

## Assignment 6

### Owner – Mohit R

#### 1. **What is a Microservices?**

- Microservices are an architectural and organizational approach to software development where software is composed of small independent services that communicate over well-defined APIs. These services are owned by small, self-contained teams.
- Example: Netflix became one of the first high-profile companies to successfully migrate from a monolith to a cloud-based microservices architecture in 2009.

#### 2. **What is Monolith architecture?**

- Traditionally apps were built using Monolith architecture
- We used to have a hugebig project, which use to have small pieces of code like APIS, UI, Authentication, Database connectivity, SMS code (notifications)
- Everything is done in the same project
- Even to make a single change like changes to a button color the entire project needs to be build, compile, deploy this entire project
- Monolith Architecture is the traditional model of software development where the application is self-contained and independent of other applications. It is also called Single Tier (One tier) architecture where a single application acts as both client and server.

#### 3. **Difference between Monolith architecture and Microservices?**

- In the monolithic architecture we will have all our code (functionalities) inside a single project and all the functionalities should use the same programming language or tech stack.
- Here maintenance is difficult because even for a single small change we need to build, compile and deploy all our code.
- In the microservice architecture we will separate all our functionalities and they communicate with each other using APIs when required. We can use different tech stacks for each functionality (microservice).
- Here maintenance is easy because we can build, compile and deploy only the necessary code (microservice).

#### 4. **Why do we need useEffect Hook?**

- `useEffect Hook` is javascript function provided by `react`. The useEffect Hook allows you to `eliminate side effects` in your components. Some examples of side effects are: `fetching API data`, `directly updating the DOM`, and `setting up subscriptions or timers`, etc can be lead to unwarranted side-effects.
- useEffect accepts `two arguments`, a `callback function` and a `dependency array`. The second argument is optional.
- ## Syntax  
  
useEffect(() => {}, [])

- The `() => {}` is callback function and `[]` is called an empty dependency array.
- If we pass anything (suppose `currentState`) inside the `[]` it triggers the callback function and changes the state of the application.

```
useEffect(() => {
  setCurrentState("true");
}, [currentState])
```

## 5. What is Optional Chaining?

- 'Optional chaining' is a JavaScript feature that allows you to access the properties of an object and elements of an array without having to check whether the object or array is null or undefined is known as Optional chaining.
- 'Optional Chaining' `?.` and `[]?.` is good way of accessing the object keys, it prevents the application from crashing and if the key that we are trying to access is not present, then instead of throwing a key error, it returns undefined.

## 6. What is Shimmer UI?

- A Shimmer UI resembles the actual UI page, so that the user will understand how quickly the web or mobile app will load even before the content has shown up
- Fake cards are shown till the data is shown on the UI
- It has a much better User Experience

## 7. What is the difference between JS expression and JS statement?

- A 'JS expression' returns a value that we use in the application. for example:

```
4 + 2 // output 6
"foo".toUpperCase() // output 'FOO'
console.log(2) // logs '2'
isTrue ? true : false // returns us a true or false value based on isTrue value
```

- A 'JS statement', performs/ executes an action but does not return a value. for example:

```
let x; // variable declaration
if () { } // if condition
```

- If we want to use 'JS expression' in JSX, we must wrap in `{/* expression slot */}` and if we want to use 'JS statement' in JSX, we must wrap in `{/* statement slot */}`;

## 8. What is conditional Rendering?

- Rendering based on a certain condition is called as conditional rendering
- In React, you can conditionally render JSX using JavaScript syntax like if statements, `&&`, and `?:` operators.
- // Using Ternary operator as a shorthand way or writing an if-else statement  
`isLoggedIn ? (return <UserGreeting />) : (return <GuestGreeting />);`

// Using an if...else Statement

```

    {
        (if (isLoggedIn) {
            return <UserGreeting />;
        } else {
            return <GuestGreeting />;
        })
    }

```

### 9. What is `CORS`?

- Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.
- CORS defines a way in which a browser and server can interact to determine whether it is safe to allow the cross-origin request.
- Our browsers are blocking us to call an API from one origin to another origin when they don't match
- Swiggys API from localhost (mis-match)
- If there is an origin mismatch the browser blocks that API call

### 10. What is `async and await`?

- **`Async`**: It simply allows us to write promises-based code as if it was synchronous and it checks that we are not breaking the execution thread. It operates asynchronously via the event loop. Async functions will always return a promise. It makes sure that a promise is returned and if it is not returned then JavaScript automatically wraps it in a promise which is resolved with its value.
- **`Await`**: Await function is used to wait for the promise. It could be used within the `async` block only. It makes the code wait until the promise returns a result. It only makes the async block wait.

**## for example:**

```

// async function getRestaurant to fetch Swiggy API data
async function getRestaurants() {
    const data = await fetch(
        "Swiggy_API_URL"
    );
    const json = await data.json();
    // we get the Swiggy API data in json format
    console.log(json);
}

```

### 11. What is the use of `const json = await data.json();` in `getRestaurants()`?

- The `data` object, returned by the `await fetch()`, is a generic placeholder for multiple data formats. so we can extract the `JSON object` from a `fetch` response by using `await data.json()`.
  - `data.json()` is a method on the data object that lets you extract a `JSON object` from the data or response. The method returns a promise because we have used `await` keyword.

- so `data.json()` returns a promise resolved to a `JSON object`.