**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>

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.

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.

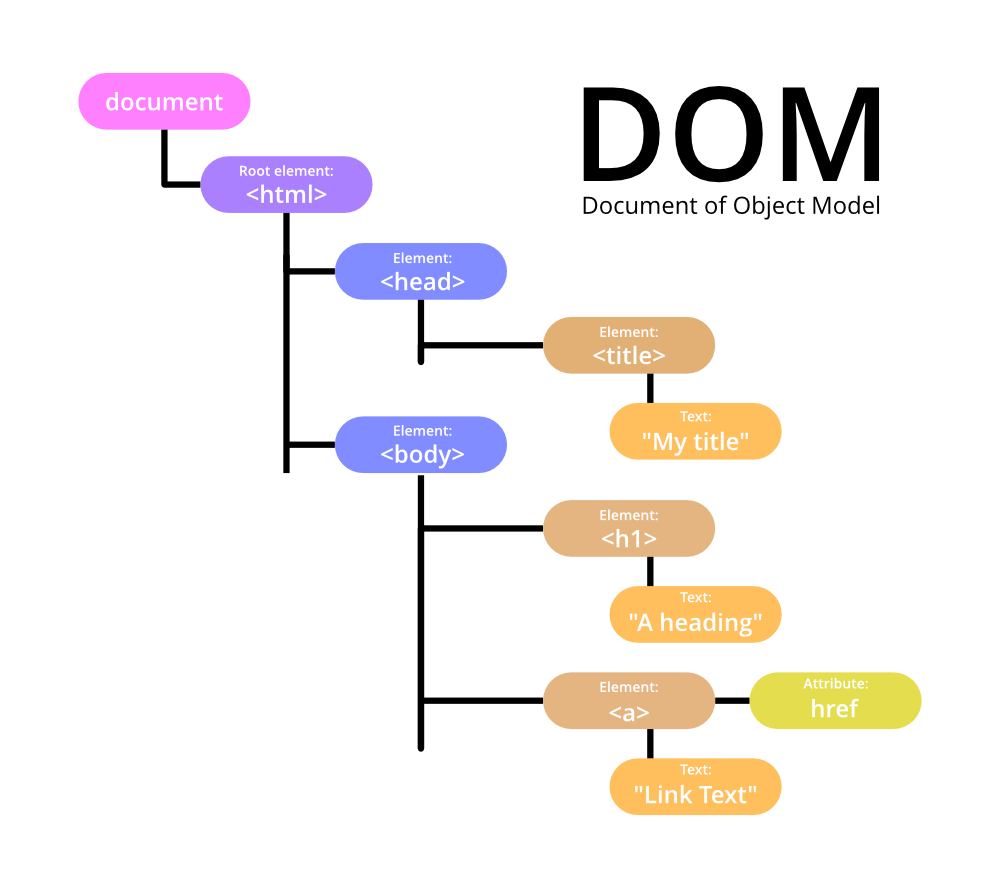
**What Is React?!**

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?**

We will not directly use virtual-DOM but React will be using it extensively. 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. Rather than manipulate the real DOM directly, React updates the virtual DOM and then will intelligently figure out the changes and the least amount of real DOM work that can be done to accomplish the update. Most importantly, this allows React to batch up real DOM changes and apply them all in one go, which is much more efficient than doing each one individually. The result is better performance and user experience (no refreshing the page often) than can typically be achieved with direct DOM manipulation.