Experiment – 1 a: TypeScript

Name of Student	**Rohan Lalchandani**
Class Roll No	**D15A_25**
D.O.P.	**23/01/2025**
D.O.S.	**30/01/2025**
Sign and Grade	

AIM : Write a simple TypeScript program using basic data types (number, string, boolean) and operators.

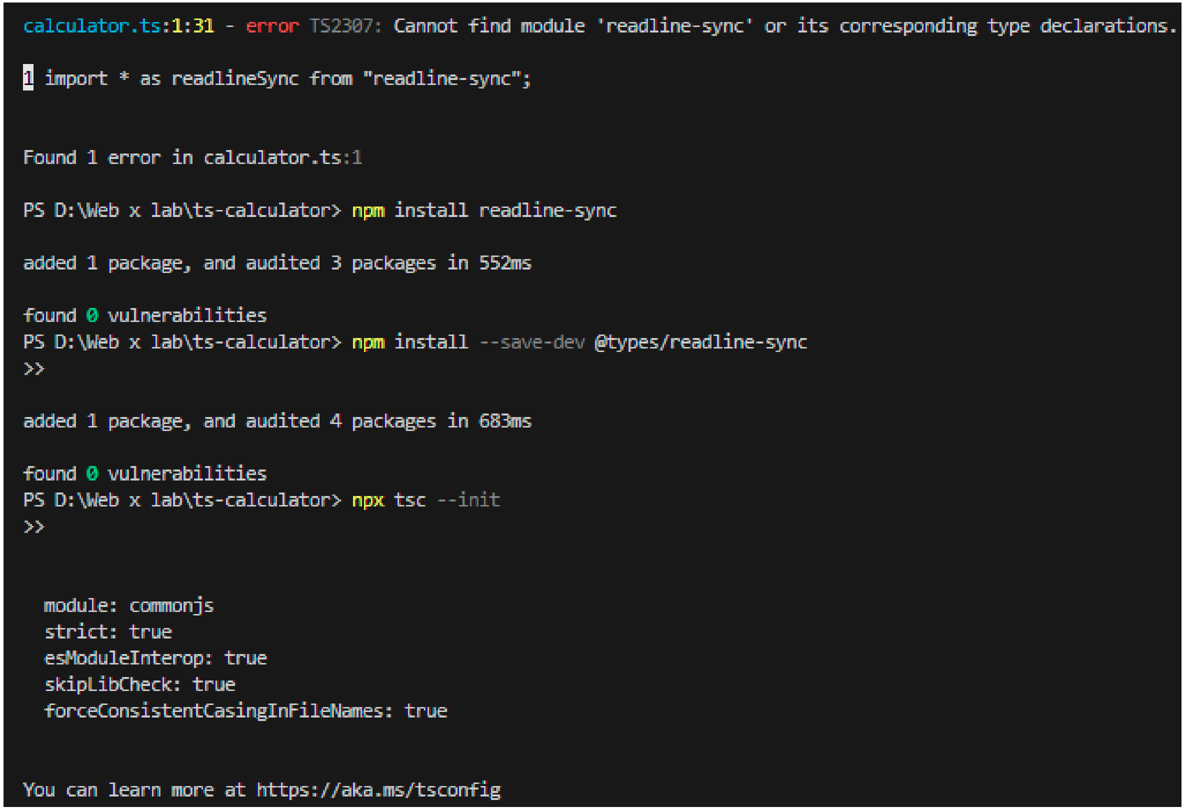
OVERVIEW OF TASKS PERFORMED :

The experiment involves implementing a **calculator** in TypeScript that performs basic arithmetic operations (addition, subtraction, multiplication, and division) while handling invalid operations and division by zero gracefully. Additionally, a **Student Result Database Management System** was designed using TypeScript, where student details and subject marks were stored as variables. The total and average marks were calculated, and a condition was applied to determine whether the student passed or failed. The results were then displayed in the console.

