Activity:

Q:1 You are working on a web application that requires you to manipulate data from an API response. The API response returns an object with the following structure:

const apiResponse = {

status: "success",

data: {

user: {

name: "John Doe",

age: 32,

email: "johndoe@example.com",

address: {

street: "123 Main St",

city: "Anytown",

state: "CA",

zip: "12345"

}

},

orders: [

{ id: 1, total: 100 },

{ id: 2, total: 200 },

{ id: 3, total: 300 }

]

}

};

Your task is to write a function processData() that takes the above object as an argument and returns an object with the following structure:

{

name: "John Doe",

age: 32,

email: "johndoe@example.com",

street: "123 Main St",

city: "Anytown",

state: "CA",

zip: "12345",

orders: [100, 200, 300]

}

Requirements:

1. Use destructuring to extract the necessary data from the apiResponse object.
2. Use destructuring to extract the total property from each order object in the orders array.
3. Return an object with the structure shown above.

Q2:ask: Error Handling with Try-Catch in Console

Instructions:

1. Write a function called "divideByZero" that takes two arguments, "numerator" and "denominator".
2. Inside the function, write code to divide the numerator by the denominator.
3. Use a try-catch block to handle the error that occurs if the denominator is zero. In the catch block, log an error message to the console.
4. Call the "divideByZero" function multiple times with different numerator and denominator values to test the error handling functionality.

Q3:

Task 2: Error Handling in a Function

Instructions:

1. Write a function called "checkString" that takes one argument, "str".
2. Inside the function, write code to check if "str" is a string. If it is not a string, throw an error with a message "Input is not a string".
3. Use a try-catch block to call the "checkString" function with different values, including a string and a non-string value.
4. In the catch block, log the error message to the console.