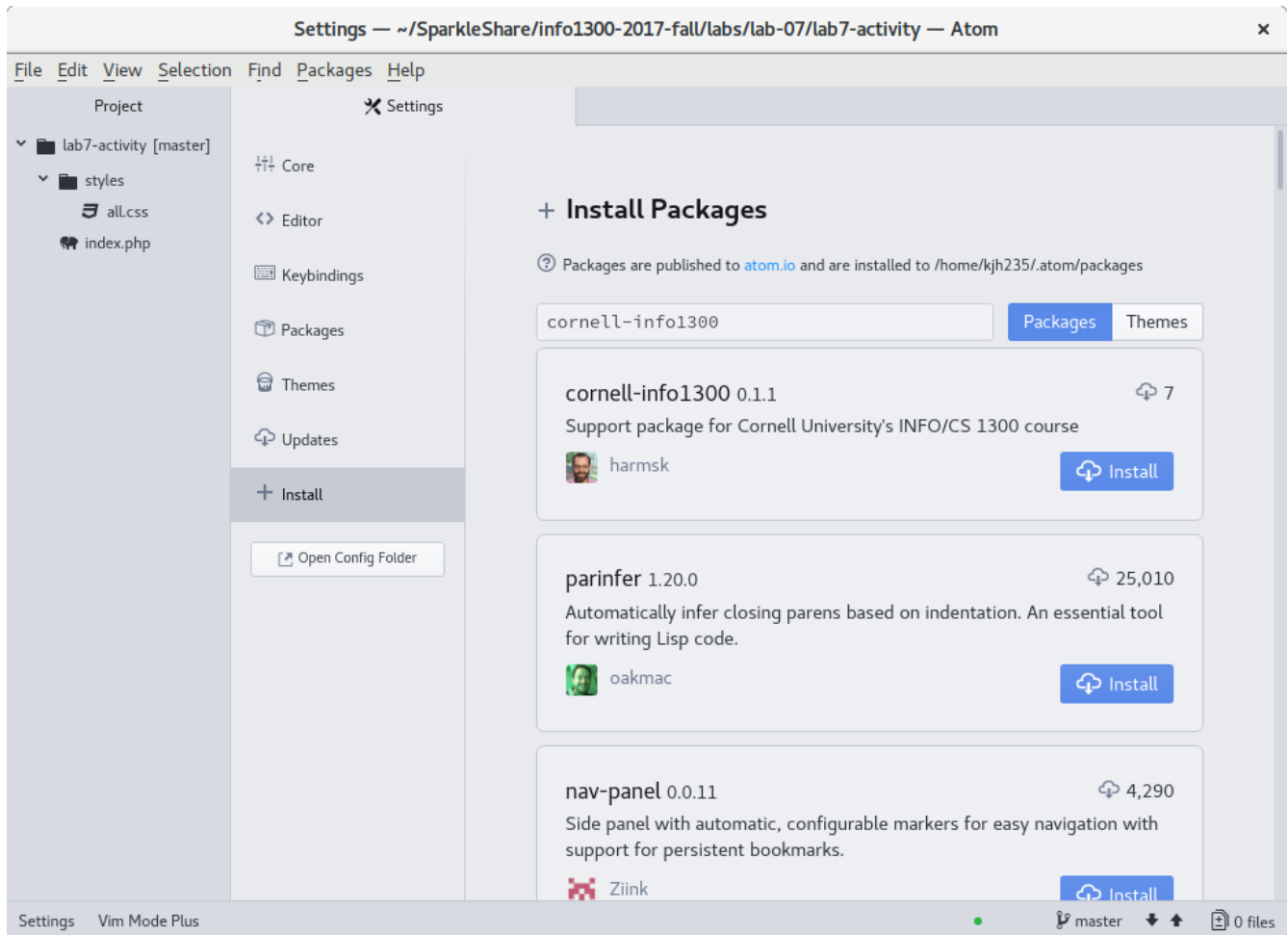
To open the lab7-activity project in Atom, right click on the lab7-activity folder in your file manger and select **Open in Atom**. If this works, go to the next section.

