Monolith Architecture

A monolithic architecture is a traditional approach of software development. In Monolithic architecture entire application is designed and deveolped as an single unit. For example, a traditional application will have a frontend, API, services, load balancer, and database. If you build everything together and deploy it on the server, than it is called a monolithic architecture, where services tightly coupled together.

Microservices

Microservices are an architectural approach in which entire application developemnet is divided into smaller and independent services. These serives can be developed ,deployed and maintained separately.Within a microservices architecture, each microservice is a single service built to accommodate an application feature and handle discrete tasks. Each microservice communicates with other services through simple interfaces to solve business problems.

different between Monolith and Microservice architecture

Application Architecture Comparison

Monolith has a simple straightforward structure of one undivided unit. Microservices have a complex structure that consists of many different services and databases.

Scalability Comparison

Monolithic application is scaled as a whole single unit, but microservices can be scaled unevenly, and this will save business and development teams a lot of time and resources. We can do in cahnges in one service and can deploy it on server without affecting the other services but in monolith architecture if we do changes in one service others also we need to deploy again.

Troubleshooting Comparison

Large monolith is harder to troubleshoot due to various dependencies and tight coupling between system components. But, Microservices enable fast and easy problem tracing and troubleshooting.

Resilience Comparison

Microservices applications are more resilient thanks to independent deployment and loose coupling if we compared to tight coupling in monolith applications.

Q: Why do we need a useEffect Hook?

Difference between function statement and function expression

Function statement

if we write our function by using function keyword than it is called function statement

function A()

{

console.log("A Called");

}

During memory creation it is treated as function and entire code is written as it value.That make us enable to call them before their writing.

Function Expression

var b=function (){

console.log("B called");

}

If we write our function like the expression to assign a variable than it is called function expression.During memory creation it will be treated as variable and have undefined as its value.

Conditinal Rendering

Sometimes we need to show different things depending on different conditions. It can be done in same way we use if else or ternery operator in javascript. ex-- using if else

if (condition) {

return <li >{name} ✔</li>;

}

return <li >{name}</li>;

ex- using ternery operator

return (

{condition ?<li>True compoenent</li>: <li>False compoenent</li>}

</li>

);

What is async await

async is a keyword that is used to create async function in javascript. async keyword makes the execution of asynchronous function look like synchronous function execution.await is a keyword that can be used only in async function . await waits at the same line where we are expecting the promise result till that promise resolved it will not execute the next line of that function till the promise resolved but it doesn't mean that it will block the call stack it will remove that function from call stack until that promise reolved once that promise gets resolved it will again put that method in call stack.

ex--

async function asyncCall() {

console.log('calling');

const result = await returningPromise();

console.log(result);

}

Q: What is the use of const json = await data.json(); in getRestaurants()?

data was the object that was returned by the fetch() this conatins many other things along with data in different formats and also json function. Json function returns a promise with data in json format so we are calling json function to get the data in json format and we are suing await to wait till that promise resolved.

Q: What is Shimmer UI?

Shimmer UI is just a better way to Show Loading States.A shimmer UI is a replica of the actual UI that doesn’t contain actual content, but instead shows the layout and shapes of the content that will eventually appear that gives the better user experience.

Optinal chaining

The optional chaining ?. is a safe way to access nested object properties, even if an intermediate property doesn’t exist. if we are using a property of nested object and accessing a property that doesn't exist than ?. operator will stop executing further and will return the undefined instead of throwing exception

let user = {}; // a user without "address" property

alert(user.address.street);

this code will go to the exception.

let user = {}; // a user without "address" property

alert(user?.address?.street);

this code will return the undefined

what is useEffect

useEffect is built-in hook of react. It takes two argument 1- A callback function 2- A dependency array while rendering our component a function gets registered and it executed once our component gets rendred it always excuted after first render but for other renders it depends on the second aregument dependency array that we given.If it finds that the value of the dependency array is same as previous than it skips the again executing that callback function

ex--

import {useEffect} from React

useEffect(()=>{

console.log("useEffect Called");

},[dependecy1,dependecy2])