**React Js questions**

1. **What is React js?**
2. **What is component and types of components and syntaxs?**
3. **What is return ()?**
4. **What is export and default?**
5. **What is difference between .js and .Jsx?**
6. **What is Nested components in React Js?**
7. **What is rfce?**
8. **What is const?**
9. **What is ()=>function ?**
10. **What is React js?**

React.js is a **JavaScript library** for **building** user **interfaces**, **especially** for **web** **applications**. It allows developers to create reusable components, making it easier to manage and build complex UIs efficiently.

1. **What is component and types of components and syntaxs?**

It is a **reusable** **unit** of a **web page's** UI. It can reuse across your application.

**Types:**

1. **Functional Components:** Simple functions that **return UI elements**. function MyComponent() {
2. return <h1>Hello, World!</h1>; }
3. **Class Components:** Classes that create UI with a **render** **method**.
4. **Pure Components:** Optimized class components that minimize re-rendering.
5. **Higher-Order Components:** Functions that enhance components by adding extra functionality.
6. **What is return ()?**

**return()** is used **within** a **component** to specify **what** the **component** should **render**.

1. **What is export and default and render?**
2. **What is difference between .js and .Jsx?**
3. **What is Nested components in React Js?**
4. **What is rfce?**
5. **What is const?**
6. **What is arrow function =()=>?**

It is a way to **define a function and stored** it in a **variable** and it easy to **reuse**

**Syntax: const functionName =() =>{}**

1. **<Section> Tag**

<Section> tag in HTML is used to group related content together. ex: This groups the heading and paragraph together as a section of the webpage.

**Day3**

**Applying styles**

**Image Tag**

**Props (**short for "properties"**)?**

It is way to **pass** **data** **from** a **parent** component to a **child** **component** in **one** **way** communication **OR**

A parent component can pass data to its child components via props.

Then child component uses these props to render/call content from parent class

**Day4**

**Destructuring**?

Destructuring is used to make the code cleaner and easier to read. **OR**

It is a **JavaScript feature** that allows you to **extract values from objects** or **arrays** and **assign them to variables** in a more concise and readable way. It's often used in React to simplify code, especially when dealing with **props, state, and other objects.**

 **Destructuring Props**: Makes it easier to extract specific props.

 **Destructuring State**: Simplifies **access** to **state** **values** and **updater** **functions**.

 **Destructuring Function Arguments**: Directly extracts properties from objects passed as arguments.

**Day 5**

**Use state management? useState** hook in React is used to **add state(data) to functional components. OR**

State management in React is the process of **handling the data** (state) that **controls a component's** behaviour and **rendering**.

**useState** **Syntax :**

const [state, setState] = useState();

**Hooks** are functions that **add** React **features** like **state** and **side** **effects** to **functional** components.

**Prev(**previous**):** it is **used** in **state** update functions to refer to the previous value of the state before its updated

**prev** usually **refers** to the **previous** **state** or object.

**...prev** **copies** **all** the **properties** from the **prev** **state** or object into a **new** **object** or array.

**Use effect?**

**useEffect** hook in React is used to **handle side effects in functional components**. Side effects include tasks like **data fetching, subscriptions,** or manually updating the DOM.

**useEffect Syntax:**

  useEffect(() => { },[]);

**Day 7**

setTimeout(() => {}, time) is a function that **runs some code after a delay.**

**setTimeout**: A built-in JavaScript **function** that **sets a timer** to execute a **call-back function** after a specified delay.

 **() => {}`**: This is an arrow function, which serves as the call-back that will be executed after the delay.

**Day 8**

**Home work:**

**useContext** is a React hook that allows functional components to **access** and **use** **data** from a **context** **directly**, **without** passing **props** manually at every level of the **component** **tree**. It **share** the **data** **globally** across **multiple** **components**. (e.g., user data, themes, language settings).

