

Creating a calculator in typescript using Inquirer

Name : Abdul Nasir

Batch No : 51 (Q1)

Create a calculator in TypeScript by installing Inquirer for user input. Set up the TypeScript configuration file (tsconfig.json) for compilation. With Inquirer, prompt the user for numbers and operations to perform basic calculations.

Type Script Installation:

Type below command in Command Prompt and press Enter Key.

npm i typescript -g

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm i typescript -g

changed 1 package in 34s
```

To make Package.json file current working directory, type below command and press Enter Key.

npm init -y

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm i typescript -g

changed 1 package in 34s

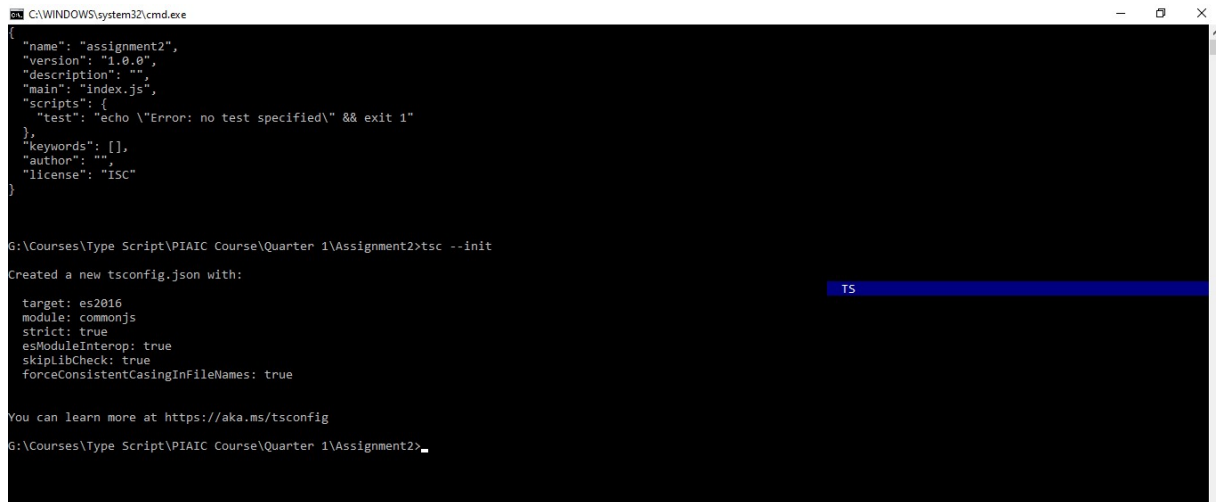
G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm init -y
Wrote to G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2\package.json:

{
  "name": "assignment2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>
```

To make config file current working directory, type below command and press Enter Key.

tsc --init



```
C:\WINDOWS\system32\cmd.exe
{
  "name": "assignment2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\Error: no test specified\\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>tsc --init
Created a new tsconfig.json with:

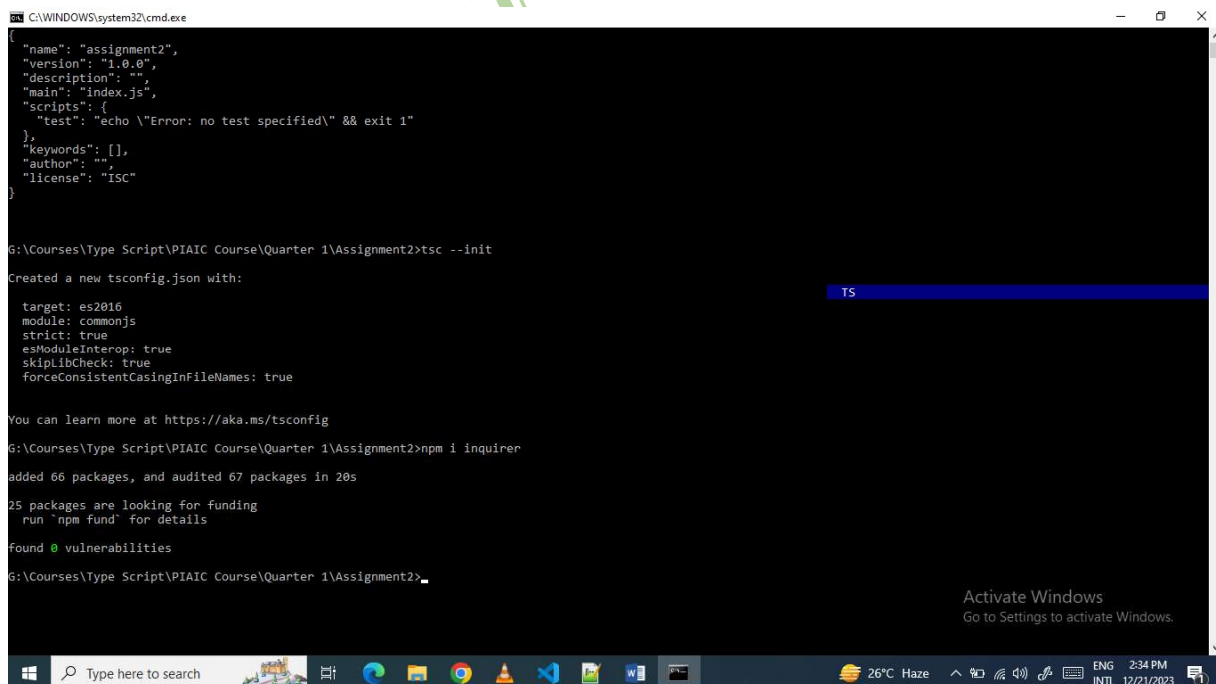
  target: es2016
  module: commonjs
  strict: true
  esModuleInterop: true
  skipLibCheck: true
  forceConsistentCasingInFileNames: true

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

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>
```