If you do not see this option, then you might need to turn on these menus in Atom. On Windows, open Atom, go to the Preferences or Settings, and select the **System** tab. Check the register boxes as necessary (see the figure below). For your Mac, check out this web page: <https://blog.brettski.com/2015/07/09/adding-open-in-atom-in-secondary-menu-on-a-mac/>

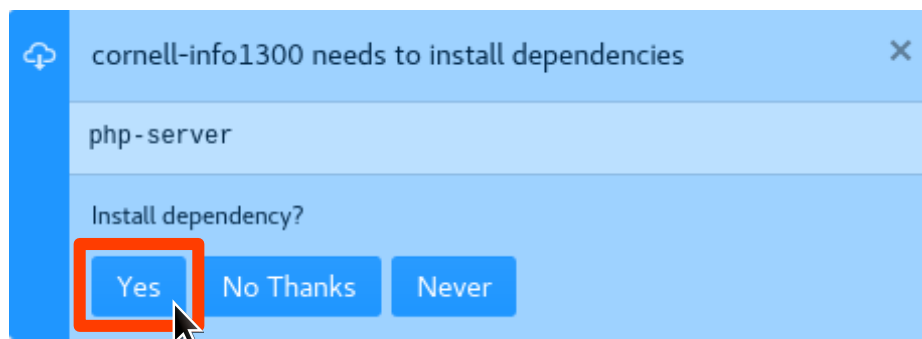


Install PHP Support in Atom

Open the **Settings/Preferences** panel in Atom, then navigate to the **Install** section. Search for the “**cornell-info1300**” package (see below) and click **Install**.



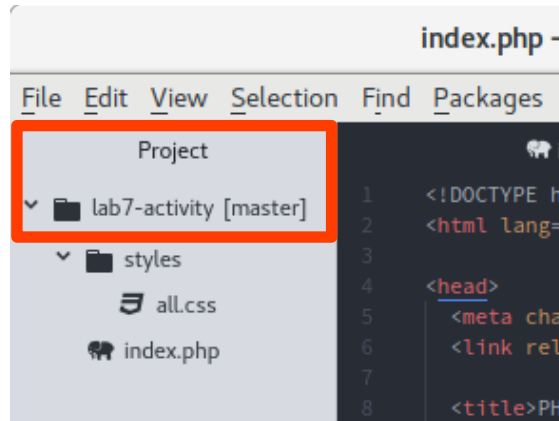
When prompted to install the **php-server** dependency, select Yes.



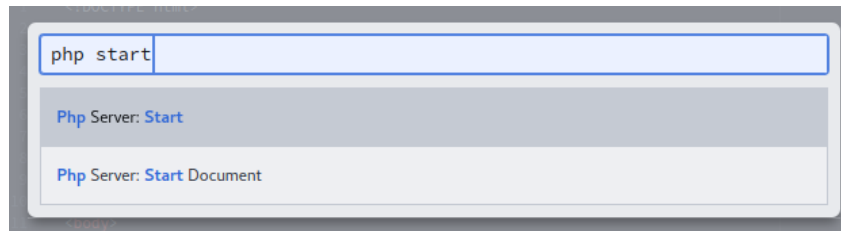
IMPORTANT! Do not install php-server directly from the install menu; the current release will not work for this class! You must use the modified version of php-server installed from the cornell-info1300 package.

Test PHP in Atom

Make sure you only have the lab7-activity folder/project open in Atom. If need help, please ask a TA.

