REACTJS ASSIGNMENT

MODULE – 3 REACTJS

Submitted to :-

Mr. Raj Nagar

Submitted by :-

Janvi Panchal

1) What is React Js?

Ans:-

React.js, commonly referred to as React, is an open-source JavaScript library used for building user interfaces or UI components, particularly for single-page applications where user interfaces need to be highly dynamic and responsive. It was developed and is maintained by Facebook.

Key features and concepts of React include:

1. Declarative Syntax: React allows developers to describe how the UI should look and behave in a declarative way. Developers specify what they want to achieve, and React takes care of updating the DOM to match the desired state.
2. Component-Based Architecture: React applications are built using components, which are self-contained and reusable pieces of code that represent different parts of the user interface. These components can be composed to create complex UIs.
3. Virtual DOM: React uses a virtual DOM to improve performance. Instead of directly manipulating the browser's DOM, React creates a virtual representation of it in memory. When changes occur, React calculates the most efficient way to update the real DOM, reducing the need for direct and potentially costly manipulations.
4. One-Way Data Binding: React follows a unidirectional data flow, where the data flows in a single direction from parent components to child components. This helps in maintaining a predictable state and makes it easier to understand how changes in the application affect its state.
5. JSX (JavaScript XML): React uses JSX, which is a syntax extension for JavaScript that looks similar to XML or HTML. JSX allows developers to write HTML-like code in their JavaScript files, making it more readable and convenient for defining UI components.
6. React Native: React can also be used to build mobile applications through React Native. With React Native, developers can use React principles to create native mobile applications for iOS and Android platforms.

React has gained widespread popularity in the web development community due to its simplicity, efficiency, and the ability to create highly interactive user interfaces. It is often used in conjunction with other tools and libraries, such as Redux for state management, to build robust and scalable applications.

2) What is NPM in React Js?

Ans:-

In React.js, NPM (Node Package Manager) is a package manager used to install and manage external libraries and dependencies for a React project. NPM is a command-line tool that interacts with a registry of packages containing JavaScript code and associated metadata. It is widely used in the JavaScript ecosystem, including React development, to streamline the process of adding, updating, and managing project dependencies.

Here are some key points about NPM in the context of React.js:

Package Installation: NPM allows you to install packages (libraries or modules) that your React project may depend on. These packages can include utility functions, UI components, state management libraries, and more.

```
npm install package-name
```

1. **Package.json:** React projects typically include a **package.json** file that lists the project's dependencies, including the version numbers. This file can be generated using the **npm init** command and is crucial for managing dependencies and sharing the project with others.
2. **Node Modules:** When you install a package using NPM, it downloads the package and its dependencies into a folder called **node_modules** in your project directory. This folder contains all the third-party libraries required for your project.
3. **Scripts:** NPM allows you to define custom scripts in the **package.json** file. These scripts can be used to automate various tasks, such as starting the development server, building the project, or running tests.

```
"scripts": {
"start": "react-scripts start",
"build": "react-scripts build",
"test": "react-scripts test",
 "eject": "react-scripts eject"
  }
```

3) What is Role of Node Js in react Js?

Ans:-

1. **Server-side Rendering (SSR)**: Node.js can be used to implement server-side rendering for React applications. With SSR, the server pre-renders the initial state of the React components and sends the fully-rendered HTML to the client, which can improve initial loading performance and SEO.
2. **Development Environment**: Node.js is often used as the runtime environment for the development server when working with React. Tools like Create React App, Next.js, and Gatsby, which are commonly used in React development, are built on top of Node.js. Node.js provides a convenient way to set up a local development server, manage dependencies, and run build scripts.
3. **Package Management**: Node.js comes with npm (Node Package Manager) or yarn, which are commonly used for managing dependencies in React projects. React developers often use npm or yarn to install and manage third-party libraries, tools, and scripts needed for their projects.
4. **Build Tools**: Node.js can be utilized along with various build tools like Webpack, Babel, and Gulp to bundle, transpile, and optimize React code for production. These tools are often configured and run as Node.js scripts in a React project's build pipeline.
5. **API Integration**: In many React applications, Node.js is used to implement the backend server that serves as the API endpoint for the frontend React application. Node.js provides a non-blocking, event-driven architecture that is well-suited for handling asynchronous operations, making it a popular choice for building API servers.

In summary, Node.js is an integral part of the React.js ecosystem, providing tools, runtime environment, and server-side capabilities that complement and enhance the development and deployment of React applications.

4) What is CLI command In React Js?

Ans:-

React has its own command-line interface (CLI) commands. However, these CLI commands are currently only used to create a passable version of a react application using the command line.

5) What is Components in React Js?

Ans:-

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components, in this tutorial we will concentrate on Function components.

6) What is Header and Content Components in React Js?

Ans:-

Headers are compositions that extend standard navbar functionalities. They contain additional components like a jumbotron, sub-navbar, or image covers which serve as a containers for extra navigation elements - usually links, forms, or call-to-action buttons.