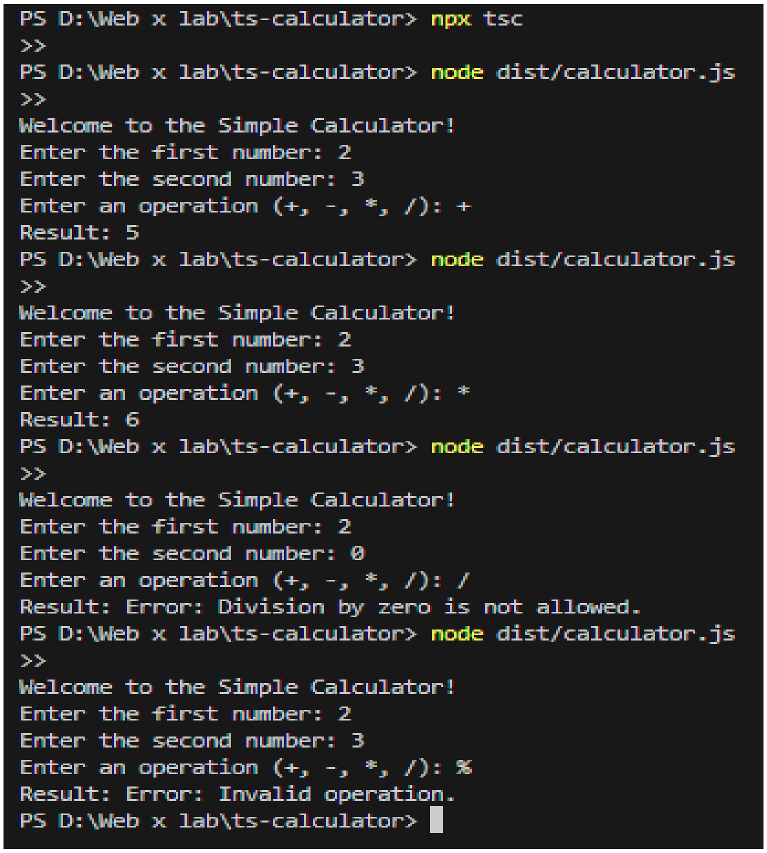
GITHUB LINK -

https://github.com/Rohan-Lalchandani08/WebX_Lab/tree/main/WebX_Exp1aOUTPUT

(a) TypeScript Calculator



```
calculator.ts:1:31 - error TS2307: Cannot find module 'readline-sync' or its corresponding type declarations.

1 import * as readlineSync from "readline-sync";


Found 1 error in calculator.ts:1

PS D:\Web x lab\ts-calculator> npm install readline-sync

added 1 package, and audited 3 packages in 552ms

found 0 vulnerabilities
PS D:\Web x lab\ts-calculator> npm install --save-dev @types/readline-sync
>>

added 1 package, and audited 4 packages in 683ms

found 0 vulnerabilities
PS D:\Web x lab\ts-calculator> npx tsc --init
>>


  module: commonjs
  strict: true
  esModuleInterop: true
  skipLibCheck: true
  forceConsistentCasingInFileNames: true


You can learn more at https://aka.ms/tsconfig
```

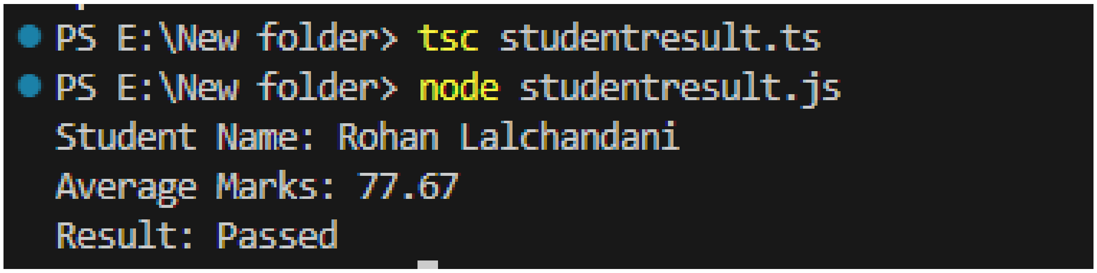


```
PS D:\Web x lab\ts-calculator> npx tsc
>>
PS D:\Web x lab\ts-calculator> node dist/calculator.js
>>
Welcome to the Simple Calculator!
Enter the first number: 2
Enter the second number: 3
Enter an operation (+, -, *, /): +
Result: 5
PS D:\Web x lab\ts-calculator> node dist/calculator.js
>>
Welcome to the Simple Calculator!
Enter the first number: 2
Enter the second number: 3
Enter an operation (+, -, *, /): *
Result: 6
PS D:\Web x lab\ts-calculator> node dist/calculator.js
>>
Welcome to the Simple Calculator!
Enter the first number: 2
Enter the second number: 0
Enter an operation (+, -, *, /): /
Result: Error: Division by zero is not allowed.
PS D:\Web x lab\ts-calculator> node dist/calculator.js
>>
Welcome to the Simple Calculator!
Enter the first number: 2
Enter the second number: 3
Enter an operation (+, -, *, /): %
Result: Error: Invalid operation.
PS D:\Web x lab\ts-calculator>
```

This screenshot showcases the output of the **TypeScript Calculator**, which performs basic arithmetic operations such as addition, subtraction, multiplication, and division. The console displays:

- The results of valid operations (`add`, `subtract`, `multiply`, and `divide`).
- An error message when attempting **division by zero**.
- An error message for an **invalid operation** (e.g., `modulus`).

(b) Student Result Database Management System



```
● PS E:\New folder> tsc studentresult.ts
● PS E:\New folder> node studentresult.js
  Student Name: Rohan Lalchandani
  Average Marks: 77.67
  Result: Passed
```

This screenshot displays the output of the **Student Result Database Management System**, which calculates and prints:

- The **student's name**.
- The **average marks** (formatted to two decimal places).
- The **final result** (either "Passed" or "Failed" based on a 40% passing criteria).

CONCLUSION

This experiment successfully showcased the development of a **calculator** and a **student result management system** using TypeScript.

The **calculator** efficiently performs arithmetic operations while ensuring proper error handling, such as managing invalid inputs and preventing division by zero.

The **student result management system** effectively organizes student data, computes total and average marks, and determines pass/fail status using object-oriented programming principles.