Module 6

Q. What is React JS? Explain different features of React JS. Write a code in React Js to display “Hello world".

- **React JS** is a free library for making websites look and feel cool. It’s like a special helper for JavaScript. People from Facebook and other communities work together to keep it awesome and up-to-date.
- **React** is Developed by **Facebook**, React is a powerful JavaScript library used for building user interfaces, particularly for single-page applications.
- It allows developers to create large web applications that can change data, without reloading the page. The main purpose of React is to be fast, scalable, and simple.
- **React** is a **JavaScript library** for building **user interfaces** (UIs) on the web.
- React is a declarative, component based library that allows developers to build reusable UI components and It follows the Virtual DOM (Document Object Model) approach

Features of React JS :

1. JSX(JavaScript Syntax Extension):

JSX is a combination of HTML and JavaScript. You can embed JavaScript objects inside the HTML elements. JSX is not supported by the browsers, as a result, Babel compiler transcompile the code into JavaScript code. JSX makes codes easy and understandable. It is easy to learn if you know HTML and JavaScript.

2. Virtual DOM:

DOM stands for Document Object Model. It is the most important part of the web as it divides into modules and executes the code. Usually, JavaScript Frameworks updates the whole DOM at once, which makes the web application slow. But react uses virtual DOM which is an exact copy of real DOM. Whenever there is a modification in the web application, the whole virtual DOM is updated first and finds the difference between real DOM and Virtual DOM.

3. One-way Data Binding:

One-way data binding, the name itself says that it is a one-direction flow. The data in react flows only in one direction i.e. the data is transferred from top to bottom i.e. from parent components to child components. The properties(props) in the child component cannot return the data to its parent component but it can have communication with the parent components to modify the states according to the provided inputs.

4. Performance:

As we discussed earlier, react uses virtual DOM and updates only the modified parts. So , this makes the DOM to run faster. DOM executes in memory so we can create separate components which makes the DOM run faster.

5. Extension:

React has many extensions that we can use to create full-fledged UI applications. It supports mobile app development and provides server-side rendering. React is extended with Flux, Redux, React Native, etc. which helps us to create good-looking UI.

6. Conditional Statements:

JSX allows us to write conditional statements. The data in the browser is displayed according to the conditions provided inside the JSX.

8. Simplicity:

React.js is a component-based which makes the code reusable and React.js uses JSX which is a combination of HTML and JavaScript. This makes code easy to understand and easy to debug and has less code.

Write a code in React Js to display “Hello world".

```
import React from 'react';

import './App.css';

function App() {

 return (

  <div className="App">

   <h1>Hello, world!</h1>

  </div>);}

export default App;
```

Advantages and Disadvantages of React JS

Advantages	**Disadvantages**
Component-based architecture (reusability, maintainability)	High learning curve for beginners
Virtual DOM for better performance	SEO challenges with client-side rendering
Declarative syntax and simplicity	Frequent updates require ongoing learning

Strong ecosystem and community support	Boilerplate code and complex setup for state management
Cross-platform development (React Native)	Too much flexibility can lead to inconsistent codebases
JSX syntax for combining HTML and JavaScript	Performance overhead in very large applications
Reusable components for faster development	Requires additional libraries for full application features
Unidirectional data flow for predictability	JSX syntax can be confusing to new developers
SEO-friendly with SSR (e.g., using Next.js)	Heavy reliance on JavaScript can be problematic for accessibility
High performance with Virtual DOM and reconciliation	Integration with legacy systems can be difficult

How React Js Works ?

Q. What is JSX. Write JSX attributes with example.

- JSX stands for JavaScript XML.
- JSX allows us to write HTML in React.
- JSX makes it easier to write and add HTML in React.
- JSX allows you to write HTML tags directly within JavaScript code, which are then transformed into React elements. This transformation is typically done using Babel, a JavaScript compiler that converts JSX into React.createElement calls that the browser can understand.
- This is simple JSX code in React. But the browser does not understand this JSX because it's not valid JavaScript code. This is because we're assigning an HTML tag to a variable that is not a string but just HTML code.
- So to convert it to browser understandable JavaScript code, we use a tool like Babel which is a JavaScript compiler/transpiler.

JSX attributes

1. **The change of class attribute to className**

The class attribute is a much used attribute in HTML, but since JSX is rendered as JavaScript, and the class keyword is a reserved word in JavaScript, you are not allowed to use it in JSX. Use attribute className instead. JSX solved this by using className instead. When JSX is rendered, it translates className attributes into class attributes.

Example:

```
const myElement = <h1 className="myclass">Hello World</h1>;
```

2. **Creation of custom attributes**

We can also use custom attributes in JSX. For custom attributes, the names of such attributes should be prefixed by **data-*** attribute.

Example: This example has a custom attribute with the <h2> tag and we are using className attribute instead of class.

```
// Filename - App.js

import React from "react";

import ReactDOM from "react-dom";

const element = (

  <div>

    <h1 className="hello">Hello Geek</h1>

    <h2 data-sampleAttribute="sample">

      Custom attribute

    </h2>

  </div>

);

ReactDOM.render(element, document.getElementById("root"));
```

Q. What is NPM

1. **NPM (Node Package Manager)** is a **package manager** for **JavaScript** programming language, and it is the default package manager for **Node.js**.
2. NPM is used to manage dependencies in a project, allowing developers to install, update, and manage libraries or packages (i.e., reusable pieces of code) that are used in their applications.
3. NPM is an essential tool for modern JavaScript development, as it provides access to a huge ecosystem of open-source libraries, modules, and tools.

4. It helps streamline the development process by enabling easy installation and management of dependencies.

Features of NPM :

1. Package Management

- **Install Packages**: NPM allows you to install packages (libraries, tools, utilities) that your project depends on. These packages can be local (only for your project) or global (installed globally and available across projects).
- **Update Packages**: NPM helps you keep your packages up-to-date by allowing you to upgrade to newer versions of the libraries you use.
- **Uninstall Packages**: You can easily remove unnecessary or outdated packages.

2. Dependency Management

- NPM automatically handles dependencies between packages, meaning that when you install a package, it also installs any packages that the package depends on. It saves time and reduces errors in manually managing dependencies.

3. Versioning

- NPM uses **semantic versioning** (semver) to manage versions of packages. This helps developers specify which versions of a package they want to use, ensuring compatibility and stability in their projects.

4. Package Distribution

- NPM hosts a massive **registry** of open-source JavaScript libraries. The registry contains millions of reusable modules, from simple utilities to large frameworks, which are easily available to install via NPM.

5. Scripts and Automation

- NPM allows you to define custom scripts (such as build, test, and deploy scripts) in the package.json file, which can be executed via the command line. These scripts can automate common tasks like minification, transpilation, or testing.