**CS 628 Modern Full-Stack Development**

**HOS01: ……..**

February 3, 2023 - Developed by Shingo Kise

School of Technology & Computing (STC)

City University of Seattle (CityU)

Icon

Description automatically generated

Icon

Description automatically generatedIcon

Description automatically generated

**Before You Start**

* This tutorial targets Windows users and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgment while going through this cookbook-style tutorial to complete each step.
* For your working directory, use your course number. This tutorial may use a different course number as an example.
* The directory path shown in the screenshots may be different from yours.
* If you are not sure what to do or confused with any steps:
  + Consult the resources listed below.
  + If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Create a single-page react application with no configuration
* Understand the purpose of package.json file
* Understand the purpose of node\_modules file
* JavaScript?
* HTML?
* CSS?

**Resources**

Required

* create-react-app - Getting Started - <https://create-react-app.dev/docs/getting-started/>
* React - React Tutorial - <https://beta.reactjs.org/learn>
* W3School - React Tutorial - <https://www.w3schools.com/react/default.asp>
  + React Introduction - <https://www.w3schools.com/react/react_intro.asp>

Recommended

* W3School - HTML Tutorial - <https://www.w3schools.com/html/default.asp>
* W3School - CSS Tutorial - <https://www.w3schools.com/css/default.asp>
* W3School - JavaScript Tutorial - <https://www.w3schools.com/js/default.asp>
* W3School - Node Tutorial - [W3Schools - Node Tutorial](W3Schools%20-%20Node%20Tutorial)

**Introduction**

In this course, as we will learn MERN “full-stack” development, we’ll be learning about coding clients (so-called frontend) as well as the server code (so-called backend) to form a cohesive application. At first, we will focus on the development of the frontend side, by learning about fundamentals of React.

**React**

There are many framework/library/toolkit for building web-based client applications.

But, over the last couple of years, a few popular options have floated to the top of the pile and React is one of them. (We will also learn about Node.js and Express in this course as well, as they are part of M**E**R**N** Stack!)

Chart

Description automatically generated

2022 stackoverflow Developer Survey

**A Brief History of React**

React (sometimes referred to as React.js or ReactJS) is a product of Facebook. React is a library for building web-based user interfaces.

It all started back in around 2010 when Facebook developers began to run into a lot of issues with code maintenance, and the engineers at Facebook developed React to solve the problem.

Today, many websites in a variety of industries are built with React. The most famous companies that use the framework include Meta, Netflix, Uber, Airbnb, and The New York Times.

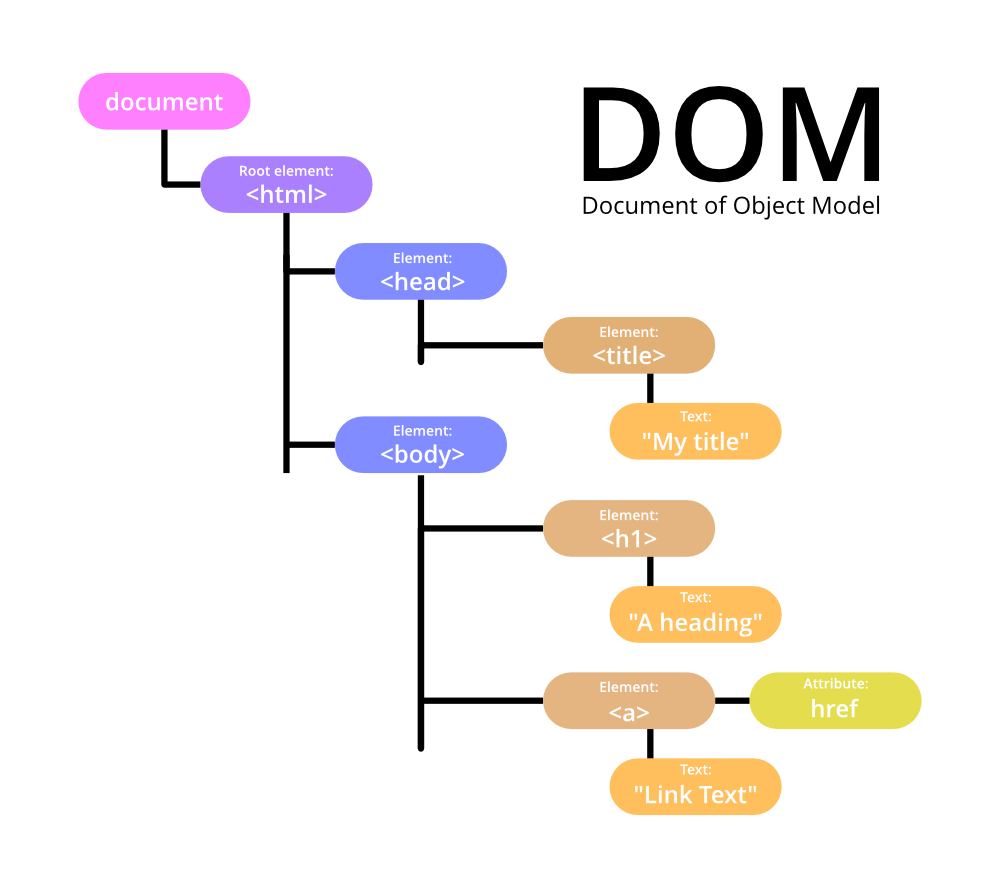
**What Is React?!**

At a very high level, the point of React is to make it easy to reason about the structure of your interface at any given moment in time. This is accomplished by way of *components*, which you can think of as self-contained pieces of the interface. With many of those components, a user interface can be built.