7) How to install React Js on Windows, linux Operating System? How to install NPM and How to check version of NPM?

Ans:-

- Windows :

```
npm -g install create-react-app

npm Start
```

- Install NPM:

```
npm  Install
```

- Check version of NPM:

```
Cmd: npm -v
```

8) How to check version of React Js?

Ans:-

- `Cmd: create-react-app --version`

9) How to change in components of React Js?

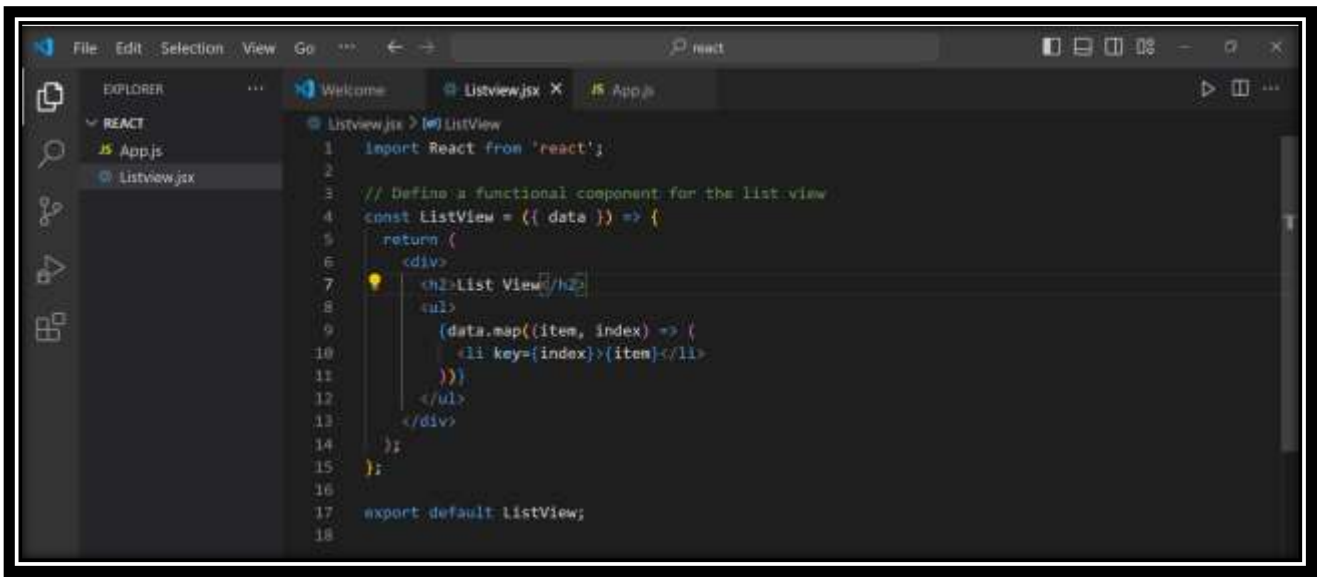
Ans:-

SetState() enqueues changes to the component state and tells React that this component and its children need to be re-rendered with the updated state. This is the primary method you use to update the user interface in response to event handlers and server responses.

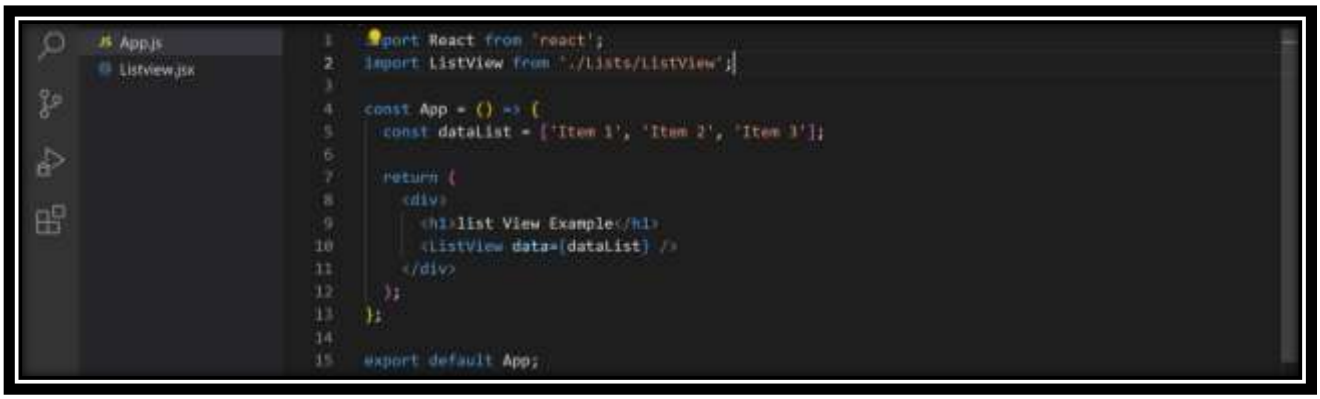
10)How to Create a List View in React Js?

