Resources for Assignment 2

Get started

Install Node. There are several ways to do this as you can see in the following link: https://nodejs.org/en/download/package-manager/. (this link has instructions for Linux, OS X, Windows and other operating systems). The easiest way is to download the installer directly from the nodejs.org website.
Better if you can install version, 8.10 or higher (the same version React tutorial requires)
The latest version of Node is 16.04.0

You also need NPM, and you should install it too. The best way to install npm is to install node (previous step) using the node.js installer. NPM is installed as part of Node. If you have downloaded the installer directly from the nodejs.org website, you have NPM installed.
Better if you can install version, 5.6 or higher (the same version React tutorial requires)
The latest version of NPM is 8.4.1

You would also need to install Git to get the boilerplate easily:
https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Then, follow this link and the instructions on the page to clone and run the boilerplate:
https://github.com/Juan-Alvarado/weatherapp-boilerplate

Preact

Preact is a library based on React. React was developed by Facebook to make user interfaces. We're using Preact because it's much smaller and contained - React is vast, and contains far more than what we need. However, through Preact, you'll learn how React works, and control data flow between UI components.

Official site
Preact | Preact: Fast 3kb React alternative with the same ES6 API. Components & Virtual DOM.

Documentation on components
Components

A nice gentle Getting Started
https://preactjs.com/guide/v10/getting-started/#app

The NPM documentation for Preact has a convenient list of existing projects. Some are on codepen.io (a brilliant website with lots of working examples of bits of code - great for seeing how other people build things)
https://www.npmjs.com/package/preact

Weather API

There are many weather APIs that you can use, and they all have different rules (like the number of times you can ask them for data per minute, hour or day) and provide additional data. OpenWeatherMap is used in the boilerplate. This is a really good one - there are tons of data you can get from it, also because you can use it for free (lots of basic weather data and not so basic).

OpenWeatherMaps API documentation:
https://openweathermap.org/api

You'll need to get an API key, an identifier that you send with each request, so they know you are a registered user, who you are, how often you're requesting data, etc. Get an API key for free.

Though OpenWeatherMaps is useful in its free version, you are free to use any other APIs you want to, for weather data or any other data you might want to bring in.

Learning HTML/CSS/JavaScript

You'll use HTML/CSS to style your design and position your components.

Understanding the Document Object Model: Brush up if you're not sure
https://www.w3.org/TR/WD-DOM/introduction.html

W3 schools: Great resource for good HTML/CSS guidance
http://www.w3schools.com/

Mozilla Developer Network: Another great source of HTML/CSS guidelines
https://developer.mozilla.org/en-US/

Understanding LESS

Preact uses a CSS pre-processor called LESS. CSS is a bit of a pain to write, so LESS is a way of writing a lot less code (hence the name!). There are other preprocessors (such as SASS), and everyone tends to have a favourite, but Preact uses LESS, so we encourage you to get to grips with it.

Official page
http://lesscss.org/

A good LESS tutorial
http://www.developerdrive.com/2012/04/learning-less-an-introduction/

Node and NPM

Node is the web server that will run your project, and NPM is the package manager that allows you to use useful stuff that other people made to extend Node.

NPM documentation
https://docs.npmjs.com/

Node documentation
https://nodejs.org/en/docs/

Versioning and collaboration tools

It's important to collaborate on code and be able to version. It's also important to communicate and divide up the work, and know what still has to be done. These are some great tools for doing these things:

Slack, a great way to communicate and share files
http://www.slack.com

Trello, project management and to-do list software that's great for prioritising and assigning tasks, and understanding where your team is in terms of implementation
http://www.trello.com

Github, a way to manage your team's code (you can sign up for a student account and get lots of private repos, usually a paid feature):
http://www.github.com

Graphical resources

You shouldn’t use graphical assets that you don’t have permission to use. However, there are lots and lots of graphical assets around that are royalty-free. Here are a few:

Stock photos: Unsplash, beautiful hi-res images you can use for free
http://www.unsplash.com

Icons: Font Awesome, an icon font set that is great and widely used
http://fontawesome.io

Fonts: FontSquirrel allows you to browse available fonts and make CSS font kits:
http://fontsquirrel.com

FlatIcons: Download icons for free
http://www.flaticons.net

Google Material: A very very good pattern library. Many corporations have pattern libraries, which are documented rules of layout that are applied uniformly across a brand. Google Material isn’t the absolute answer to all design questions, but it’s worth looking at to get a sense of how Google standardises their look and feel across a massive range of products, how they use fonts, and how they use colour.
https://material.io/