**Asmeret Mehari, Assignment Week 10: Notes for Week 9 and Week 10**

This note mainly focuses on React.

**What is React ?**

**React** is a JavaScript library and used for building user interfaces(UIs) - usually used with npm to manage dependencies and scripts. React applications are built from parts of UI call components.

React allows developers to combine HTML, CSS, and JavaScript to create components. These components are reusable, so can be imported to other components. This speeds up the web development process. .

***React uses Virtual DOM(Document Object Model)***. It works first directly with the virtual DOM (smaller size of the actual real dom) and then updates the actual DOM with necessary changes. The virtual DoM prevents the real DOM from direct manipulation. The Virtual DOM works in a more efficient way than updating the real DOM directly. Thus, it optimizes the process of updating the user interface by using lightweight representation of the actual DOM.

***React uses JSX***. JSX (JavaScript XML is a syntax extension for JavaScript and allows developers to write HTML-like code directly in JavaScript code. JavaScript expressions are embedded using curly braces `{ }` It is useful to define components. It allows react components to be more readable and natural.

JSX uses single root elements: Also called parent tags. For example, <div>, <Fragment>

JSX uses camelCase property, unlike HTML. For example, onclick in HTML => onClick in JSX.

Basic example of JSX:

` import React from 'react';

function ExampleComponent() {

const message = "JSX is great!";

return (

<div className="example-class" style={{ color: 'blue', fontSize: '20px' }}>

<h1>{message}</h1>

<p>JSX allows us to write HTML-like syntax in JavaScript.</p>

</div>

);

}

export default ExampleComponent;

In the above example copied from our class week 9 notes, className is used instead of class. {message} is used to display content. Style is written as a JavaScript object.

**What is a react component?**

**A react component is a JavaScript function and can be** built from small buttons, text, images or an entire page. Components are UI building blocks. React allows us to combine these units into reusable, netable components. Components can be used even in different projects. Every react component has its own virtual Dom, logic and UI.. Components can be composed, they can be combined to create bigger projects.

There are two types of components in React. Functional Components and Class components, both are used as primarily approaches to define components. **Functional components** are easy to use and with the advent of hooks they can even manage state and have lifecycle methods.

Components have benefits because they make UI easier to understand and maintain, they are reusable which provides consistency, the way components work also allow high level of collaboration among developers. They can also be tested separately to behave correctly. Thus easy to spot and fix problems.

React Components must start with a capital letter, that differentiate them from HTML tags

Example <Hero/> is React Component while <div> is the basic HTML tag.

**Example of Defining Components, adapted from Week9 notes**

**import React from 'react';**

**function Greeting(props) {**

**return <h1>Hello, {props.name}!</h1>;**

**}**

**export default Greeting;**

**```**

**In this example:**

**- `Greeting` is a functional component.**

**- It accepts `props` (short for properties), which is an object containing all the passed attributes.**

**- We use `{props.name}` to dynamically insert the value of `name` into our `<h1>` tag.**

**- Finally, we export `Greeting` so it can be used in other parts of the application.**

**What is Babel?**

Bable is a JavaScript compiler. It ensures compatibility across different environment.

**JSX**  cannot transpile data. So Babel changes JSX syntax into regular JavaScript function calls, which is called React.createElement() calls.

Bable makes changes that appear in ES6+ more readable and accessible in older browsers by transpiling modern JavaScript into ES5. Features of ES6 include const, let import/export,

**What are React Developer Tools?**

React developer tools are used to inspect React components, edit props, and state and identify any performance problems. Example, Google Chrome developer tools.To install React Developer Tool, type npm install -g react-devtools . To open it from the Terminal type react-devtools .

**How to Start a New React project?**

React can be created without a framework, but it is recommended that developers use from the React-powered frameworks. This prevents building the framework later. Frameworks are useful to handle problems such as code-splitting, routing, data fetching, and generating HTML.

To start a new react project: type in the VScode Terminal:

Npx create-react-app + (name of project- my-app)

Cd my-app

Npm start

This is one way to start to react. There are other ways to install the program. During installation if problem appears like this:

npx create-react-app my-app

npm ERR! code ENOENT

npm ERR! syscall lstat

npm ERR! path C:\Users\asmer\AppData\Roaming\npm

npm ERR! errno -4058

npm ERR! enoent ENOENT: no such file or directory, lstat 'C:\Users\asmer\AppData\Roaming\npm'

npm ERR! enoent This is related to npm not being able to find a file.

npm ERR! enoent

npm ERR! A complete log of this run can be found in:

Then:

npm uninstall -g create-react-app

npm uninstall create-react-app

npx create-react-app project-app

**What is npm?**

**Npm** is an abbreviation for **Node Package Manager**. **Node Package Manager** is the default package manager for Node.js and JavaScript. It handles packages/modules for Node.js applications, thus web developers can easily install, update, and manage dependencies and scripts that are needed to run a project, including React projects.

**What does it mean npm run dev?**

Npm manages dependencies and scripts.

`run` command This is the npm command and executes a script defined “scripts” in the package.json file used for the project, in this case the react project.

Scripts in ‘package.json’ looks like this

{ "name": "my-vite-react-app",

"version": "1.0.0",

"scripts": { "dev": "vite" }, dev:cite defines the dev script. When type npm run dev – npm executes `vite`. Vite starts the development server for the Vite based React application. The vite’s development server officers fast HMR (Hot Module Replacement) to update the application in the browser without a full page reload. It also optimizes the build process.

"dependencies": {

"react": "^17.0.2",

"react-dom": "^17.0.2",

"vite": "^2.7.1" } }

[Dependencies are external packages or libraries that allow an application to function properly - in the node.js and npm environment, dependencies are managed via package.json file. There are two types of dependencies: Production dependencies and Development Dependencies. In production dependencies are required for an application to run in a given environment, for example web frameworks such as express. In Development dependencies they are used during development and testing. Dependencies are managed by package managers such as npm or yarn.

**‘Npm run dev**’ executes a development server and compiles a react application, monitors its changes and gives fast feedback during development. Withs capabilities of automatic compilation, bundling, and HMR it provides a smooth development experience. HMR speeds up development cycles by allowing modules to be updated in a running application without requiring a full page refresh.