Node.js Setup Win

Node/NPM/Validator Setup - Windows

Command Line Reference

Key Controls/Shortcuts

Enter/Return	Executes the current command
Up/Down Arrow	Auto-fills previously/next run commands
Control+C	Stops the process currently running
Tab	Autocomplete a file or directory name

File System Commands

ls	Lists all of the files/folders inside of the current directory
cd	Changes directory to the folder you specify (not including the <>)
cd ..	Goes up one to the parent of the current directory
pwd	Displays the full path to the current directory

Return to Top (https://courses.cs.washington.edu/courses/cse154/20au/resources/assets/node/windows/#nodenpmvalidator-setup---windows)

1) Installing Node.js and NPM

Preface/Explanation

Programming languages are just like real languages. They are used to communicate ideas in a structured
way. In this course, we will be using the JavaScript programming language to both communicate to your
browser how to handle the behavior of your website and to communicate
what we want our server to do and respond to requests with.

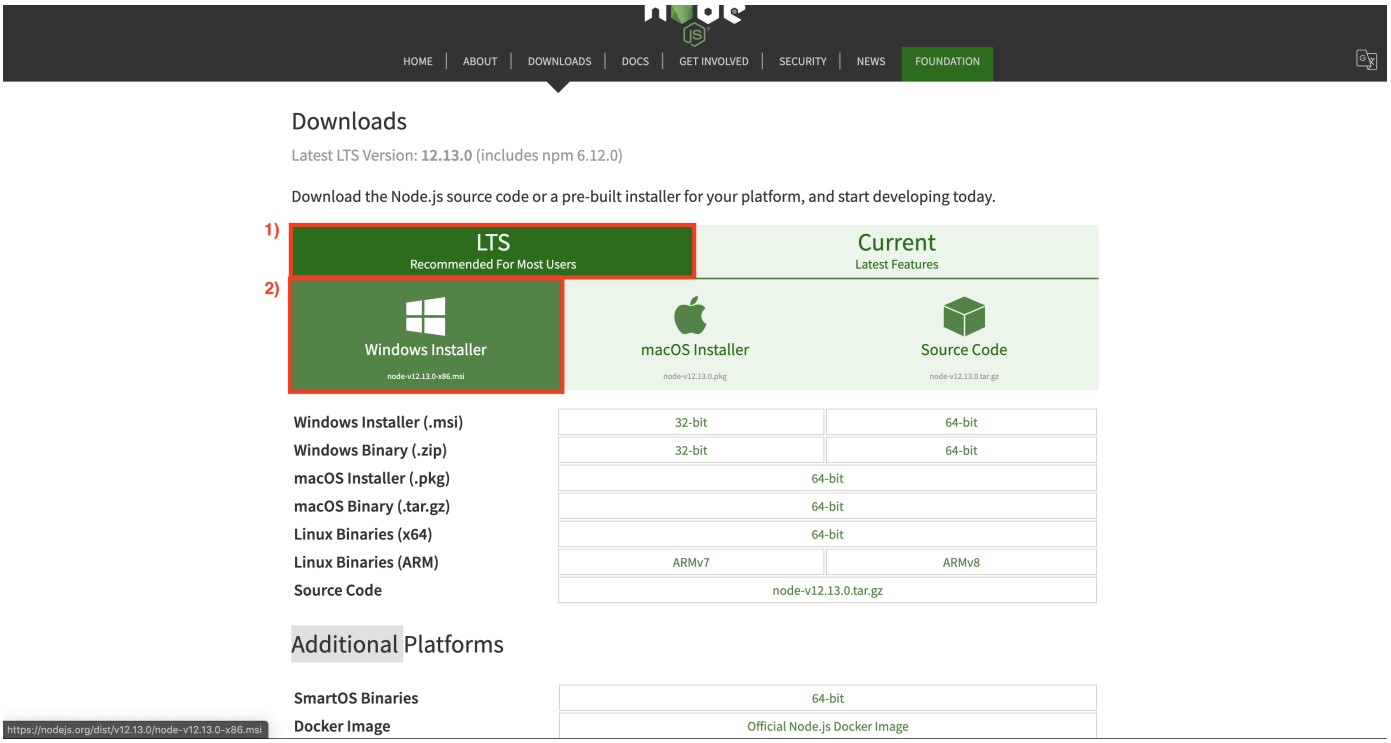
On the client-side, your JavaScript is interpreted by your browser, so it knows how to respond to user events like button clicks and pages loads. On the server-side, your JavaScript is interpreted by a piece of software called Node.js, so that it knows how to respond to user requests over the network.
This includes responding with the client-side files (HTML/CSS/Client-side JavaScript) for the client to

display,
or responding with information stored or calculated on the server. While developing for this course, we will be running both the client-side browser and the server on the same machine (your computer). (We will go more in depth into the server-side later on in the course).