### When to Use useContext: When you need to share data globally or across multiple components

1. **Create context** // **it shares the data** across **multiple** **components** without props

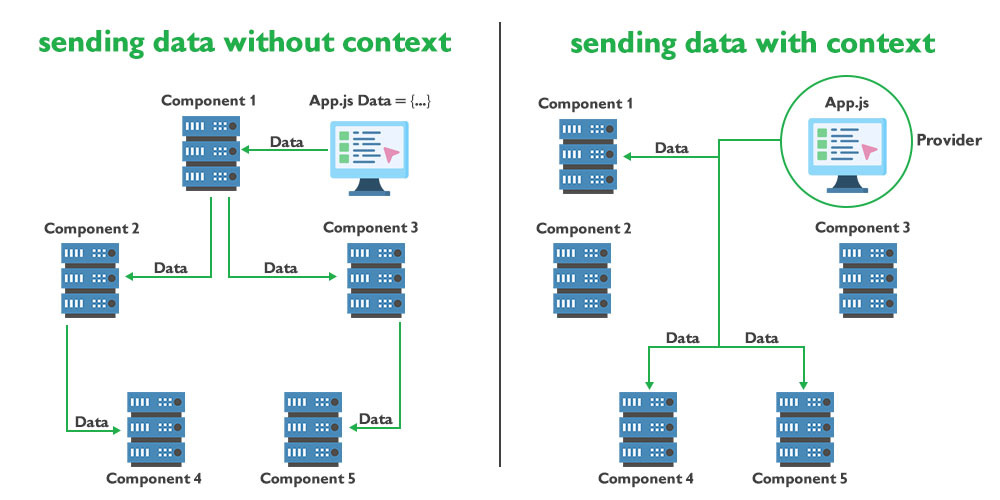
const UserConstVar = createContext();

1. **Provide the context value** // allows you to **share data** (like user) with child components, without props at every level.

<UserConstVar.Provider value={user}>

1. Consume/**access** the data **(context value**)

 const user = useContext(UserConstVar)



**Why useContext?**

**Efficiency**: **Avoids** the need to pass **props** through intermediate **components**, **reducing** **boilerplate**.

**useRef** **creates** a **reference object** to **store** a **value** or **DOM** element, that **doesn't** **change**, **even** when the **component** **re**-**renders**.

**Uses**

 **Access DOM elements** directly (e.g., focus an input).

 **Store values** between renders without causing re-renders.

 **Avoid unnecessary re-renders**, improving performance.

**Day9**

**Npm** Node Package Manager. It is the **default package manager** for **Node.js**, and **it** helps developers **install**, **manage**, and **share** **reusable** **code** (called packages or modules).

**Getting Started**

**Install dependencies:**

**npm install //it** is a command used to install packages (libraries or modules) that your project depends on.

**Start the server on http://localhost:3000 (http://localhost:3000/) :**

**npm run dev //it** is a command that **starts** the **development** **server** for your **project**. It **runs** a **script** named "dev" **defined** in your **package**.**json**

**Run unit tests:**

**npm test //it** automatically runs the tests and it will check if these tests pass or fail, helping you catch any mistakes

**npm run test:e2e //** means running **end-to-end (E2E) tests and testing** the **entire application flow**, **from** the **user interface** to the **database**, to **ensure** **everything** **works** as expected from a **user's perspective.**

**Ex:** The goal is to verify that all components (like UI, APIs, and database) work together properly.

**Linting and Formatting the Code**

**npm run lint // it** will **check** your **JavaScript files** for any **issues** **like** missing **semicolons**, **unused** **variables**, or **incorrect formatting**.

**npm run format //** it automatically adjust the style of your code to follow a specific format or style guide

**Your Task**

**1. Complete the Button component: src/components/Button/Button.jsx .**

**2. Style the Button component using CSS: src/components/Button/Button.css .**

**Component Properties**

**Implement the component to accept the props below:**

**Button.propTypes = {**

**children: PropTypes.node.isRequired,**

**9/11/24, 3:28 PM 2.11 Code Challenge 8: Create a React Button Component**

**https://ensign.instructure.com/courses/20792/assignments/1263132?module\_item\_id=2579071 2/7**

**onClick: PropTypes.func,**

**};**

**Set default values for the props using defaultProps :**

**Button.defaultProps = {**

**onClick: () => {},**

**};**

**Component Structure**

**The component should render the following structure:**

**<button data-testid="button" class="btn">Click me</button>**

**Component Behavior**

**The component renders a <button> element with the class name btn .**

**The button's text content is specified by the children prop.**

**Clicking the button triggers the provided onClick event handler function.**

**The button element includes a data-testid attribute with the value "button" for easy identification and testing purposes.**