To download inquirer in current working directory, type below command and press Enter Key.

npm i inquirer



```
C:\WINDOWS\system32\cmd.exe
{
  "name": "assignment2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\Error: no test specified\\ && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>tsc --init
Created a new tsconfig.json with:

  target: es2016
  module: commonjs
  strict: true
  esModuleInterop: true
  skipLibCheck: true
  forceConsistentCasingInFileNames: true

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

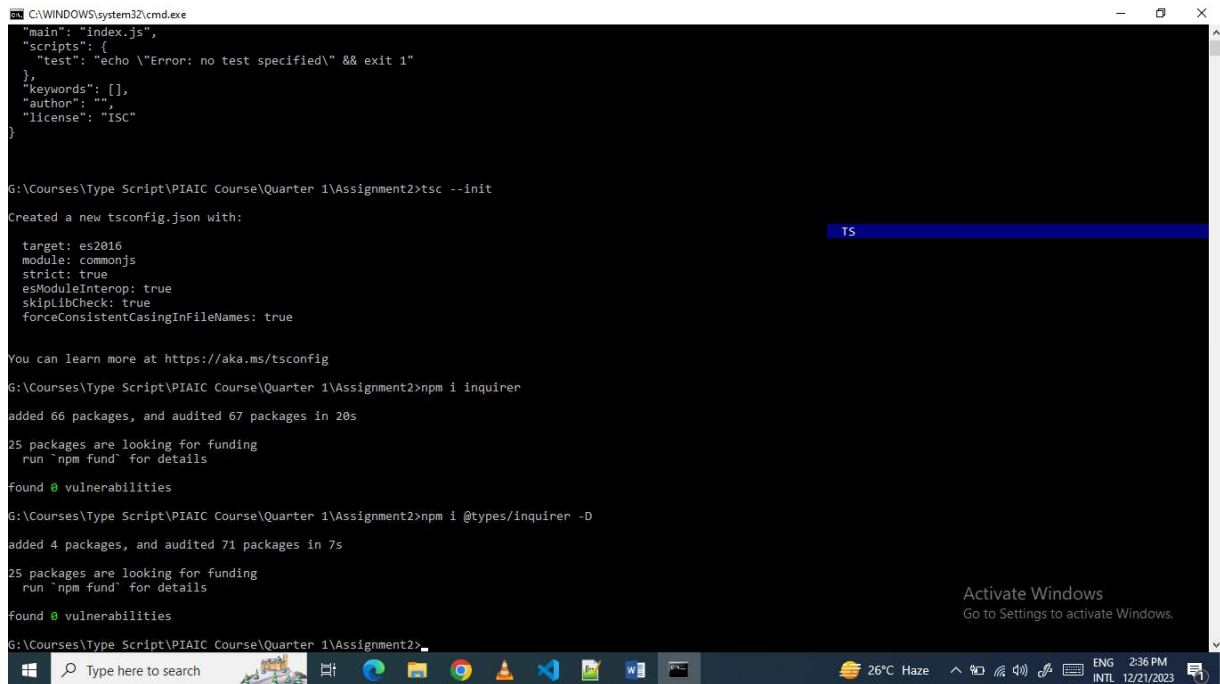
G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm i inquirer
added 66 packages, and audited 67 packages in 20s
25 packages are looking for funding
  run `npm fund` for details
Found 0 vulnerabilities

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>
```

Activate Windows
Go to Settings to activate Windows.

