Q.1• What is React Js?

react is a open source javascript library.it's used for a building [UI (user interface)](ReactjsAssigment.docx)

->its handeling the web Application.

->its was devloped by facebook.

->react desined to easily simplyfy for dynamic ,interective we application.

Q.2• What is NPM in React Js?

->Node Package Manager

->its packege to javascript easily install ,manage and share the centrel repository.

->Initialize a new React project

npx create-react-app my-app

-> Navigate to your project directory

cd my-app

Q.3What is Role of Node Js in react Js?

Node.js and React.js are both JavaScript-based technologies commonly used in modern web development. While they serve different purposes, they can complement each other in various ways. Here's the role of Node.js in a React.js application:

1. **Server-Side Rendering (SSR) and Universal Applications**
2. **APIs and Backend Services**
3. **Build and Development Tools**
4. **Server Environment for Testing**
5. **Deployment and Hosting**
6. **Full Stack JavaScript Development**

Q.4What is CLI command In React Js?

**Creating a New React App:**

npx create-react-app my-app

Q.5 What is Components in React Js?

There are two main types of components in React

1. **Functional Components (Stateless Components):**

**import React from 'react';**

**function Welcome(props) {**

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

**}**

**export default Welcome;**

1. **Class Components**

import React, { Component } from 'react';

class Counter extends Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

render() {

return (

<div>

<p>Count: {this.state.count}</p>

<button onClick={() => this.setState({ count: this.state.count + 1 })}>

Increment

</button>

</div>

);

}

}

export default Counter;

Q.6 What is Header and Content Components in React Js?

The Header component typically represents the top section of a web page or application

The Content component represents the main content area of your web page or application.

Q.7How to install React Js on Windows, linux Operating System?

To install React.js on both Windows and Linux operating systems, you will need to set up Node.js and npm (Node Package Manager) first because React is a JavaScript library that relies on these tools for development. Here are the general steps for installation on both operating systems.

Q.8How to install NPM and How to check version of NPM?

npm -v

npm install -g npm

Q.9How to check version of React Js?

npm list react

**Programmatically within Your React Application:**

import React from 'react';

console.log(React.version);

Q.10How to change in components of React Js?

In functional components, you can use the useState hook to manage component state:

import React, { useState } from 'react';

function MyComponent() {

const [value, setValue] = useState('Initial Value');

const handleChange = () => {

// Update the component's state

setValue('New Value');

}

return (

<div>

<p>{value}</p>

<button onClick={handleChange}>Change Value</button>

</div>

);

}

export default MyComponent;

Q.11• How to Create a List View in React Js?

Before you create a list view, make sure you have a React project set up. You can use tools like create-react-app to quickly scaffold a new React application. If you've already set up your project, you can skip this step.

2. Create a Component for the List View:

You should create a new component that will represent your list view. This component will be responsible for rendering the list items.

jsx

Copy code

// ListView.js

import React from 'react';

function ListView({ items }) {

return (

<ul>

{items.map((item, index) => (

<li key={index}>{item}</li>

))}

</ul>

);

}

export default ListView;

Q.12Create Increment decrement state change by button click?

To create a simple React component that increments and decrements a state value by button click, you can use the useState hook for functional components.

import React, { useState } from 'react';

function Counter() {

// Initialize a state variable called "count" with an initial value of 0

const [count, setCount] = useState(0);

// Function to handle incrementing the count

const increment = () => {

setCount(count + 1);

};

// Function to handle decrementing the count

const decrement = () => {

setCount(count - 1);

};

return (

<div>

<h1>Counter</h1>

<p>Count: {count}</p>

<button onClick={increment}>Increment</button>

<button onClick={decrement}>Decrement</button>

</div>

);

}

export default Counter;

Module – 4 Lists and Hook

Q.1• Explain Life cycle in Class Component and functional component with Hooks?

Mounting Phase:

constructor(): This is the first method called when an instance of the class is created. It's used for initializing state and binding methods.

componentWillMount(): Deprecated. It was called just before the component was inserted into the DOM.

render(): This method is called to render the component and return the JSX for it.

componentDidMount(): This is called immediately after the component is inserted into the DOM. It's often used for data fetching, subscriptions, or setting up timers.

Updating Phase:

componentWillReceiveProps(nextProps): Deprecated. Called when the component is about to receive new props.

shouldComponentUpdate(nextProps, nextState): This method is used to control whether the component should re-render after receiving new props or state. It can optimize performance by preventing unnecessary renders.

componentWillUpdate(nextProps, nextState): Deprecated. Called just before the component is updated.

render(): Re-renders the component with updated props or state.

componentDidUpdate(prevProps, prevState): Called after the component is re-rendered due to changes in props or state.

Unmounting Phase:

componentWillUnmount(): Called just before the component is removed from the DOM. It's used for cleanup tasks like cancelling network requests or clearing timers.

\*Functional Component with Hooks Lifecycle:

import React, { useState, useEffect } from 'react';

function FunctionalComponentWithHooks() {

const [count, setCount] = useState(0);

// componentDidMount equivalent

useEffect(() => {

console.log('Component did mount');

return () => {

console.log('Component will unmount');

};

}, []);

// componentDidUpdate equivalent

useEffect(() => {

console.log('Component did update');

}, [count]);

return (

<div>

<p>Count: {count}</p>

<button onClick={() => setCount(count + 1)}>Increment</button>

</div>

);

}

export default FunctionalComponentWithHooks;

Module-5) React - Styling & Advance React

Q.1Create Shopping site home page with Styled- component.

import React from 'react';

import styled from 'styled-components';

const HomePage = () => {

return (

<Container>

<Header>

<h1>My Shopping Site</h1>

<p>Welcome to our online store!</p>

</Header>

<ProductList>

<Product>

<ProductImage src="product1.jpg" alt="Product 1" />

<ProductName>Product 1</ProductName>

<ProductPrice>$19.99</ProductPrice>

</Product>

<Product>

<ProductImage src="product2.jpg" alt="Product 2" />

<ProductName>Product 2</ProductName>

<ProductPrice>$24.99</ProductPrice>

</Product>

{/\* Add more products here \*/}

</ProductList>

</Container>

);

};

const Container = styled.div`

max-width: 1200px;

margin: 0 auto;

padding: 20px;

`;

const Header = styled.div`

text-align: center;

margin-bottom: 20px;

`;

const ProductList = styled.div`

display: flex;

flex-wrap: wrap;

justify-content: space-around;

`;

const Product = styled.div`

width: 300px;

margin: 20px;

padding: 10px;

border: 1px solid #ddd;

text-align: center;

`;

const ProductImage = styled.img`

max-width: 100%;

`;

const ProductName = styled.h2`

margin: 10px 0;

`;

const ProductPrice = styled.p`

font-weight: bold;

`;

export default HomePage;