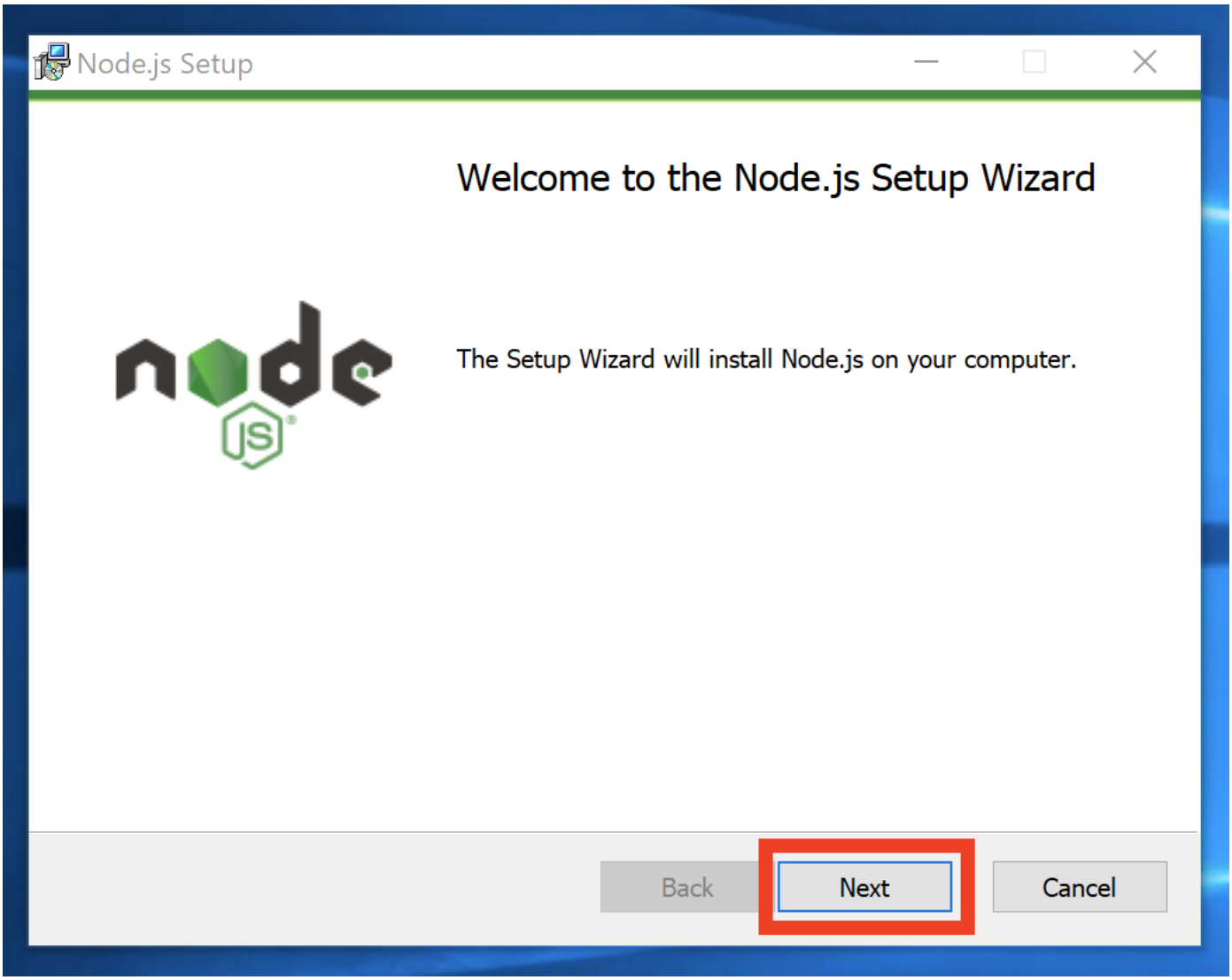
There are a wide array of community-made modules that expand upon the default features of Node.js.
We will be using some of these modules in this class. In order to install and manage these modules, we will use a program called npm that is included when installing Node.js. Follow the steps below to install both Node.js and npm.

