**Part 2 – Error Handling & Reverse Number Palindrome Problem**

**1. Error Handling with try...catch**

**Definition:**

The try...catch statement is used to handle runtime errors so your program doesn’t crash. ● **try block** → Contains code that might throw an error.

● **catch block** → Runs only if an error occurs in the try block. The error details are stored in the error object.

**Syntax:**

try

{

// code that may throw error

}

catch (error)

{

// handle error

}

**Example 1 – Division by Zero**

try

{

let result = 10 / 0;

if (!isFinite(result)) throw new Error("Cannot divide by zero!");

console.log(result);

}

catch (err) {

console.log("Error:", err.message);

}

**Example 2 – Invalid JSON**

try

{

let data = JSON.parse("invalid json");

}

catch (err)

{

console.log("JSON Parse Error:", err.message); }

**Example 3 – Manual Error Throwing**

function checkAge(age)

{

if (age < 18) throw new Error("You must be 18+"); return "Access granted";

}

try

{

console.log(checkAge(16));

}

catch (err)

{

console.log(err.message);

}

**Example 4 – Using finally**

try {

let fileOpen = true;

console.log("Opening file...");

// Simulate an error

throw new Error("File read error!");

console.log("Reading file..."); // this line will not run }

catch (err)

{

console.log("Error:", err.message);

}

finally

{

console.log("Closing file (cleanup)...");

}

**2. Reverse Number Function & Palindrome Check Logic:**

● Convert the number to string.

● Reverse the string.

● Convert back to number.

● Compare with original.

**Implementation:**

function reverseNumber(num)

{

if (typeof num !== "number" || isNaN(num))

{

throw new Error("Invalid number input");

}

return parseInt(num.toString().split("").reverse().join("")); }

function isPalindrome(num) {

return num === reverseNumber(num);

}

try {

let num = 1221;

console.log("Reversed:", reverseNumber(num)); console.log("Is Palindrome:", isPalindrome(num)); } catch (err) {

console.log("Error:", err.message);

