🧩 **Task 3: Functions** — a key concept you'll use constantly in both React and backend development.

✅ **Task 3: Functions in JavaScript**

🎯 **Goals:**

- Understand how to declare and use functions

- Know the difference between function declarations, expressions, and arrow functions

- Use default parameters and return values

- Learn how to pass functions as arguments (callbacks)

📘 **Concepts to Learn**

🔸 **1. Function Declaration**

function greet(name) {

 return `Hello, ${name}!`;

}

🔸 **2. Function Expression**

const greet = function(name) {

 return `Hello, ${name}!`;

};

🔸 **3. Arrow Functions (ES6)**

const greet = (name) => `Hello, ${name}!`;

◆ 4. Default Parameters

```
function registerUser(name = "Guest") {

  console.log(`Registered: ${name}`);

}

registerUser(); // "Registered: Guest"
```

◆ 5. Returning Values

```
function add(a, b) {

  return a + b;

}

let sum = add(5, 3); // sum = 8
```

◆ 6. Callback Functions

```
function processInput(input, callback) {

  callback(input.toUpperCase());

}


processInput("hello", function(result) {

  console.log(result); // "HELLO"

});
```

🧪 Practice Exercises

✅ Exercise 1: Greeting Function

Write a function greetUser(name) that returns a string like:

"Welcome back, Ali!"

Use arrow function syntax.

✅ Exercise 2: Simple Calculator

Create a function calculate(num1, num2, operator) that:

- Returns the result of adding, subtracting, multiplying, or dividing num1 and num2 based on the operator passed (+, -, *, /)

✅ Exercise 3: Service Processor

Write a function handleService(serviceName, callback) that:

- Accepts a string (e.g., "Car Wash") and a callback

- Calls the callback and passes the service name in uppercase

Then use it like this:

handleService("car wash", function(result) {

 console.log(`Request received for: ${result}`);

});

💬 Reflection Questions

1. What's the difference between a function declaration and a function expression?

2. What is a callback function, and why is it useful?

3. Why are arrow functions not ideal as methods inside objects?