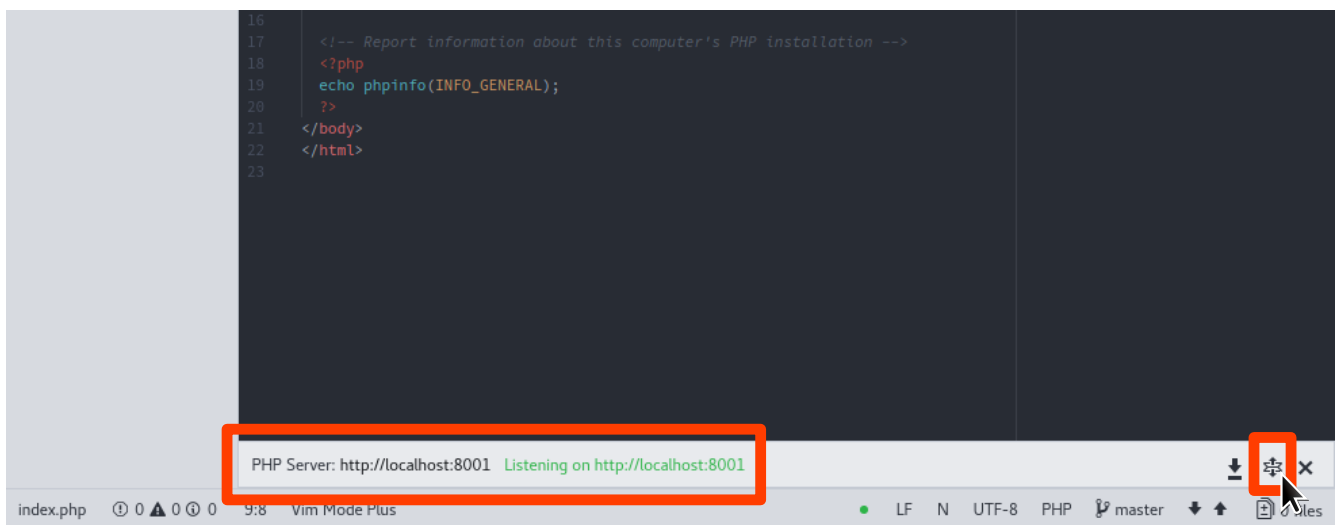


Now open Atom's **Command Palette** and search for “**php start**” and press **enter**.

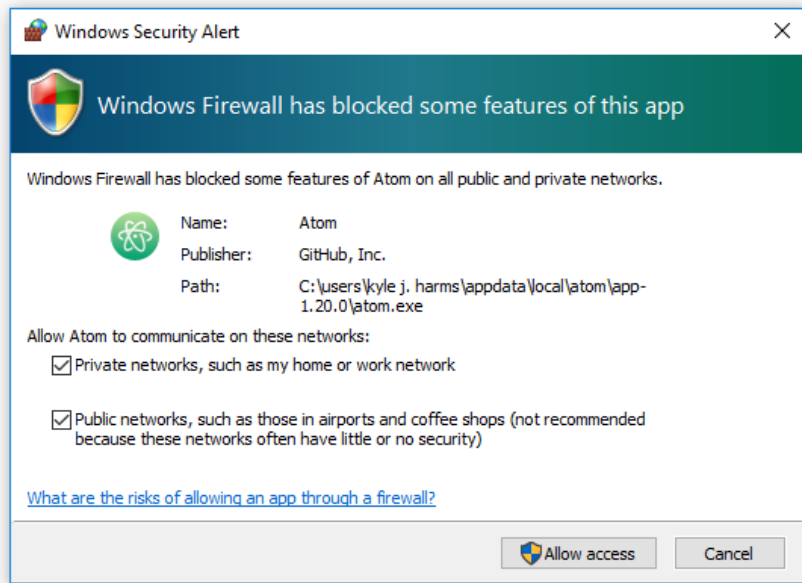


You will now see the PHP server starting in the bottom panel of Atom. See the figure below. On the left side you will see a URL for the local php server. It's usually <http://localhost:8000>, however sometimes the server might be different (e.g. <http://localhost:8001>). You will need to pay attention to this URL because it's how you will access your PHP server in your web browser. Note: you only need to start the server once when using Atom.

The PHP server panel takes up a lot of space in Atom. It really shouldn't do that, it's bad design. Let's get some screen real estate back by clicking on the collapse button as highlighted in the figure below.

