Module No: 3

Subject: React

Week Number: 7

Table of Contents

Key Concepts:

React Introduction :

a) What is React :

- · React is a JavaScript library developed by Facebook for building fast and reusable user interfaces, particularly for single-page applications.
- · It utilizes a virtual DOM to optimize performance by only updating necessary parts of the browser's DOM.
- · Compared to frameworks, libraries like React provide more flexibility and less overhead, allowing developers to integrate them easily into existing projects.
- · Essential tools for using React include Node.js and npm, and Babel is used to ensure compatibility of modern JavaScript features across browsers.
- · Current version of React.JS is V18.3.1 (April2024).
- · Initial Release to the Public (V0.3.0) was in July 2013.
- · React.JS was first used in 2011 for Facebook's Newsfeed feature.
- · Facebook Software Engineer, Jordan Walke, created it.
-

b) Framework vs Library:

Frameworks are collections of libraries that provide structure and pre-built functions for developing software efficiently, such as Angular and Spring. They are challenging to integrate into existing projects and can impact performance due to their size. Libraries, like jQuery and React, are simply collections of reusable functions that enhance performance and can be easily swapped out. While frameworks define how applications are built, libraries offer specific functionalities that developers can utilize as needed.

Highlights

1. Frameworks vs. Libraries:

Frameworks consist of collections of libraries.

Examples:

Frameworks include Angular, Spring, and Node.js; libraries include jQuery and React.

2. Performance:

Frameworks can reduce performance due to extensive code; libraries improve performance.

3. Integration:

Frameworks are difficult to incorporate into existing projects; libraries are easily substitutable.

4. Functionality:

Frameworks provide structure for custom applications; libraries offer reusable functions.

6. Code Reuse:

Libraries enable developers to use pre-written code for efficiency.

7. Scope:

Frameworks have a broader scope, while libraries focus on specific tasks.

c) React Features:

1) React is a Library :

React is often confused with frameworks but is a library that offers flexibility and minimal overhead, making it easier to adopt in various projects. This distinction is crucial for developers looking to choose the right tool for their needs.

2) Performance Benefits:

React's virtual DOM allows it to efficiently update the UI, ensuring better performance compared to traditional DOM manipulation. This efficiency is vital for applications requiring quick, responsive interfaces.

3) Component Reusability:

One of React's strongest features is its ability to create reusable components, which streamlines development and promotes consistency across applications—saving time and reducing errors.

4) Tooling Requirements:

To use React effectively in production, developers need to understand and use tools like npm, Node.js, and Babel. These tools help manage packages and ensure compatibility with modern JavaScript features.

5) Modern JavaScript Compatibility:

Babel plays a key role in allowing developers to use the latest JavaScript features while maintaining support for older browsers. This capability is essential for reaching a wider audience.

Virtual DOM :

The virtual DOM is a lightweight copy of the real DOM that allows React to manage changes more efficiently by minimizing the direct manipulation required on the real DOM. This process significantly enhances the performance of web apps.

A lightweight in-memory representation of the real DOM, implemented as JavaScript objects, enabling efficient UI updates.

How It Works:

1. Initial Rendering:
React renders the entire UI as a Virtual DOM structure.

2. State & Props Changes:
React updates the Virtual DOM when state or props change,

without directly modifying the real DOM.

3. Diff Algorithm:
Compares the updated Virtual DOM with the previous version to

identify changes ("diffs").

4. Reconciliation:
Determines the minimal set of updates required to efficiently update the real DOM.

5. Real DOM Update:
Applies changes to the real DOM selectively, avoiding full re-rendering for performance.

Node Package Manager[NPM] :

To use React in production, you need npm which is included with Node.js.

React application will be created by installing using npm.

Install the Nodejs from the Nodejs official website - https://nodejs.org/en/download/source-code

Checking installation:

Check by typing npm –version in terminal

Check node installation by typing node –version

Creating React App :

Create React App lets you focus on code, not build tools.

To create a project called my-app, run this command:

```
npx create-react-app my-app
```

The create-react-app will set up everything you need to run a React application.

Run the React Application:

Now you are ready to run your first real React application!

Run this command to move to the my-app directory:

```
cd my-app
```

Run this command to run the React application my-app:

```
npm start
```

A new browser window will pop up with your newly created React App! If not, open your browser and type localhost:3000 in the address bar.

Editing Application code:

So far so good, but how do I change the content?

Look in the my-app directory, and you will find a src folder. Inside the src folder there is a file called App.js, open it and it will look like this:

Try changing the HTML content inside App.js and save the file.

Example:

Replace all the content inside the with a <h1> element.

See the changes in the browser when you click Save.

```
function App() {
```

```
 return (
<div className="App">
<h1>Hello World!</h1>
 </div>
);
}
```

export default App;

Example folder Structure:

Package.json –

has total information about the package names installed and their versions.

Package-lock.json-

has indetail about the dependencies information of these packages.

Node_modules-

is a folder where the packages get installed manually into it.

Creating Vite App :

It will provide a faster and learner development experience for modern web pages.

Faster than create-react-app

Creating React Vite app:

```
$ npm create vite@latest
```

Then give your project name, and select Framework as React and Technology as Javascript

It will create a same react like folder structure.

Execute the below commands:

```
cd my-project
```

```
npm install
```

```
npm run dev
```

```
npm run dev- will run the app.
```

Code Snippets:

```
function App() {
 return (
<div className="App">
<h1>Hello World!</h1>
 </div>
);
}
export default App;
```

Conclusion:

By using React framework, we can easily create the frontend UI. And we can create a single page application with only one html file, and modifying that html file content by using Virtual DOM concept, making the interaction faster .

References:

1) MDN WebDocs:

https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started

2) React Official Documentation:

https://react.dev/

3.Brain Teaser

1) Analyze the below code snippet and advise what will be shown on the screen when the App component is rendered with <App name=”Claire” />.

```
import React from "react";
class App extends React.Component {
 render() {
    return <div>Hello, {this.props.name}!</div>;
```

```
  }
}
export default App;
```