**JS-Daddies**

**Sharvayu Zade**

**Sushmit Partakke**

**Harshit Selarka**

**Activity for Opponent Team**

**Instructions:** The following JavaScript code contains multiple bugs and errors across different concepts like variable scope, data types, loops, and error handling. Your task is to identify and list all the errors.

**Part 1: Find the Errors**

Find at least **8 errors** in the code snippet below.

JavaScript

// Global Scope Variable

const API\_KEY = "XYZ-123";

function checkAdminStatus() {

console.log(`User's admin status on entry: ${isAdmin}`);

if (!isAdmin) {

var isAdmin = true;

console.log("Admin status has been set inside the block.");

}

return isAdmin;

}

function fetchUserData(userId) {

console.log(`Current level is: ${level}`);

if(userId = 0) {

throw new Error("Invalid User ID provided");

}

let level = "Beginner";

return { id: userId, score: "50" };

}

const processScores = (user, bonusPoints) => {

const totalScore = user.score + bonusPoints;

console.log(`${user.id}'s total score is ${totalScore}`);

for (var i = 1; i <= 3; i++) {

setTimeout(function() {

console.log(`Processing reward #${i} for the user...`);

}, 10);

}

API\_KEY = "ABC-456"; // Attempting to change the constant key

return totalScore;

};

function initializeUserJourney() {

try {

const admin = checkAdminStatus();

const user = fetchUserData(10);

const finalScore = processScores(user, '25');

console.log(`Final user score processed: ${finalScore}`);

} catch {

console.log('An error was caught, but not the right one', e);

}

}

initializeUserJourney();

**Part 2: Complete the Code**

Bilkul, yeh le Part 2 ek proper question ke format mein.

**Question: The Dynamic Value Processor**

**Problem Statement:**

Create a higher-order function named createChallengeRunner that acts as a factory. It should accept a single numerical argument, modifier. This factory function must produce and return a new function that can process a variable number of inputs.

The returned function's core purpose is to take any number of numerical inputs, transform each one using the modifier from its parent, and then use a provided helper function to find the highest value among the transformed numbers.

**Your Task:**

Complete the code inside the // YOUR CODE HERE block. You must implement the logic that processes the incoming numbers according to the rules below.

JavaScript

const createChallengeRunner = (modifier) => {

function findHighestValue(val1, val2, val3) {

let max = val1 > val2 ? val1 : val2;

return max > val3 ? max : val3;

}

return (...inputNumbers) => {

let processedNumbers = [];

// YOUR CODE HERE

const result = findHighestValue(...processedNumbers);

return `The highest calculated value is: ${result}`;

}

}

const runner = createChallengeRunner(5); // modifier = 5

const output = runner(10, 25, 15);

console.log(output);

**Expected Output:**

The highest calculated value is: 125