**Project: JavaScript Test Script Analysis**

Here’s a set of JavaScript scripts that simulate automated testing scenarios. Each script includes errors, incomplete test cases, and opportunities for optimization to align with the project requirements.

**Script 1: Calculator Function Tests**

This script tests the functionality of a simple calculator, but it contains errors and incomplete test cases.

JavaScript

// Calculator Functions

function add(a, b) {

return a + b;

}

function subtract(a, b) {

return a - b;

}

function multiply(a, b) {

return a \* b;

}

function divide(a, b) {

if (b === 0) {

return "Error: Division by zero"; // Incorrect handling for division by zero

}

return a / b;

}

// Test Cases

console.log("Addition Test 1:", add(5, 3) === 8 ? "Pass" : "Fail"); console.log("Subtraction Test 1:", subtract(10, 4) === 5 ? "Pass" : "Fail"); // Incorrect expected value

console.log("Multiplication Test 1:", multiply(7, 6) === 42 ? "Pass" : "Fail"); console.log("Division Test 1:", divide(12, 4) === 3 ? "Pass" : "Fail");

// Incomplete Test Cases

// Add tests for edge cases (e.g., division by zero, negative numbers, large numbers)

// No test for divide by 0 or non-integer inputs

**Correction 1**

// Calculator Functions

function add(a, b) {

return a + b;

}

function subtract(a, b) {

return a - b;

}

function multiply(a, b) {

return a \* b;

}

function divide(a, b) {

if (b === 0) {

return Infinity; // Better handling for division by zero

}

return a / b;

}

// Test Cases

console.log("Addition Test 1:", add(5, 3) === 8 ? "Pass" : "Fail");

console.log("Addition Test 2 (Negative Numbers):", add(-5, -3) === -8 ? "Pass" : "Fail");

console.log("Subtraction Test 1:", subtract(10, 4) === 6 ? "Pass" : "Fail");

console.log("Subtraction Test 2 (Negative Result):", subtract(4, 10) === -6 ? "Pass" : "Fail");

console.log("Multiplication Test 1:", multiply(7, 6) === 42 ? "Pass" : "Fail");

console.log("Multiplication Test 2 (Negative Numbers):", multiply(-7, 6) === -42 ? "Pass" : "Fail");

console.log("Multiplication Test 3 (Zero):", multiply(7, 0) === 0 ? "Pass" : "Fail");

console.log("Division Test 1:", divide(12, 4) === 3 ? "Pass" : "Fail");

console.log("Division Test 2 (Division by Zero):", divide(12, 0) === Infinity ? "Pass" : "Fail");

console.log("Division Test 3 (Negative Division):", divide(-12, 4) === -3 ? "Pass" : "Fail");

console.log("Division Test 4 (Non-Integer Division):", divide(7, 2) === 3.5 ? "Pass" : "Fail");

// Edge Cases

console.log("Large Numbers Addition:", add(1e15, 1e15) === 2e15 ? "Pass" : "Fail");

console.log("Large Numbers Division:", divide(1e15, 1e5) === 1e10 ? "Pass" : "Fail");

**Script 2: Login Validation Tests**

This script tests a login validation function but contains logical errors and lacks test coverage.

JavaScript

// Login Validation Function

function validateLogin(username, password) {

if (!username || !password) {

return false; // Missing input check is correct

}

if (username === "admin" && password === "1234") {

return true; // Only valid login credentials are hardcoded

}

return false;

}

// Test Cases

console.log("Login Test 1:", validateLogin("admin", "1234") === true ? "Pass" : "Fail");

console.log("Login Test 2:", validateLogin("", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 3:", validateLogin("admin", "") === true ? "Pass" : "Fail"); // Incorrect expected value

// Missing Tests

// - Invalid username/password combinations

// - Edge cases like long strings or special characters

**Correction 2:**

// Login Validation Function

function validateLogin(username, password) {

if (!username || !password) {

return false; // Check for missing inputs

}

if (username === "admin" && password === "1234") {

return true; // Valid credentials

}

return false; // Invalid credentials

}

// Test Cases

console.log("Login Test 1 (Valid Credentials):", validateLogin("admin", "1234") === true ? "Pass" : "Fail");

console.log("Login Test 2 (Empty Username):", validateLogin("", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 3 (Empty Password):", validateLogin("admin", "") === false ? "Pass" : "Fail"); // Corrected expected value

console.log("Login Test 4 (Empty Both):", validateLogin("", "") === false ? "Pass" : "Fail");

console.log("Login Test 5 (Invalid Username):", validateLogin("user", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 6 (Invalid Password):", validateLogin("admin", "wrongpass") === false ? "Pass" : "Fail");

console.log("Login Test 7 (Invalid Both):", validateLogin("user", "wrongpass") === false ? "Pass" : "Fail");

