**Chapter 05 - Let's get Hooked!**

**1️⃣ Difference between Named export, Default export, and \* as export**

👉 In JavaScript (ES6 modules), we export/import values in 3 ways:

A screenshot of a computer

AI-generated content may be incorrect.

✅ **Interview Tip:**

*Default = one main thing,*

*Named = multiple exports,*

*\* as = import everything as object.*

**2️⃣ Importance of config.js file**

* config.js is a **central place to store constants, API URLs, keys, and app settings**.
* Example:
* // config.js
* export const API\_URL = "https://api.foodapp.com";
* export const LOGO\_URL = "https://cdn.logo.png";

Then use in components:

import { API\_URL } from "./config.js";

✅ Helps in **maintainability** (change once, reflects everywhere).

**3️⃣ What are React Hooks?**

* “Hooks are special functions introduced in React 16.8 that allow functional components to use state and lifecycle features which were earlier only possible in class components.”
* Example hooks: useState, useEffect, useContext, useRef.

🡪**State = memory of component (data it can change & remember).**

🡪**Lifecycle = what should happen when the component starts, updates, or ends.**

✅ Before hooks → had to use **class components**.  
✅ After hooks → can do everything with **functional components**.

**4️⃣ Why do we need useState Hook?**

* In React, UI updates when **state changes**.
* useState allows a component to have its own **local state**.

Example:



✅ Without useState, variables don’t trigger re-render.  
✅ With useState, React knows when to re-render.

⚡ **Interview Short Answers:**

* **Named export:** export/import by same name.
* **Default export:** one main export, can rename while importing.
* **\* as:** imports everything as object.
* **config.js:** keeps constants, API URLs, improves maintainability.
* **Hooks:** functions to use React features in functional components.
* **useState:** allows functional components to have reactive state.