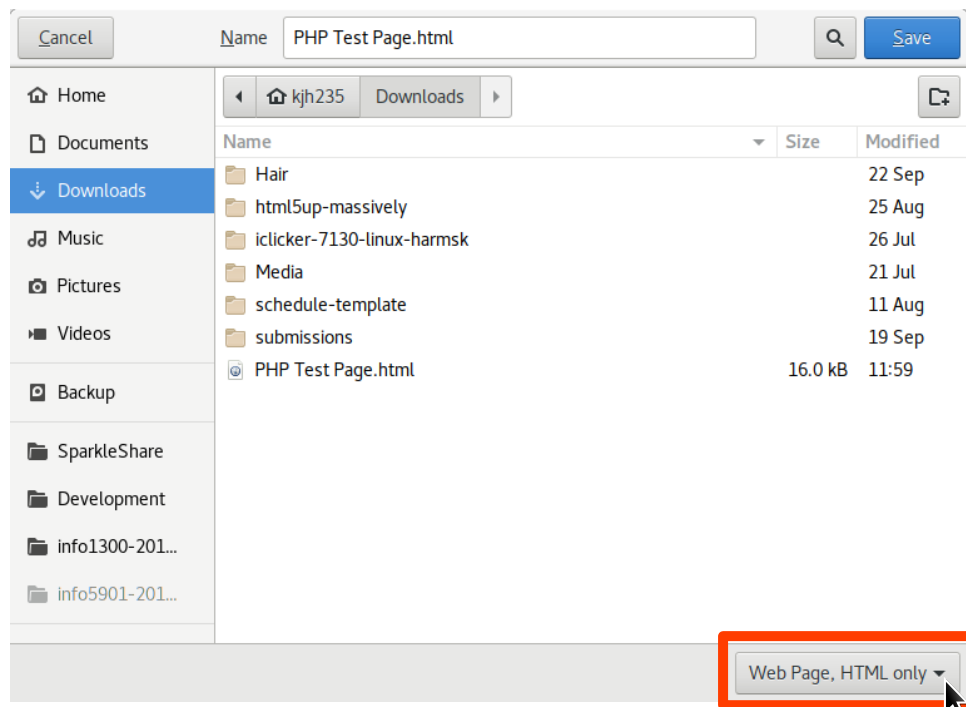


If you are on Windows, the Windows firewall may complain about Atom. Just grant it access to your networks.



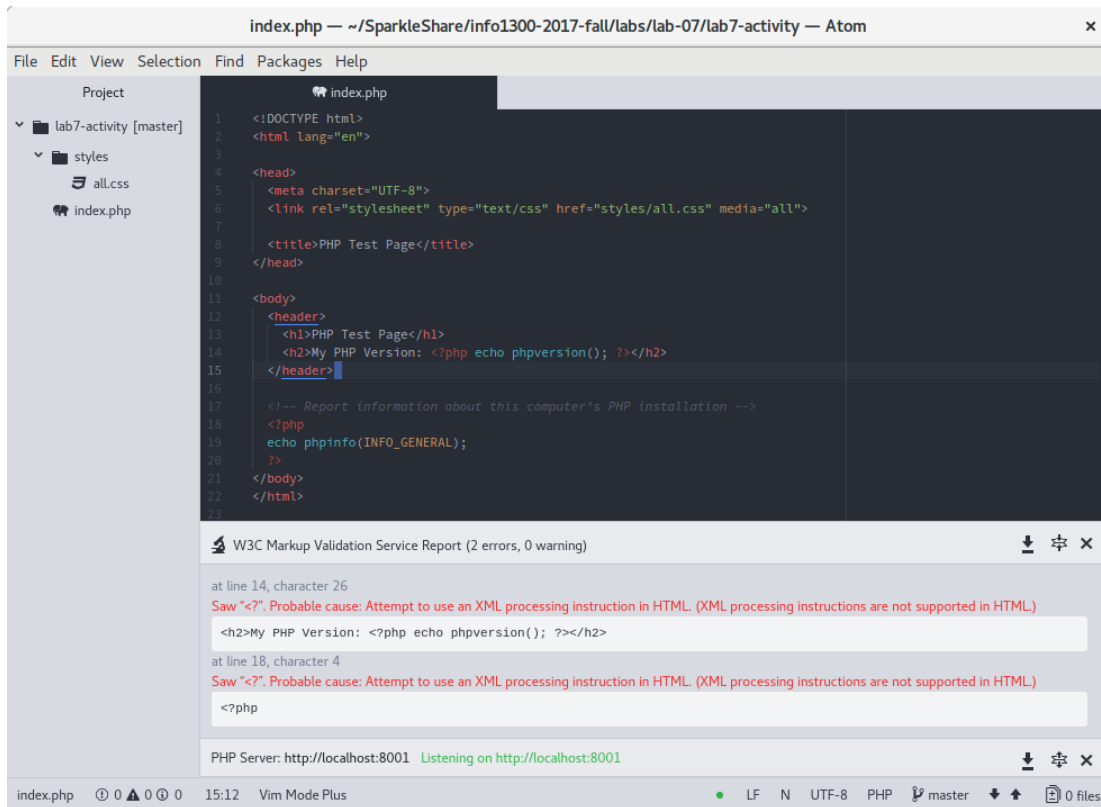
After starting the server, your web browser should have opened a page that says “PHP Test Page.” This test web page was generated by the PHP code inside of the **index.php** file in your lab7-activity project!