}

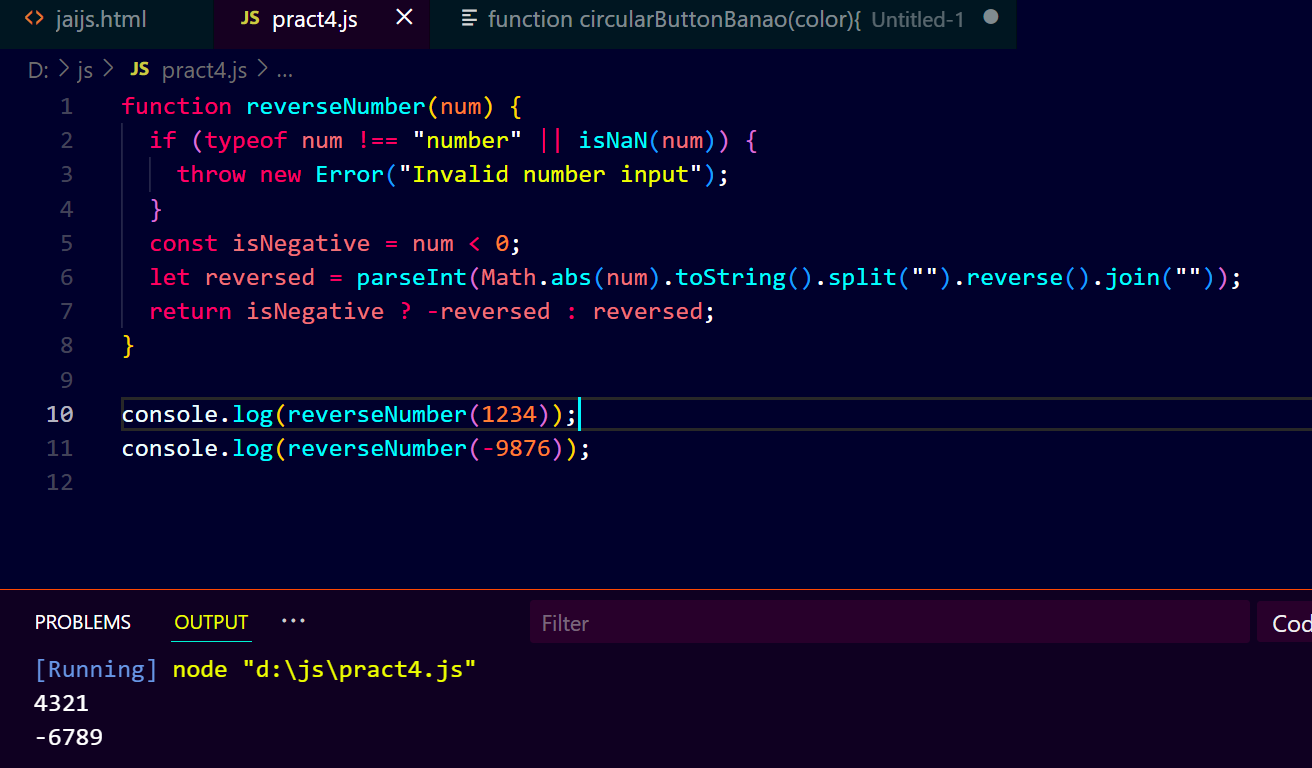
**Examples:**

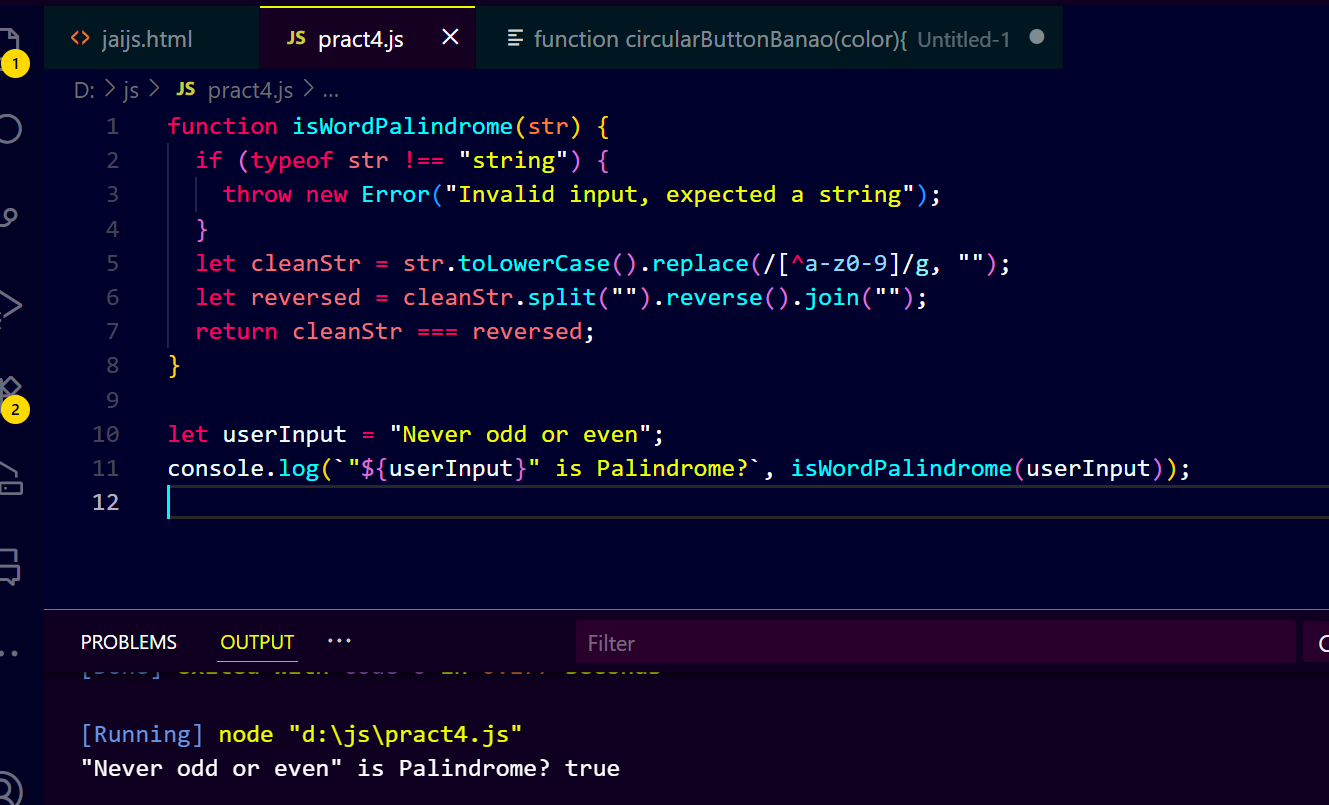
console.log(isPalindrome(12321)); // true