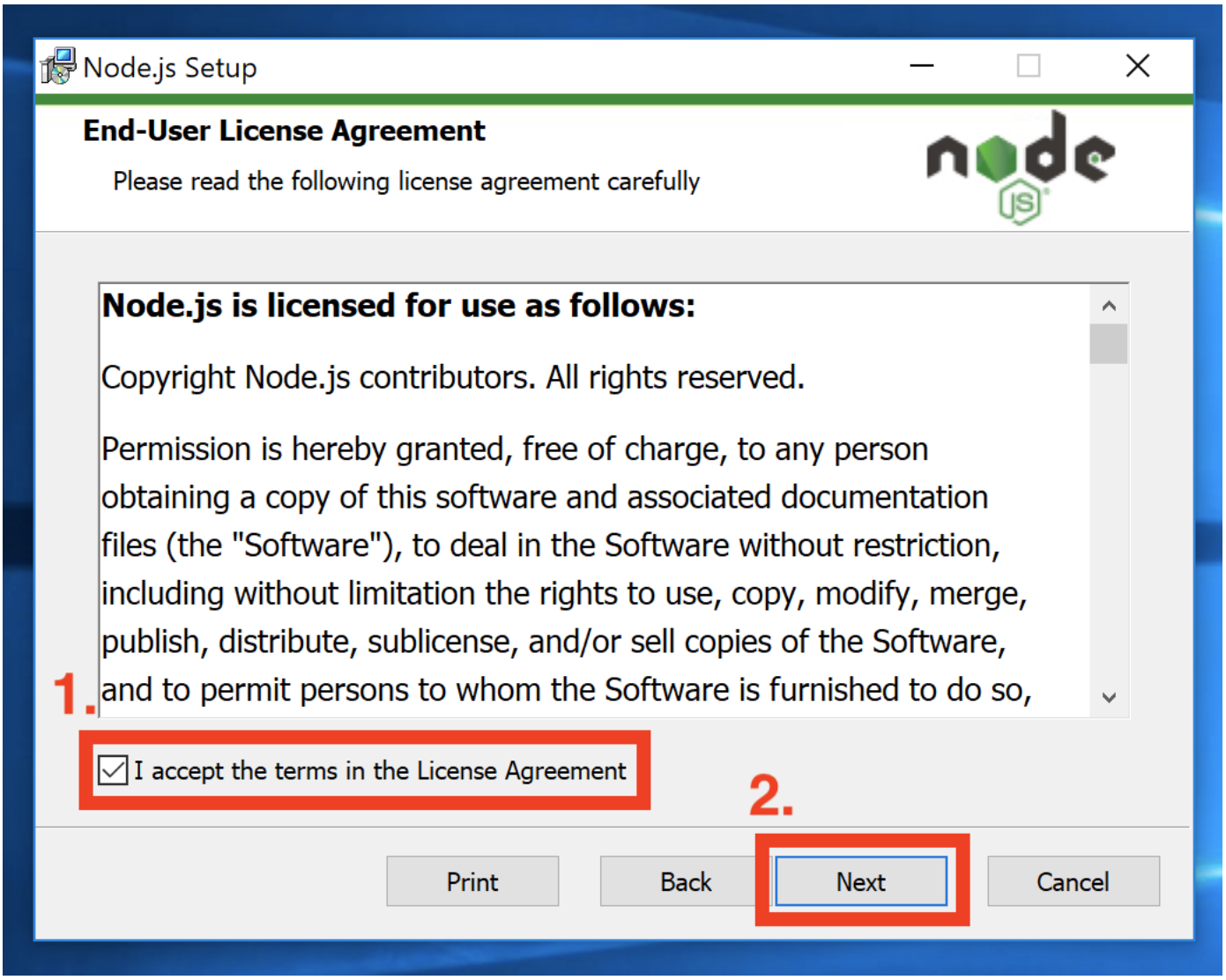
Steps

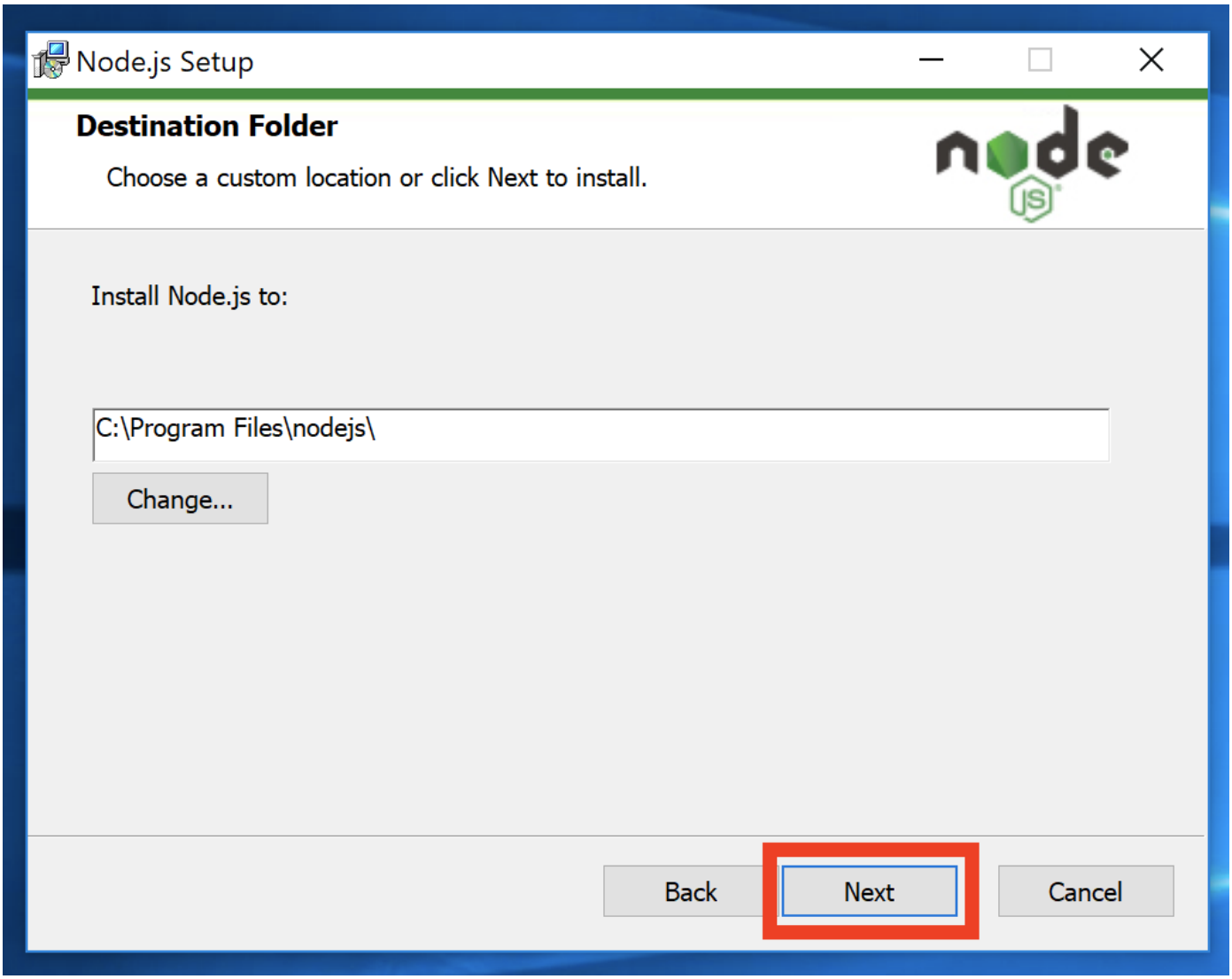
1. Start off by downloading the the LTS (version number >=14.15.0) Node.js/NPM `.msi` installer from **https://nodejs.org/en/download/ (https://nodejs.org/en/download/)** . Then, run the installer to start the installation process.