To download types inquirer in current working directory, type below command and press Enter Key.

npm i @types/inquirer -D



```
C:\WINDOWS\system32\cmd.exe
"main": "index.js",
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>tsc --init
Created a new tsconfig.json with:
  target: es2016
  module: commonjs
  strict: true
  esModuleInterop: true
  skipLibCheck: true
  forceConsistentCasingInFileNames: true
You can learn more at https://aka.ms/tsconfig

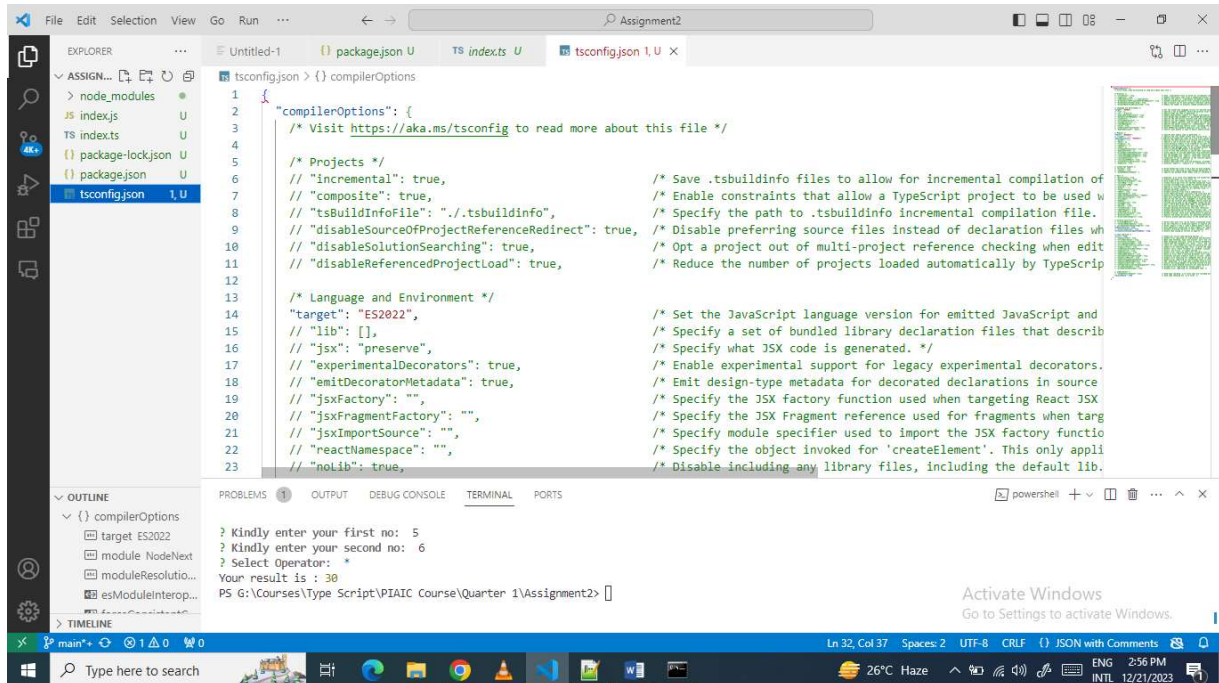
G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm i inquirer
added 66 packages, and audited 67 packages in 20s
25 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>npm i @types/inquirer -D
added 4 packages, and audited 71 packages in 7s
25 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities

G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2>
```

