CHAPTER 06 – Exploring the World

**MicroServices Vs Monolithic, Hooks-UseEffect, CORS, async/await, Shimmer Effect 🔥**

1. **What is a Microservice?**

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.

Many small projects with same domain name

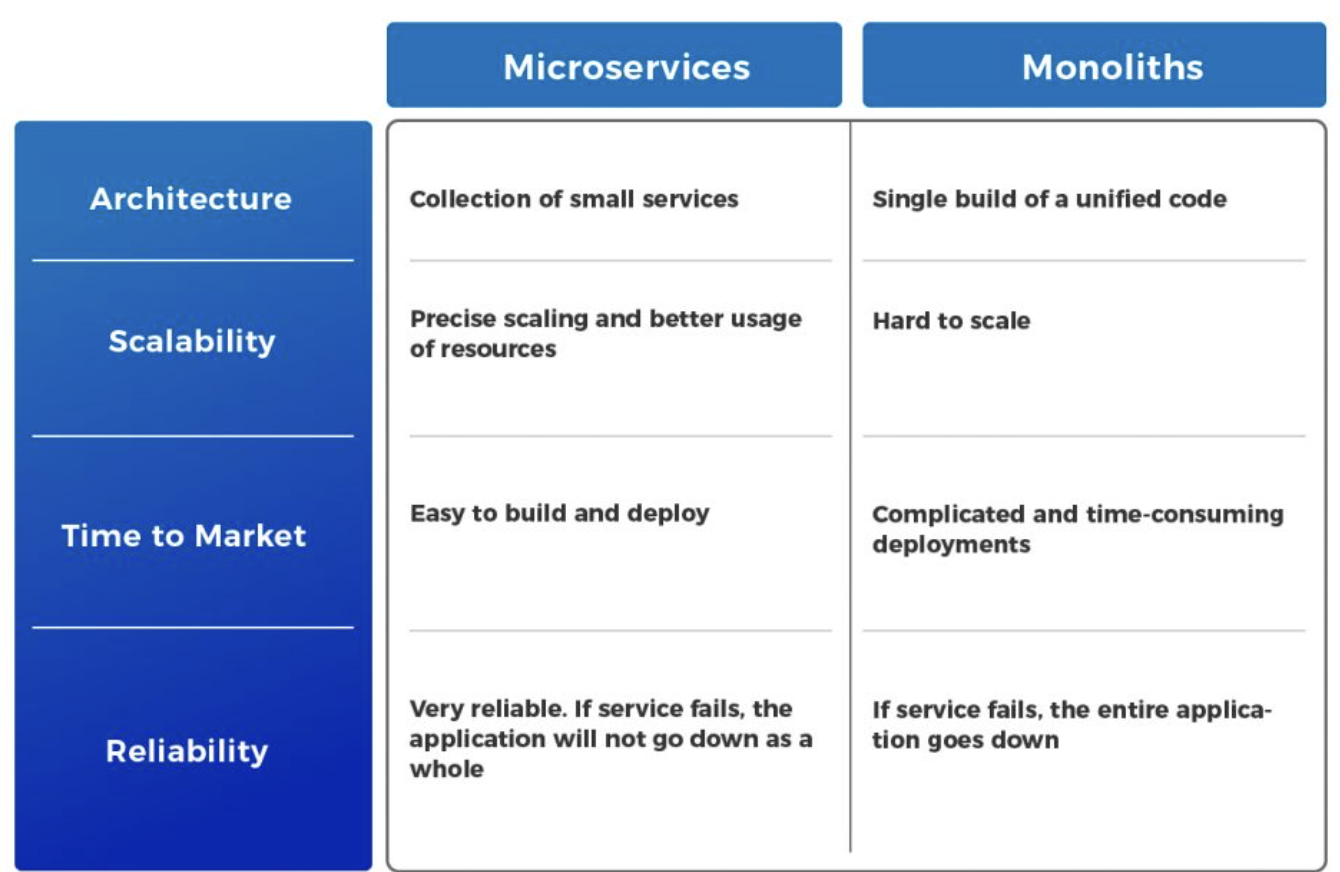


With a microservice architecture, we’ll **divide the application into multiple “projects”**.

**2. What is Monolith architecture?**

A monolithic architecture is **a singular, large computing network with one code base that couples all of the business concerns together**. To make a change to this sort of application requires updating the entire stack by accessing the code base and building and deploying an updated version of the service-side interface.