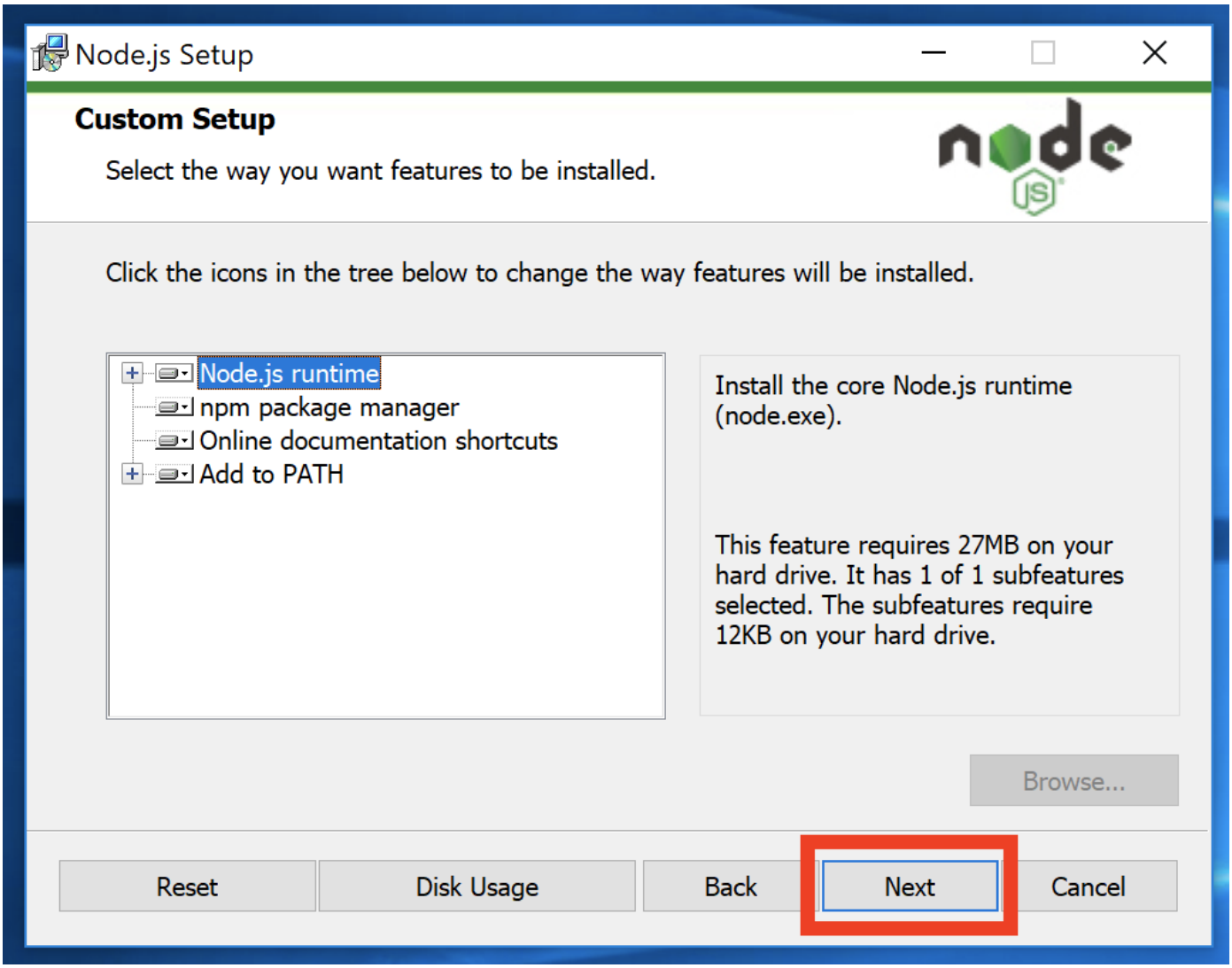


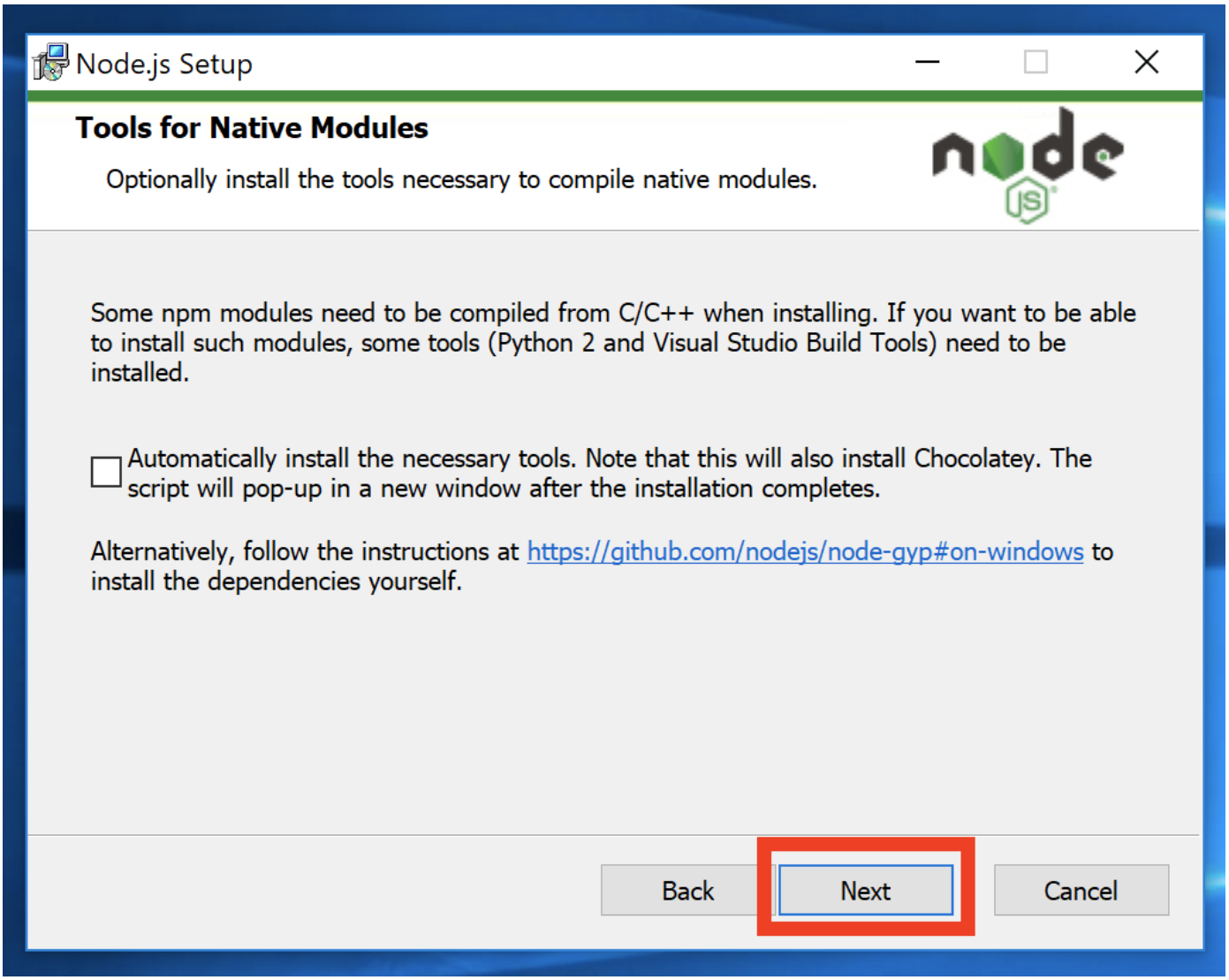
2. Go through the prompts hitting `next`/`accept`/`finish` each time with the default options.