If the web page opened in Chrome, close it and **visit the URL of your local PHP server in Firefox**. Now save a copy of this generated web page and submit it to CMS. From the **Firefox menu** select “**Save Page**.” Next, change the format in the Save dialog to “**Web Page, HTML only**.” Upload the saved file to CMS in the Lab 7 assignment.



Note About HTML Validation and PHP

You will learn much more about this in lecture. However, for now you may be very concerned that the Atom HTML validator claims PHP code is invalid. This is because PHP code is not valid HTML code. You may get validator warnings like this:



These errors are normal. We'll learn more about this in the upcoming weeks. Please just be patient.

If you are curious about what “echo”, “phpversion()”, and “phpinfo(INFO_GENERAL)” do, we'll cover this in more depth later. The short answer is that “phpversion()” and “phpinfo(INFO_GENERAL)” produce information about the version of PHP that is being used, and “echo” puts this information inside the HTML at the location of that PHP snippet.

When you are Finished

When you are finished with the course evaluations, have PHP working in Atom, and have submitted the PHP page to CMS as specified in section 3, you are free to leave. Enjoy your fall break! Y'all have worked very hard and you deserve it!

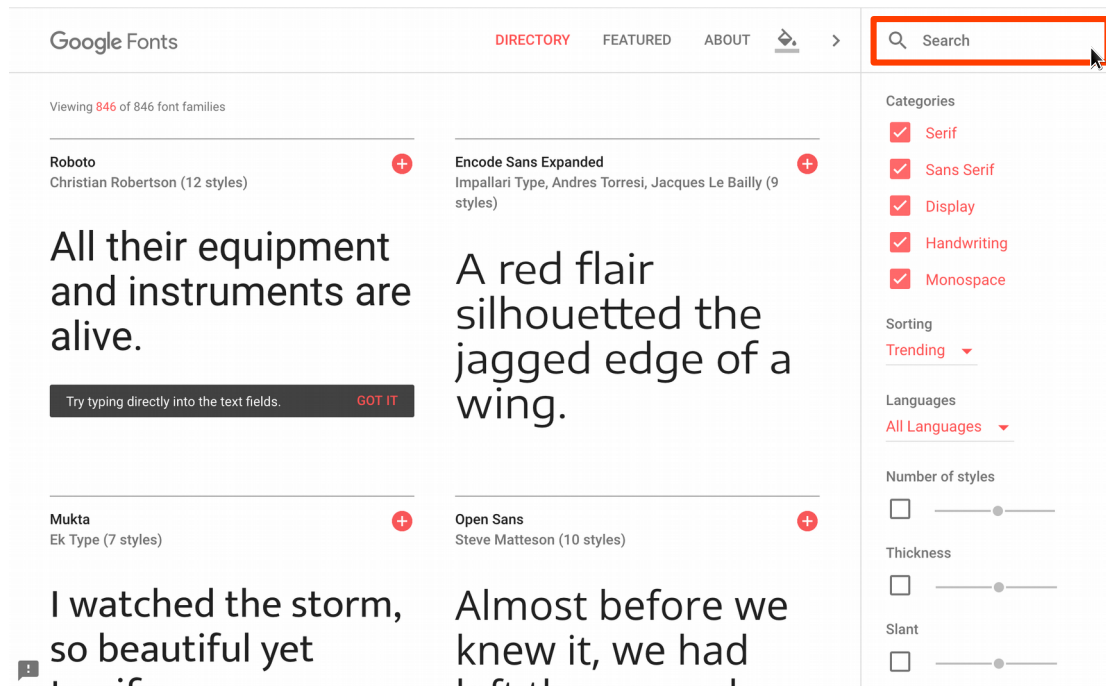
Please note that you get credit for this lab in Part 1. You do not get credit just for submitting the file to CMS!

