**Episode 6: Exploring the world**

**Theory:**

1. What is a Microservice?

Microservices is an architectural pattern, that arranges an application as a collection of loosely coupled, fine-grained services, communicating through lightweight protocols.

In Microservice architecture, an application is built as independent components that run each application process as a service. These services communicate via a well-defined interface using lightweight APIs. Services are built for business capabilities and each service performs a single function. Because they are independently run, each service can be updated, deployed, and scaled to meet demand for specific functions of an application.

2. What is Monolith architecture?

A Monolith architecture is a traditional model of a software program, which is built as a unified unit that is self-contained and independent from other applications.

We can also say that it is designed to be self-contained, wherein the program’s components or functions are tightly coupled rather than loosely coupled. In Monolith architecture, each component and its associated components must all be present for code to be executed or compiled and for the software to run.

These types of applications are single tiered, which means multiple components are combined into one large application. Consequently, they have large codebase, which can be cumbersome to manage over time.

3. What is the difference between Monolith and Microservice?

4. Why do we need a useEffect Hook?

5. What is Optional Chaining?

6. What is Shimmer UI?

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

8. What is Conditional Rendering, explain with a code example

8. What is CORS?

9. What is async and await?

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

**Coding:**

1. Play with the useEffect Hook to see when it is called? (before or after render)

2. Play with dependency array in useEffect Hook

3. Play with the developer console by putting a debugger in render and useEffect

4. Call an actual API to get data

5. Handle Error in your API call

6. Build Shimmer UI when data in not load

7. Render your UI with actual API data

8. Make Search functionality work

9. Make a Login Logout button which toggles with a state