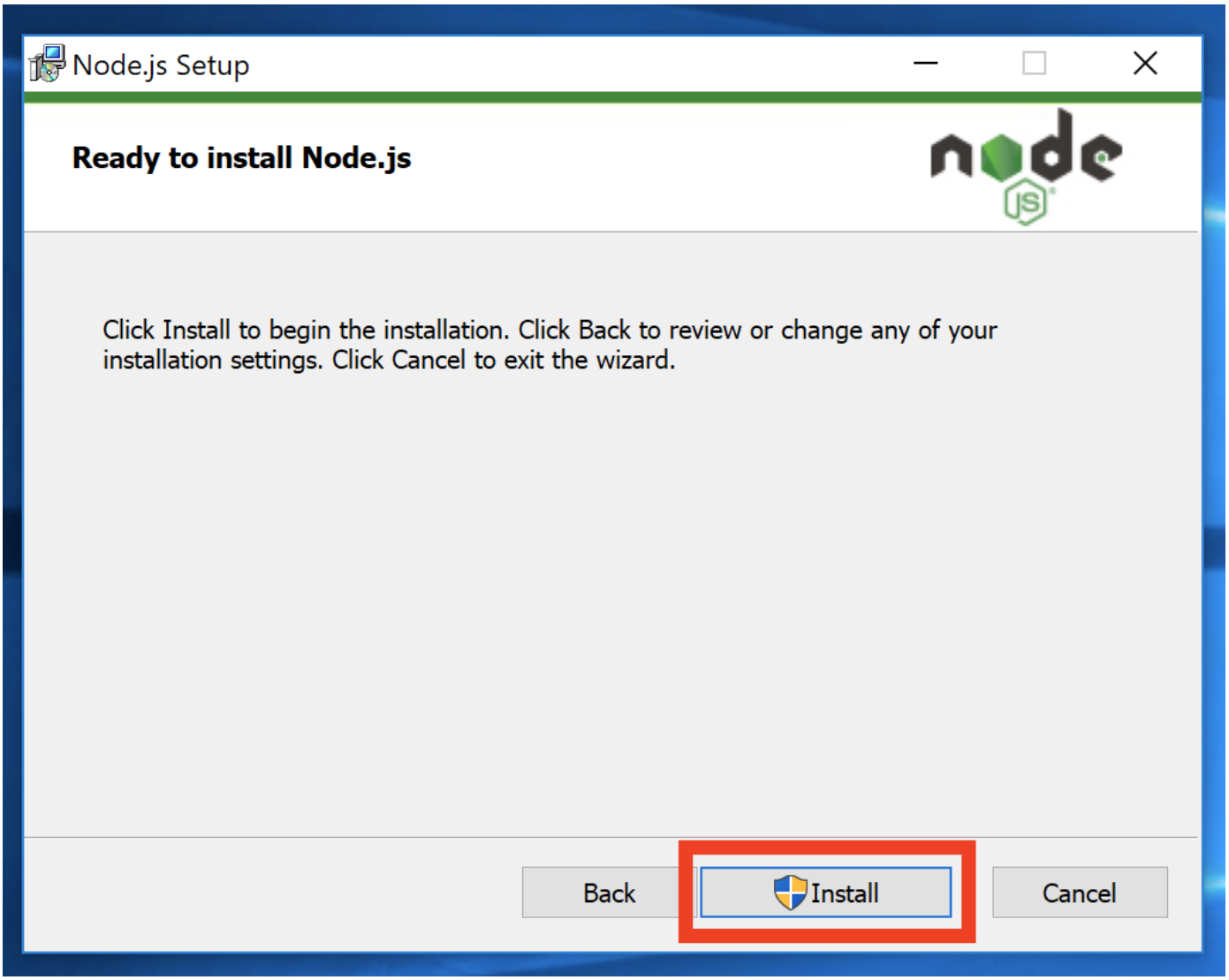

Node.js Setup
Welcome to the Node.js Setup Wizard
The Setup Wizard will install Node.js on your computer.
Back
Next
Cancel