**3. What is the difference between Monolith and Microservice?**



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

The useEffect Hook allows you to perform side effects in your components.

Some examples of side effects are: fetching data, directly updating the DOM, and timers.

useEffect accepts two arguments. The second argument is optional.

useEffect(<function>, <dependency>)

**1. No dependency passed:**

useEffect(() => {  
 //Runs on every render  
});

**2. An empty array:**

calls once after render

useEffect(() => {  
 //Runs only on the first render  
}, []);

**3. Props or state values:**

Calls once after render + every time my search text changes after every render

useEffect(() => {  
 //Runs on the first render  
 //And any time any dependency value changes  
}, [prop, state]);

So, to fix this issue, let’s only run this effect on the initial render.

**5. What is Optional Chaining?**

Optional chaining ? is **a process for querying and calling properties, methods, and subscripts on an optional that might currently be nil/undefined**. If the optional contains a value, the property, method, or subscript call succeeds; if the optional is nil/undefined, the property, method, or subscript call returns nil .

**6. Is optional chaining good or bad?**

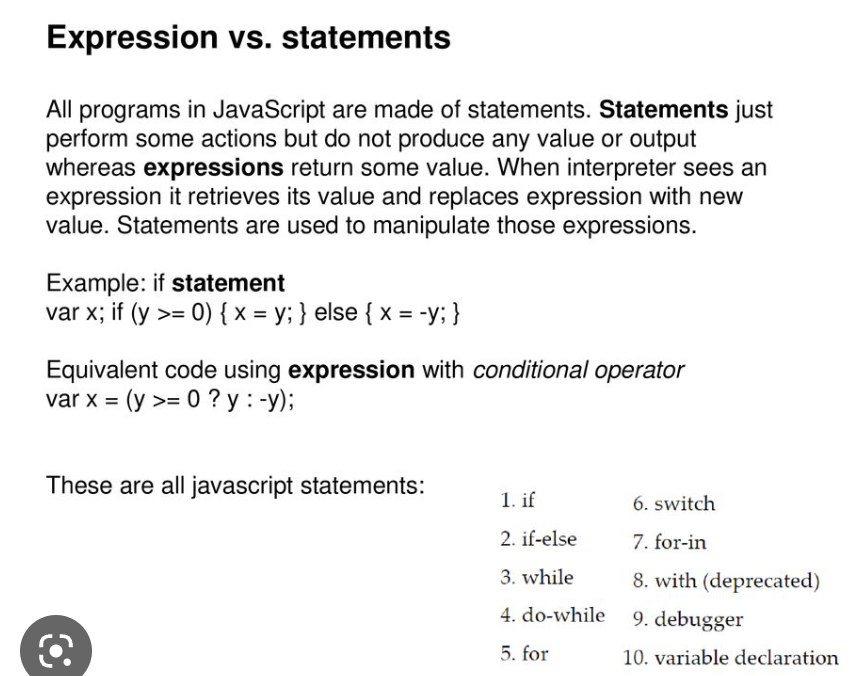
The optional chaining ?. is **a safe way to access nested object properties**, even if an intermediate property doesn’t exist.

**7. What is Shimmer UI?**

Shimmer is **a temporary animation placeholder for when data from the service call takes time to get back** and we don’t want to block rendering the rest of the UI.

Spinner is a bad user experience. Phycologist said to use empty boxes in UI to cause Shimmer Effect.

**8. What is the difference between JS expression and JS statement?**



**9. What is Conditional Rendering, explain with a code example?**

Your components will often need to display different things depending on different conditions. In React, you can conditionally render JSX using JavaScript syntax like if statements, &&, and ? : operators.

**10. 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**.

**11. 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 value. 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.

**Example 1:** In this example, we will see the basic use of async in Javascript.

* javascript

const getData = async() => {

**var** data = "Hello World";

**return** data;

}

getData().then(data => console.log(data));

**Output:**

Hello World

**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.

**Example 2:** This example shows the basic use of the await keyword in Javascript.

* javascript

const getData = async() => {

**var** y = await "Hello World";

console.log(y);

}

console.log(1);

getData();

console.log(2);

**Output:**

1  
2  
Hello World

**12. What is the use of ‘const ison = await data.json();’ in getRestaurants()?**

response. json() is **a method on the Response object that lets you extract a JSON object from the response**.

**13. Best Part of React JS ?**

Best part of react -> reconciliation

**14. How to access other app apis?**

We cant use other app apis. Browser blocks us. CORS issue.

Cors chrome Extension