// Edge Cases

console.log("Login Test 8 (Special Characters in Username):", validateLogin("ad!min", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 9 (Special Characters in Password):", validateLogin("admin", "12$#4") === false ? "Pass" : "Fail");

console.log("Login Test 10 (Long Username):", validateLogin("a".repeat(100), "1234") === false ? "Pass" : "Fail");

console.log("Login Test 11 (Long Password):", validateLogin("admin", "1".repeat(100)) === false ? "Pass" : "Fail");

// Additional Edge Case Tests

console.log("Login Test 12 (Case Sensitivity):", validateLogin("Admin", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 13 (Whitespace in Username):", validateLogin(" admin ", "1234") === false ? "Pass" : "Fail");

console.log("Login Test 14 (Whitespace in Password):", validateLogin("admin", " 1234 ") === false ? "Pass" : "Fail");

**Script 3: Shopping Cart Tests**

This script tests the functionality of a shopping cart but is inefficient and has incomplete logic.

JavaScript

// Shopping Cart Functions

let cart = [];

function addItem(item, quantity) {

if (quantity < 1) {

console.log("Error: Quantity must be at least 1");

return;

}

cart.push({ item, quantity });

}

function removeItem(item) {

cart = cart.filter(cartItem => cartItem.item !== item); // Doesn't handle case sensitivity

}

function getCartTotal() {

return cart.reduce((total, cartItem) => total + cartItem.quantity, 0); }

// Test Cases

addItem("Apple", 3);

addItem("Banana", 0); // Should throw an error but still adds to cart removeItem("apple"); // Doesn't remove "Apple" due to case sensitivity

console.log("Cart Total Test 1:", getCartTotal() === 3 ? "Pass" : "Fail"); // Fails due to bug

// Missing Tests

// - Adding duplicate items

// - Removing an item not in the cart

// - Handling empty cart scenarios

**Correction 3.**

// Shopping Cart Functions

let cart = [];

function addItem(item, quantity) {

if (quantity < 1) {

console.log("Error: Quantity must be at least 1");

return;

}

const existingItem = cart.find(cartItem => cartItem.item.toLowerCase() === item.toLowerCase());

if (existingItem) {

existingItem.quantity += quantity; // Update quantity if item already exists

} else {

cart.push({ item, quantity });

}

}

function removeItem(item) {

const initialLength = cart.length;

cart = cart.filter(cartItem => cartItem.item.toLowerCase() !== item.toLowerCase());

if (cart.length === initialLength) {

console.log(`Error: Item "${item}" not found in the cart`);

}

}

function getCartTotal() {

return cart.reduce((total, cartItem) => total + cartItem.quantity, 0);

}

// Test Cases

cart = [];

addItem("Apple", 3);

addItem("Banana", 0); // Should not add due to quantity error

addItem("apple", 2); // Should update quantity of existing item

removeItem("apple"); // Should remove all "Apple" items regardless of case

console.log("Cart Total Test 1:", getCartTotal() === 0 ? "Pass" : "Fail"); // Pass

// Additional Test Cases

cart = [];

addItem("Orange", 5);

addItem("Orange", 3); // Adding duplicate item should update quantity

console.log("Cart Total Test 2 (Duplicate Items):", getCartTotal() === 8 ? "Pass" : "Fail");

removeItem("Orange");

console.log("Cart Total Test 3 (Remove Existing Item):", getCartTotal() === 0 ? "Pass" : "Fail");

removeItem("Grapes"); // Trying to remove an item not in the cart

console.log("Cart Total Test 4 (Remove Nonexistent Item):", getCartTotal() === 0 ? "Pass" : "Fail");

cart = [];

console.log("Cart Total Test 5 (Empty Cart):", getCartTotal() === 0 ? "Pass" : "Fail");

// Edge Cases

cart = [];

addItem("Watermelon", -1); // Negative quantity should not be added

addItem("Kiwi", 0); // Zero quantity should not be added

addItem("Mango", 1);

console.log("Cart Total Test 6 (Negative and Zero Quantities):", getCartTotal() === 1 ? "Pass" : "Fail");