MODULE-2 TW-01 TEAM LEAD VERSION

TEAMWORK

CLARUSWAY
WAY TO REINVENT YOURSELF

Meeting Agenda

▶ Icebreaking

▶ Workshop Activities - Tuesday

▶ Teamwork Activities - Friday

  ▶ Questions

  ▶ Interview Questions

▶ Video of the week

▶ Retro meeting

▶ Case study / project

Teamwork Schedule

Ice-breaking   10m

- Personal Questions (Stay at home & Corona, Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Workshop Activities (Tuesday)   90m

ReactJS

What is React?

React is a JavaScript library for building user interfaces. It allows developers to create reusable UI components and manage the state of an application efficiently.

Key Features:

Declarative: Describe how the UI should look, and React takes care of updating and rendering. Component-Based: Build encapsulated components that manage their state. Virtual DOM: Enhances performance by updating only the necessary parts of the DOM.

npm (Node Package Manager)

- What is npm?

npm is the default package manager for Node.js, the JavaScript runtime. It is used to install, share, and manage packages, which are libraries or modules of reusable code.

- Basic Commands:

npm install: Installs dependencies listed in the package.json file. npm install : Installs a specific package locally. npm install -g : Installs a package globally (useful for command-line tools).

- package.json File:

This file includes metadata about the project and a list of dependencies. Running npm init helps create a package.json file interactively.

- Managing Dependencies:

Dependencies are listed in the dependencies section of package.json. npm install --save : Adds a package to the dependencies. npm install --save-dev : Adds a package to the devDependencies (used during development but not in production).

Yarn

- What is Yarn?

Yarn is a package manager alternative to npm. It aims to be faster, more reliable, and more secure than npm.

- Basic Commands:

yarn install: Installs dependencies listed in the package.json file. yarn add : Adds a package to dependencies. yarn add --dev: Adds a package to devDependencies. yarn global add : Installs a package globally.

- yarn.lock File:

Similar to package-lock.json in npm, yarn.lock locks down the version of each installed package. Ensures consistent installations across different machines.

- Advantages of Yarn:

Performance: Yarn is known for faster and more parallelized package installations. Offline Mode: Yarn can install packages without an internet connection if they are already cached. Deterministic Installs: The yarn.lock file ensures consistent installations across different environments.

- Migrating from npm to Yarn:

If a project already uses npm, you can switch to Yarn by running yarn in the project directory. It will generate a yarn.lock file and install dependencies. Summary: Both npm and Yarn are powerful tools for managing dependencies in a JavaScript project. The choice between them often depends on personal preference and project requirements. Developers should be comfortable using either, as both are widely adopted in the JavaScript ecosystem.

Starting a React Project

- Create React App:

  The easiest way to start a new React project is by using Create React App. It sets up a new React project with a default configuration, allowing you to focus on writing code.

```
npx create-react-app my-react-app
cd my-react-app
npm start or yarn start
```

JSX (JavaScript XML)

- JSX is a syntax extension for JavaScript, recommended for use with React. It looks similar to XML or HTML, but it ultimately gets compiled to JavaScript.

```
import React from 'react';

const MyComponent = () => {
  return (
    <div>
      <h1>Hello, React!</h1>
      <p>This is a JSX component.</p>
    </div>
  );
};

export default MyComponent;
```

- In JSX, you can embed JavaScript expressions using curly braces {}.

- Key Points:

JSX provides a concise and readable syntax for defining React elements. It allows the use of HTML-like tags to represent components.

- Self-Closing Tags:

JSX supports self-closing tags for elements that don't have children.

```
const image = <img src="path/to/image.jpg" alt="Description" />;
```

- JSX Fragments
  Fragments allow you to group multiple JSX elements without introducing an extra parent div.

```
const MyComponent = () => {
  return (
    <>
      <h1>Title</h1>
      <p>Paragraph</p>
    </>
  );
};
```

> Understanding JSX is essential for developing React applications, as it is the primary way to define the structure of React components. It combines the power of JavaScript with a familiar HTML-like syntax, making it a powerful tool for building dynamic and interactive user interfaces.

Team Work Activities (Friday)   90m

Ask Questions   20m

1. What is the virtual DOM used for?

A. Improving performance by minimizing direct DOM manipulation

B. Rendering server-side code

C. Manipulating the browser's DOM directly

D. Handling server requests

Answer: A

2. What is JSX in React?

A. A CSS preprocessor

B. A JavaScript extension for XML

C. A form of JSON data

D. A database query language

Answer: B

3. In React, what is the purpose of props?

A. To manage component state

B. To handle routing in the application

C. To pass data from parent to child components

D. To define the component's internal methods

Answer: C

4. What is the fundamental building block of a React application?

A. Functions

B. Variables

C. DOM elements

D. Components

Answer: D

5. What is wrong with this code?

```
const MyComponent = ({ names }) => (
  <h1>Hello</h1>
  <p>Hello again</p>
);
```

A. React does not allow components to return more than one element.

B. React components cannot be defined using functions.

C. The component needs to use the return keyword.

D. String literals must be surrounded by quotes.

Answer: A

6. In MVC, what does React.js act as?

A. Model

B. View

C. Controller

D. all of them

Answer: B

7. What is ReactJS mainly used for building?

A. Database

B. Application

C. User Interface

D. Design Platform

Answer: C

8. What is the name of the tool used to take JSX and turn it into createElement calls?

A. Babel

B. JSX Editor

C. ReactDOM

D. Browser Buddy

Answer: A

9. Which props from the props object is available to the component with the following syntax?

```
<Message {...props} />
```

A. any that have not changed

B. child props

C. any that have changed

D. all of them

Answer: D

10. React components are composed to create a user interface. How are components composed?

A. by putting them in the same file

B. with webpack

C. by nesting components

D. with code splitting

Answer: C

11. Which answer best describes a function component?

A. A function component accepts a single props object and returns a React element.

B. A function component is the same as a class component.

C. A function component is the only way to create a component.

D. A function component is required to create a React component.

Answer: A

Interview Questions   15m

1. How often does the React useState update? Why?

Answer: Since developers use `useState` to enhance performance by creating queues, React doesn't update changes immediately. Candidates should know that useState doesn't implement changes to the state object directly; instead, the updates occur asynchronously.

2. Why is there a need for using keys in Lists?

Answer :

Keys are very important in lists for the following reasons:

- A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists.

- It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

3. What are the components in React?

Answer :

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately.

There are two types of components in React:

Functional Components : These types of components have no state of their own and only contain render methods, and therefore are also called stateless components. They may derive data from other components as props (properties).

```
function Greeting(props) {
  return <h1>Welcome to {props.name}</h1>;
}
```

Class Components : These types of components can hold and manage their own state and have a separate render method to return JSX on the screen. They are also called Stateful components as they can have a state.

```
class Greeting extends React.Component {
  render() {
    return <h1>Welcome to {this.props.name}</h1>;
  }
}
```

4. What is an event in React?

Answer :

An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

React events are named using camelCase, rather than lowercase in HTML. With JSX, you pass a function as the event handler, rather than a string in HTML.

☕

Coffee Break   10m

☕

Retro Meeting on a personal and team level   5m

Ask the questions below:

- What went well?
- What went wrong?
- What is the improvement areas?

Case study/Project   15m

- 01 - Image Gallery

- 02 - Product List *

> *Optinal Project *The solution class will be held on Saturday, December 28th.*

Closing   5m

- Next week's plan
- QA Session