**Component Styling**

**Apply CSS styling to the component to ensure it has the appearance described below:**

**Button Styles ( btn ):**

**padding : 8px 12px - Apply 8 pixels of padding on the top and bottom, and 12 pixels on the left and right sides of the button.**

**border-radius : 6px - Round the corners of the button with a border radius of 6 pixels.**

**border-width : 1px - Set the border width of the button to 1 pixel.**

**border-style : solid - Apply a solid border style to the button.**

**border-color : #d4d4d8 - Set the color of the button border to a light gray shade.**

**color : #18181b - Set the text color of the button to a dark gray shade.**

**background : white - Set the background color of the button to white.**

**Button Hover Styles ( btn:hover ):**

**9/11/24, 3:28 PM 2.11 Code Challenge 8: Create a React Button Component**

**https://ensign.instructure.com/courses/20792/assignments/1263132?module\_item\_id=2579071 3/7**

**background : #f2f2f7 - Change the background color of the button to a light gray shade on hover.**

**Button Active Styles ( btn:active ):**

**background : #e5e5ea - Change the background color of the button to a slightly darker shade of gray when it is active.**

**Turnery operator ?**

It allows you to choose **between two values based on a condition** in a single line.

condition ? expressionIfTrue : expressionIfFalse

**Events?**

events are used to handle user interactions like clicks or form submissions.

### Common Event Handlers in React:

* **onClick**: Handles click events.
* **onChange**: Handles changes in form inputs.
* **onSubmit**: Handles form submissions.
* **onKeyDown**: Handles keyboard key presses.

**Event listener**

Friday(Indian Saturday):

React Form Events ,

NPM Explanation,

How to show array data,

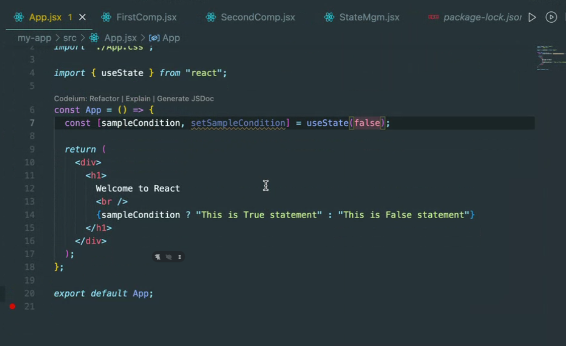
how to get values from object,

how to get values from nested objects,

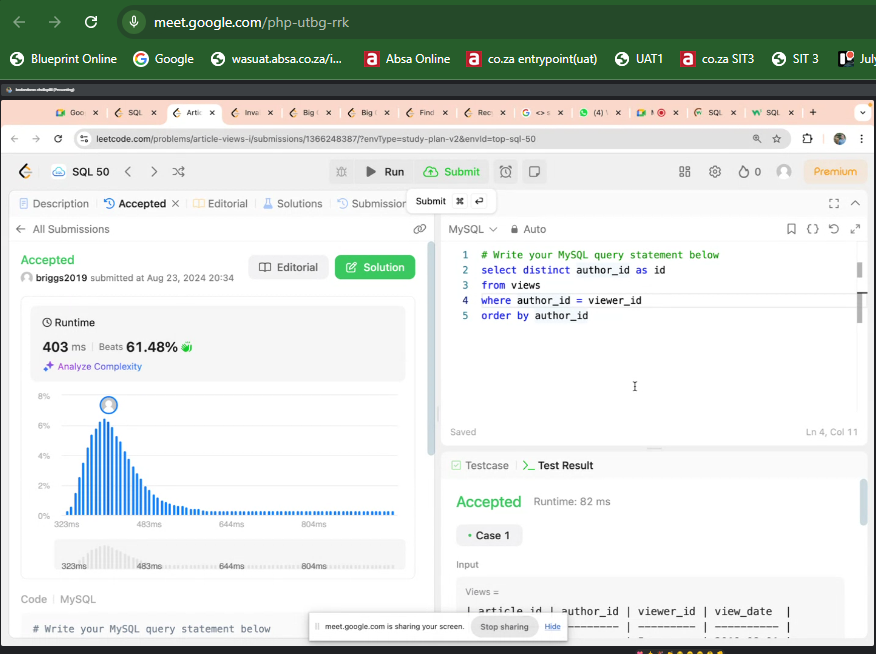
Template literals,

Fetch Method,

Error Hadndling.



SQL question 4



**SQL Question 5**