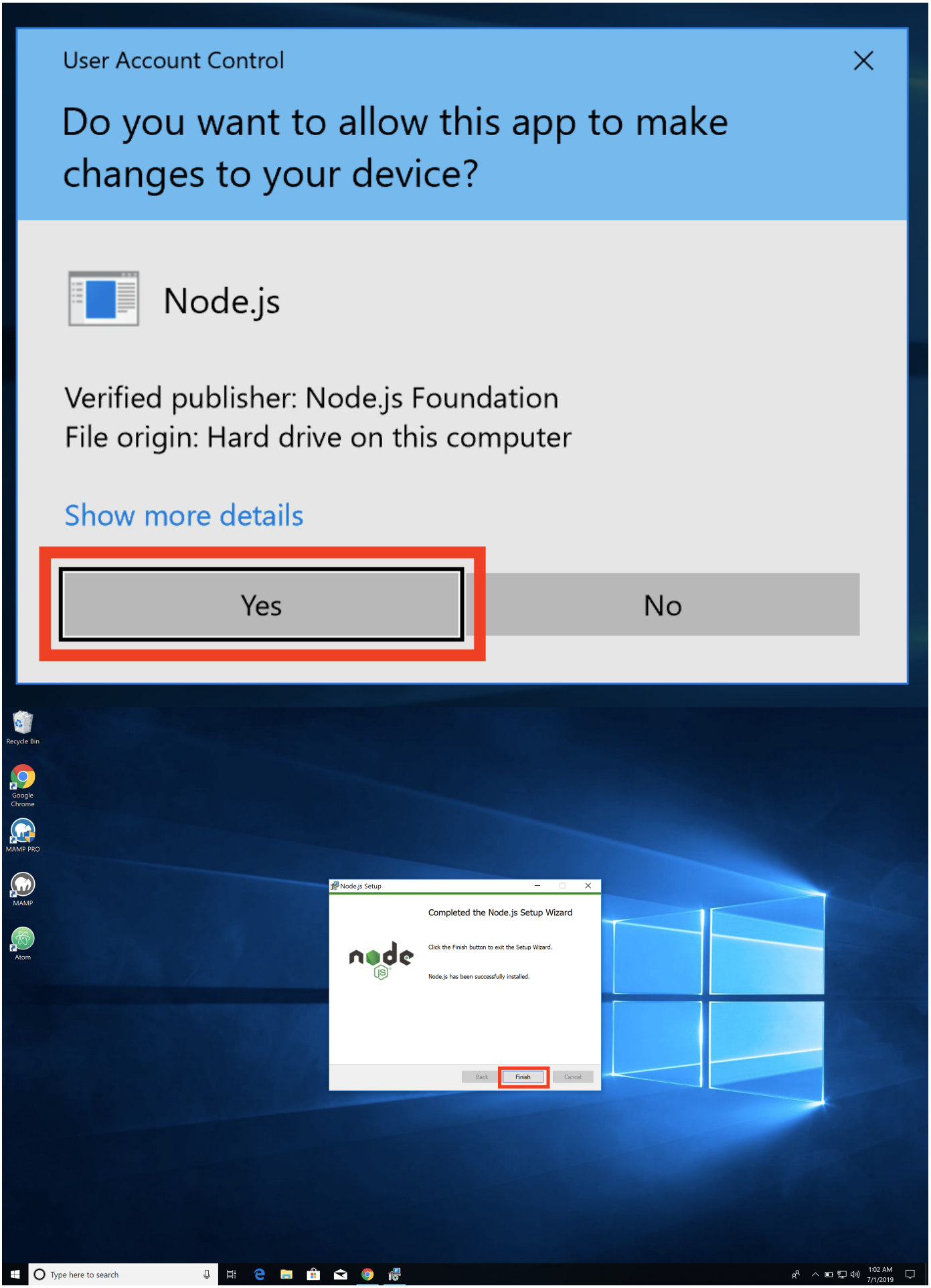

Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
1.
I accept the terms in the License Agreement
2.
Print
Back
Next
Cancel


Node.js Setup
Destination Folder
Choose a custom location or click Next to install.
Install Node.js to:
C:\Program Files\nodejs\
Change...
Back
Next
Cancel


Node.js Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Node.js runtime
npm package manager
Online documentation shortcuts
Add to PATH
Install the core Node.js runtime (node.exe).
This feature requires 27MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 12KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel


Node.js Setup
Tools for Native Modules
Optionally install the tools necessary to compile native modules.
Some npm modules need to be compiled from C/C++ when installing. If you want to be able to install such modules, some tools (Python 2 and Visual Studio Build Tools) need to be installed.
Automatically install the necessary tools. Note that this will also install Chocolatey. The script will pop-up in a new window after the installation completes.
Alternatively, follow the instructions at https://github.com/nodejs/node-gyp#on-windows to install the dependencies yourself.
Back
Next
Cancel


Node.js Setup
Ready to install Node.js
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


User Account Control
Do you want to allow this app to make changes to your device?
Node.js
Verified publisher: Node.js Foundation
File origin: Hard drive on this computer
Show more details
Yes
No
Recycle Bin
Google Chrome
MAMP PRO
MAMP
Atom
Node.js Setup
Completed the Node.js Setup Wizard
Click the Finish button to exit the Setup Wizard.
Node.js has been successfully installed.
Back
Finish
Cancel
Type here to search
1:02 AM
7/1/2019

3. Next, we make sure Node has installed correcctly. Open Git Bash on VSCode or Git Bash regularly. You should have this already set up from the beginning of the quarter. Refer back to the guide **here (https://courses.cs.washington.edu/courses/cse154/20au/resources/assets/vscode-git-tutorial/windows/index.html)** if not.

4. Then, run both `node -v` and `npm -v` to see the version numbers and check that both commands are
installed and working.