Please note you are **not allowed to use PHP for Project 2**. Project 2 is about understanding and designing for a specific audience, not PHP. If you use PHP for Project 2 expect a significant deduction in points.

(Optional) Part 5: Web Fonts

Interested in adding specific fonts to your web page that aren't generic font families? In this section, you will learn how to include web fonts into your website.

For this section, let's say you want to make the default font on this page a fun cursive font. Navigate to <https://fonts.google.com/> and type "Lobster Two" into the search bar:



Click the “+” button in the top corner to add this to your list of fonts:

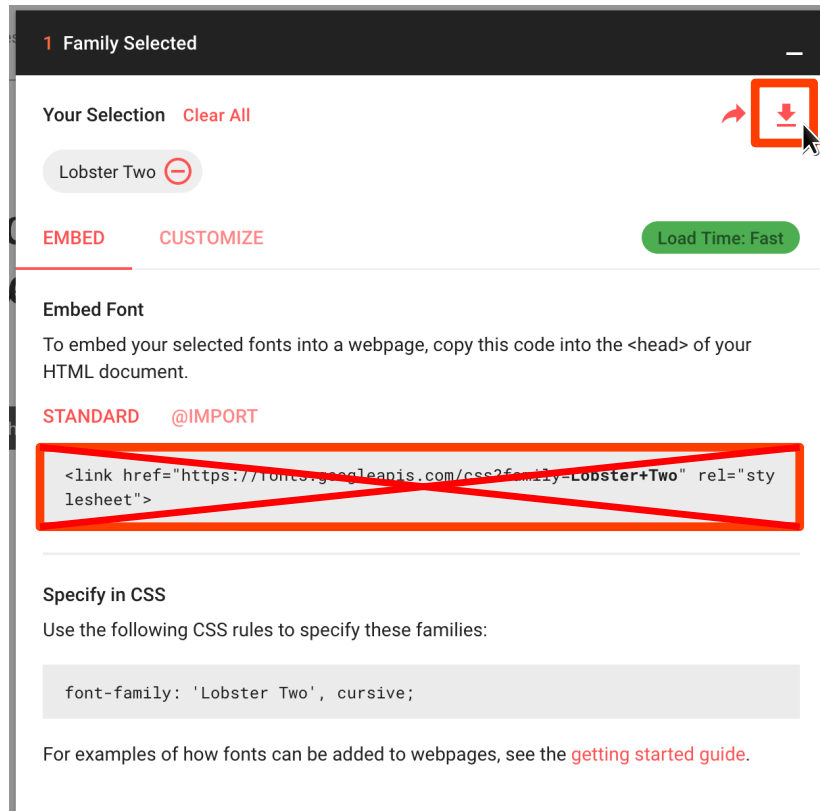


You should now see a small bar on the bottom of the page that says “1 Family Selected.” Click this bar.



Now click the download button in the top right. See the figure below.

Do not follow the directions on embedding the font from Google. We've crossed it out in the figure below to remind you! Including this code is considered embedded code. You are not allowed to embed code for Project 2. The goal of this class is to understand *how the web works*, not how to copy things from Google.



Unzip Lobster_Two.zip. Create a new folder in your lab7-activity folder called “**fonts**”, then move the **Lobster_Two** folder inside the Fonts folder.

Now, add the following to styles/all.css:

```
@font-face {
  font-family: "Lobster Two";
  src: url("../fonts/Lobster_Two/LobsterTwo-Regular.ttf");
}
```

This defines a new font family, “Lobster Two”, based on the **.ttf** font file you just placed in the **fonts** folder.

Then add the following property to the body:

```
font-family: 'Lobster Two', cursive!important;
```

This sets the font family to Lobster Two, gives a generic cursive font as a fallback, and says that this property is *important*; that is, it should overwrite any future attempts to set the font family of body. This is necessary because code loaded by phpinfo also sets the font-family of the body.