CHAPTER 6 - THEORY ASSIGNMENT – 'EXPLORE THE WORLD'

1. What is Microservice?

Ans:-

A microservices architecture is a type of application architecture where the application is developed as a collection of services. It provides the framework to develop, deploy, and maintain microservices architecture diagrams and services independently.

For eg: - UI can be in React, BackEnd in Java, notifications in Python,Logs in Paython and Authentication via GoLang

Features of Microservices:-

a. Separation of concerns,

b. Separate deployment as every work and task is different and independent.

c. Single responsibility.

Refer: - https://www.redhat.com/en/topics/microservices/what-are-microservices

https://aws.amazon.com/microservices/

2. What is Monolith architecture?

Ans:-

Monolithic is an architecture where services are tightly coupled with each other and run as a single service and hence difficult to scaled when more changes comes in pictures.

So basically, to change one button in UI we have to deploy whole project and build the whole application for each and every small change. Mostly old applications still follows Monolith architecture.

Refer:- https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith

3. What is Difference between Monolith and Microservice?

Ans:-

Sr. No.	Key	Monolithic architecture	Microservices architecture
1	Basic	Monolithic architecture is built as one large system and is usually one code-base	Microservices architecture is built as small independent module based on business functionality
2	Scale	It is not easy to scale based on demand	It is easy to scale based on demand.
3	Database	It has shared database	Each project and module has their own database
4	Deployment	Large code base makes IDE slow and build time gets increase.	Each project is independent and small in size. So overall build and development time gets decrease.
5	Tightly Coupled and Loosely coupled	It extremely difficult to change technology or language or framework because everything is tightly coupled and depend on each other	Easy to change technology or framework because every module and project is independent

4. Why do we need a useEffect Hook?

Ans:- We are telling React that your component needs to do something after initial render. It is mostly used for tasks like updating the DOM, fetching data from API end-points, setting up subscriptions or timers, etc can be lead to unwarranted side-effects.

useEffect Hook comes takes once parameter as callback function and parameter is dependency array which is optional.

5. What is Optional Chaining?

Ans:- By using the?.operator instead of just, JavaScript knows to implicitly check to be sureobject is notnullorundefinedbefore attempting to access it. If object is null or un defined, the expression automatically short-circuits, returningundefined. Basically we use it for null checks.

6. What is Shimmer UI?

Ans:- A shimmer UI resembles the page's actual empty boxes structured UI, so users will understand how quickly the web or mobile app will load even before the content has shown up. It gives people an idea of what's about to come and what's happening (it's currently loading) when a page full of content/data takes more than 300-500ms to load.

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

Ans:- JavaScript distinguishes expressions and statements. An expression produces a value and can be written wherever a value is expected, for example as an argument in a function call. Each of the following lines contains an expression:

```
myvar
3 + x
myfunc("a", "b")
```

Roughly, a statement performs an action. Loops and if statements are examples of statements. A program is basically a sequence of statements (we're ignoring declarations here). Wherever JavaScript expects a statement, you can also write an expression. Such a statement is called an expression statement. The reverse does not hold: you cannot write a statement where JavaScript expects an expression. For example, an if statement cannot become the argument of a function.

Refer :- https://2ality.com/2012/09/expressions-vs-statements.html

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

Ans:- Conditional rendering in React works the same way conditions work in JavaScript. Use JavaScript operators like if or the conditional operator to create elements representing the current state, and let React update the UI to match them.
Conditional rendering is a term to describe the ability to render different user interface (UI) markup if a condition is true or false.
For example, If we have load any API data in UI then it takes time to fetch API data, till then we can show Shimmer or Loader and once the data is fetched, we can display it.

```
import Shimmer from "./Shimmer";
import Body from "./Body";

{
  isLoading ? <Shimmer/> : <Body />
}
```

Refer:- https://reactjs.org/docs/conditional-rendering.html

9. What is CORS?

Ans:- **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.

Refer: - https://www.youtube.com/watch?v=tcLW5d0KAYE

10. What is async and await?

Ans:- Async functions can contain zero or more await expressions. Await expressions make promise-returning functions behave as though they're synchronous by suspending execution until the returned promise is fulfilled or rejected. The resolved value of the promise is treated as the return value of the await expression. Use of async and await enables the use of ordinary try / catch blocks around asynchronous code.

The word “async” before a function means one simple thing: a function always returns a promise. Other values are wrapped in a resolved promise automatically.

Refer:- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function

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

Ans:- The json() method of the Response interface takes a Response stream and reads it to completion. It returns a promise which resolves with the result of parsing the body text as JSON.

Note that despite the method being named json(), the result is not JSON but is instead the result of taking JSON as input and parsing it to produce a JavaScript object.

Refer:- https://developer.mozilla.org/en-US/docs/Web/API/Response/json