Ans:-

Code:-



```jsx
import React from 'react';

// Define a functional component for the list view
const ListView = ({ data }) => {
  return (
    <div>
      <h2>List View</h2>
      <ul>
        {data.map((item, index) => (
          <li key={index}>{item}</li>
        ))}
      </ul>
    </div>
  );
};

export default ListView;
```



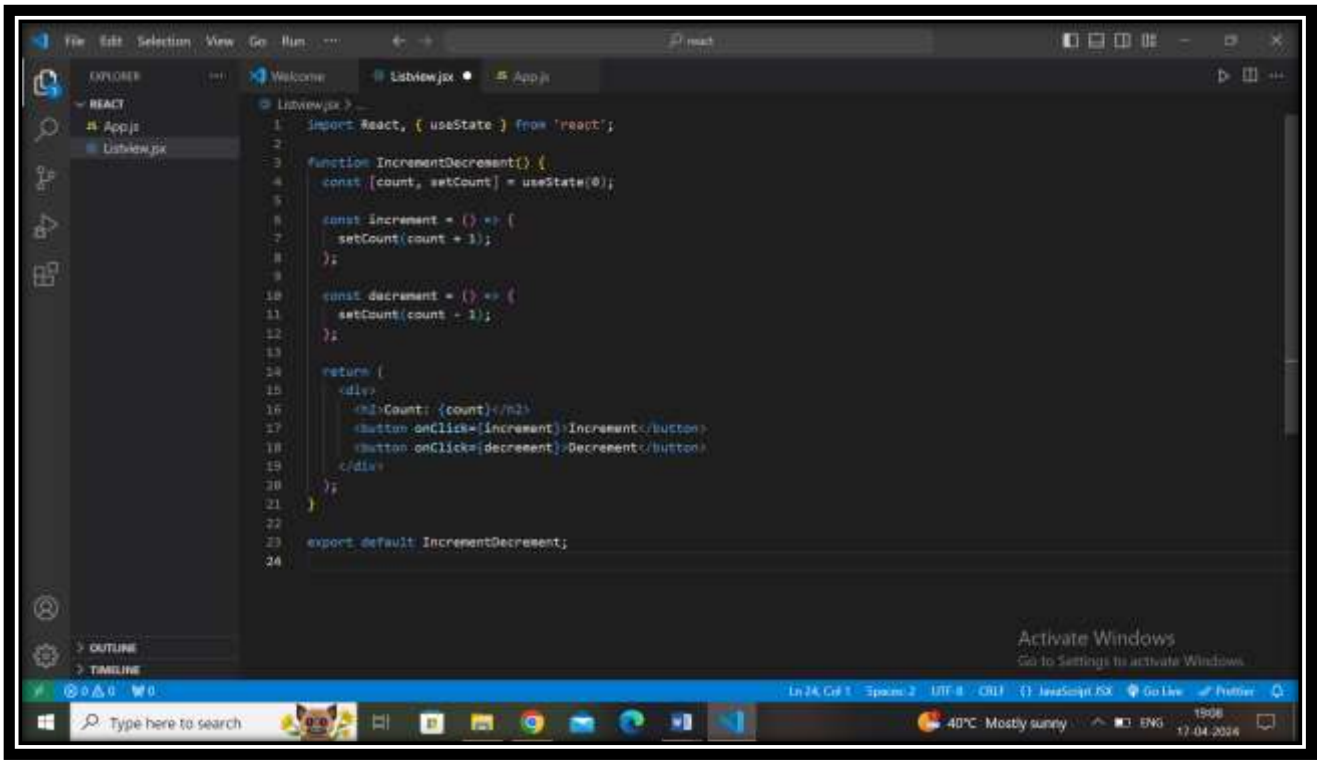
```jsx
import React from 'react';
import ListView from './lists/ListView';

const App = () => {
  const dataList = ['Item 1', 'Item 2', 'Item 3'];

  return (
    <div>
      <h1>list View Example</h1>
      <ListView data={dataList} />
    </div>
  );
};

export default App;
```

Output:-

11) Create Increment decrement state change by button click?

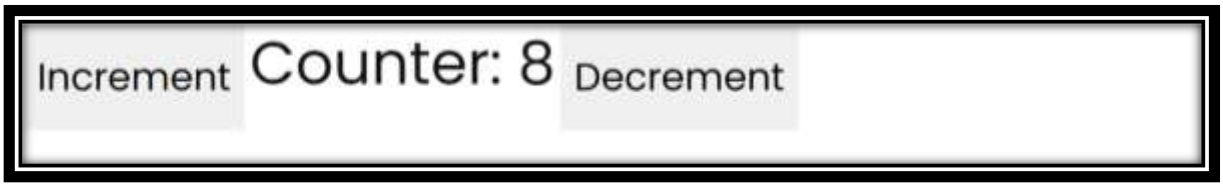
Ans:-

Code:-



Output:-

Increment



Decrement