2) Installing Global Node Modules

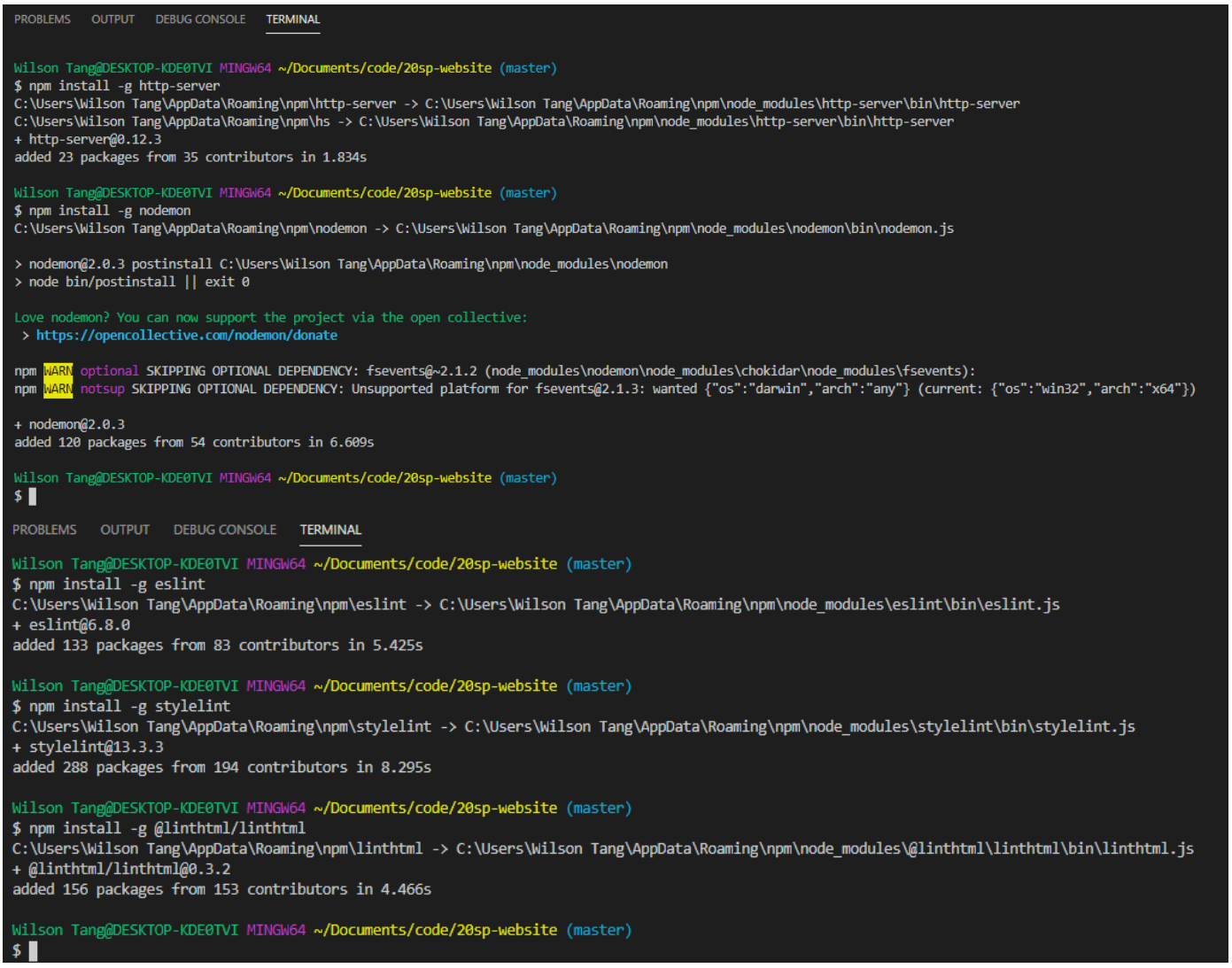
Preface/Explanation

Node modules, by default, are installed locally within a particular project, so they they are only accessible
and importable within a particular Node project directory. However, some modules are able to be installed globally
which can allow them to create useful command-line commands. Here are a few global modules used in this course:

`http-server`	Allows you to quickly run a local server that hosts the files inside of the current directory your run the `http-server` command in.
`nodemon`	A wrapper command that acts like if you run `node`, but it detects any changes you have made to your files, and restarts the `node` process so your changes are immediately reflected without manually restarting.
`eslint`	Javascript linter commandline module that we use to run the local linters.
`stylelint`	CSS linter commandline module that we use to run the local linters.
`@linthtml/linthtml`	HTML linter commandline module that we use to run the local linters.

Steps

1. Run the normal npm installation command for each module, but with a `-g` flag to tell npm that you want the modules installed globally. If any of these fail, try running Git Bash "as an administrator".
   - `npm install -g http-server`
   - `npm install -g nodemon`
   - `npm install -g eslint`
   - `npm install -g stylelint`
   - `npm install -g @linthtml/linthtml`



3) Installing Local Linters in VSCode

Preface/Explanation

Here we will be installing VSCode extensions and linter rules so that VSCode will display whenever your code does not lint and highlight any issues. **Note:** This is for your convienence, and you should always default to the Gitlab/Gradescope output over this local linter output.

Steps

1. Install the VSCode extensions by following each of these links and clicking install. It should open up VSCode and install the linter extensions.
    - LintHTML (HTML): **https://marketplace.visualstudio.com/items?itemName=kamikillerto.vscode-linthtml (https://marketplace.visualstudio.com/items?itemName=kamikillerto.vscode-linthtml)**
    - ESLint (JS): **https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint (https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint)**
    - Stylelint (CSS): **https://marketplace.visualstudio.com/items?itemName=stylelint.vscode-stylelint (https://marketplace.visualstudio.com/items?itemName=stylelint.vscode-stylelint)**