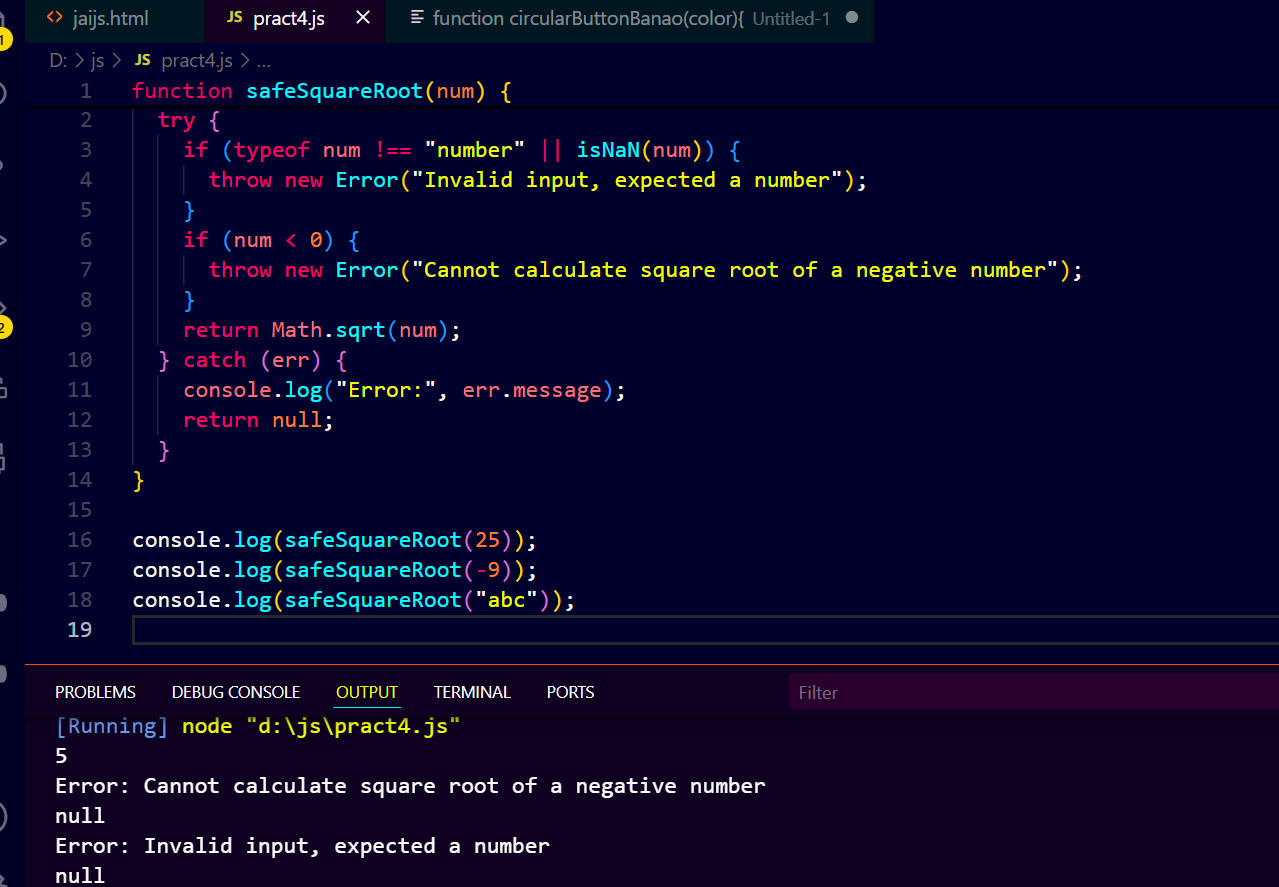
console.log(isPalindrome(45654)); // true

console.log(isPalindrome(12345)); // false

**Student Tasks for Part 2**

1. Modify reverseNumber to work for negative numbers as well. 

2. Write a program that takes user input (string) and checks if it’s a palindrome (word-based, not number). 

3. Use try...catch to handle invalid inputs for a square root function. 4. Create a function using arrow syntax that checks if a given number is prime, and handle invalid inputs with try...catch.