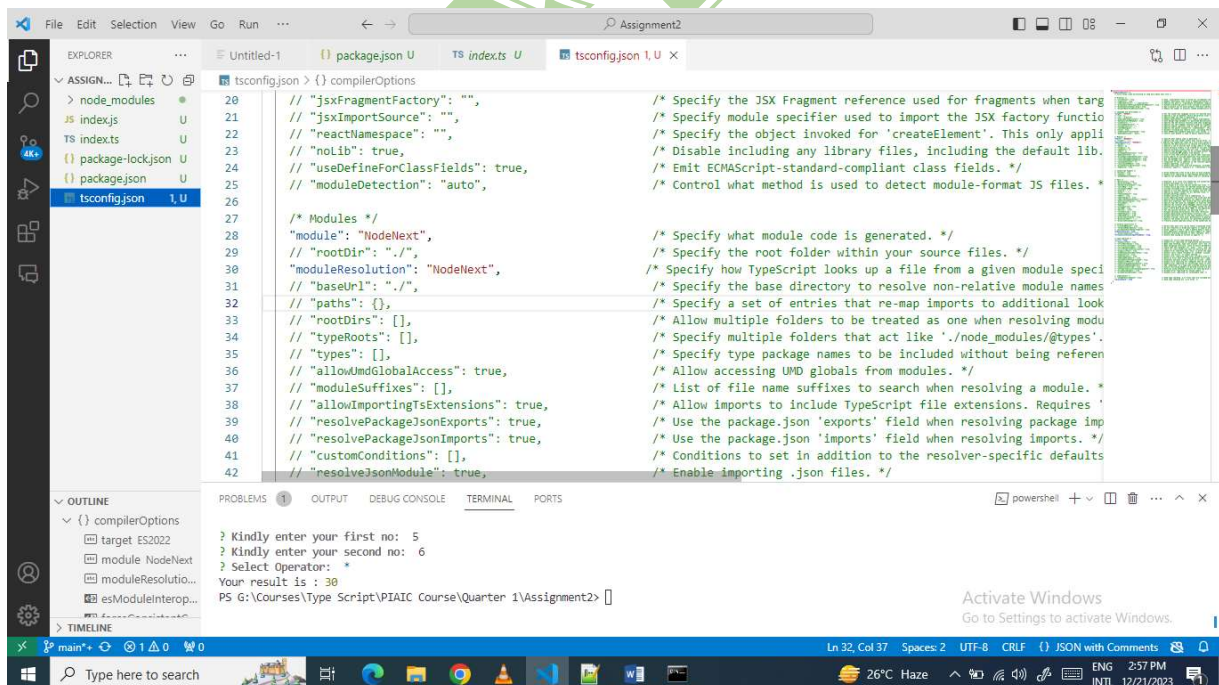
Activate Windows
Go to Settings to activate Windows.

Now make some changes in config file



```
1  "compilerOptions": {
2      /* Visit https://aka.ms/tsconfig to read more about this file */
3
4      /* Projects */
5      // "incremental": true,           /* Save .tsbuildinfo files to allow for incremental compilation of
6      // "composite": true,             /* Enables constraints that allow a TypeScript project to be used w
7      // "tsBuildInfoFile": "./.tsbuildinfo", /* Specify the path to .tsbuildinfo incremental compilation file.
8      // "disableSourceOfProjectReferenceRedirect": true, /* Disable preferring source files instead of declaration files wh
9      // "disableSolutionSearching": true, /* Opt a project out of multi-project reference checking when edit
10     // "disableReferencedProjectLoad": true, /* Reduce the number of projects loaded automatically by TypeScript
11
12     /* Language and Environment */
13     // "target": "ES2022",           /* Set the JavaScript language version for emitted JavaScript and
14     // "lib": [],                   /* Specify a set of bundled library declaration files that describ
15     // "jsx": "preserve",           /* Specify what JSX code is generated. */
16     // "experimentalDecorators": true, /* Enable experimental support for legacy experimental decorators.
17     // "emitDecoratorMetadata": true, /* Emit design-type metadata for decorated declarations in source
18     // "jsxFactory": "",            /* Specify the JSX factory function used when targeting React JSX
19     // "jsxFragmentFactory": "",    /* Specify the JSX Fragment reference used for fragments when targ
20     // "jsxImportSource": "",       /* Specify module specifier used to import the JSX factory functio
21     // "reactNamespace": "",       /* Specify the object invoked for 'createElement'. This only appli
22     // "noLib": true,              /* Disable including any library files, including the default lib.
23
```