Key elements in React are components, props, state, style, and virtual-DOM.

**What is virtual-DOM?**

DOM is the Document Object Model, which has the tree-like structure that the browser builds as it parses the HTML file. All the elements, denoted by tags, in the HTML, become nodes in this tree.

This DOM has a direct tie to what you see on the screen, and it offers an API to manipulate it with.

Typically, when you do something that makes a change to the page, whether it be as a result of user action or programmatically, the browser has to perform some relatively intensive and expensive work, primarily to render the entire screen to “repaint” the screen. All of this takes computing time and sometimes it impacts the user experience.

React uses the concept of a virtual DOM. This is, in essence, a secondary DOM that sits conceptually on top of the real DOM in memory. Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

React finds out what changes have been made, and changes only what needs to be changed.

Now that we know a little bit about React, let’s create a single-page react application!

> Open GitHub and go to [CS628](https://github.com/samchung0117/cs628-instructor) (wording might change. hyperlink could be helpful?)

> Click Code and Create codespace on main. (Do I need to explain codespace?)

Graphical user interface, application

Description automatically generated

> It will open a new tab with a develop environment with HOS folders

Graphical user interface, text

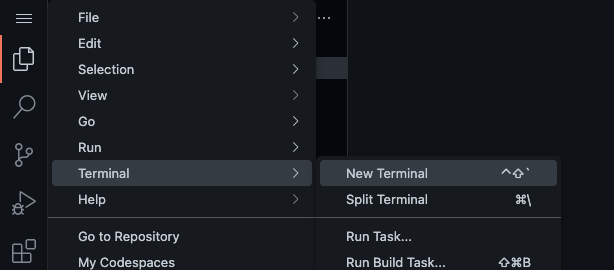
Description automatically generatedIf you take a look at the HOS1 folder, we can see that there is a package.json file, and public and src folders, which contain several JavaScript files, an HTML file, and CSS files.

These are the default setup, which was setup with [Create React App](https://create-react-app.dev/docs/getting-started), which is an officially supported way to create single-page React applications. It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production.

We will explore some files to get a better understanding later.

> Open Terminal and change your current directly to HOS1/my-app

\* if the Terminal did not open automatically, click ≡ and New Terminal



Graphical user interface, text, application

Description automatically generated

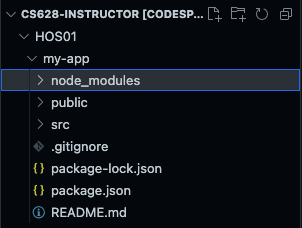
**$ cd HOS01/**

**$ cd my-app/**

> then type **$ npm install**



> the npm install command will create **node\_modules** file in my-app folder

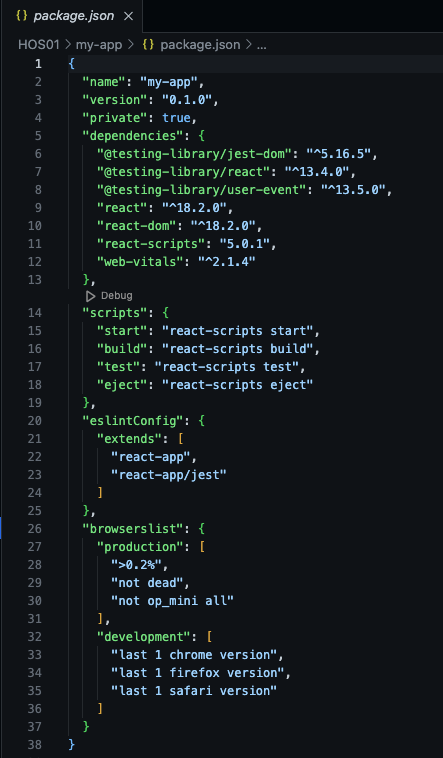


**What is npm ?**

NPM, which stands for Node Package Manager, is a companion app that installs alongside Node.

Much of Node’s utility comes from its large package library, which is accessible from the npm command. The NPM registry hosts more than 1.3 million packages of free, reusable Node.js code.

**Now, Let’s take a look at the package.json file.**



In the package.json file, you can see the default information about this project.

At the line 5, we can see that a list of packages, or so-called dependencies (a library that a project needs to function effectively).

At the line 14, we can see a list of scripts, which allows you to specify a dictionary of commands that can be run at various points in the lifecycle of your package for various purposes.

At Line 22 and 23 show that we used [Create React App](https://create-react-app.dev/docs/getting-started) to setup this environment.

As you can see, this file shows important metadata about your project.

Package.json file is powerful, especially when sharing projects with others. As the file holds important metadata about a project and dependencies, others can set up their environment the same as yours.

**What is node\_modules file?**



Text

Description automatically generated

A picture containing icon

Description automatically generated