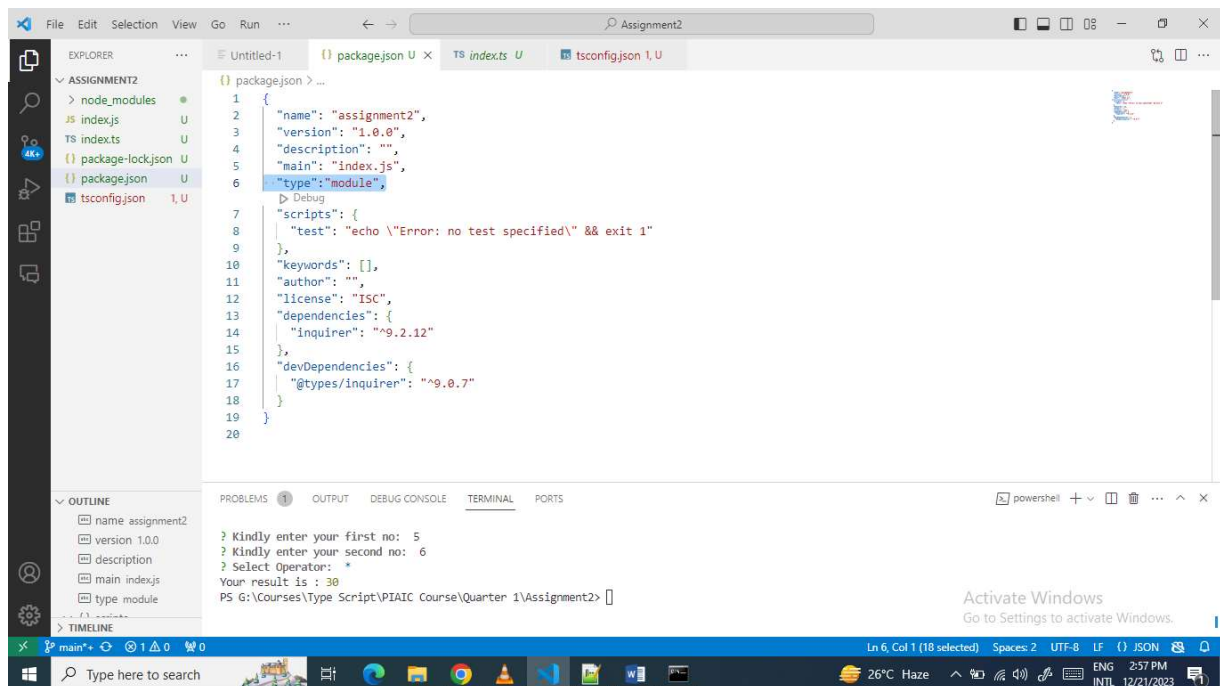
Activate Windows
Go to Settings to activate Windows.



```
20 // "jsxFragmentFactory": "", /* Specify the JSX Fragment reference used for fragments when targ
21 // "jsxImportSource": "",    /* Specify module specifier used to import the JSX factory functio
22 // "reactNamespace": "",    /* Specify the object invoked for 'createElement'. This only appli
23 // "noLib": true,          /* Disable including any library files, including the default lib.
24 // "useDefineForClassFields": true, /* Emit ECMAScript-standard-compliant class fields. */
25 // "moduleDetection": "auto", /* Control what method is used to detect module-format JS files. */
26
27 /* Modules */
28 "module": "NodeNext", /* Specify what module code is generated. */
29 // "rootDir": "./", /* Specify the root folder within your source files. */
30 "moduleResolution": "NodeNext", /* Specify how TypeScript looks up a file from a given module speci
31 // "baseUrl": "./", /* Specify the base directory to resolve non-relative module names
32 // "paths": {}, /* Specify a set of entries that re-map imports to additional look
33 // "rootDirs": [], /* Allow multiple folders to be treated as one when resolving modu
34 // "typeRoots": [], /* Specify multiple folders that act like './node_modules/@types'.
35 // "types": [], /* Specify type package names to be included without being referen
36 // "allowUmdGlobalAccess": true, /* Allow accessing UMD globals from modules. */
37 // "moduleSuffixes": [], /* List of file name suffixes to search when resolving a module. *
38 // "allowImportingTsExtensions": true, /* Allow imports to include TypeScript file extensions. Requires '
39 // "resolvePackageJsonExports": true, /* Use the package.json 'exports' field when resolving package imp
40 // "resolvePackageJsonImports": true, /* Use the package.json 'imports' field when resolving imports. */
41 // "customConditions": [], /* Conditions to set in addition to the resolver-specific defaults
42 // "resolveJsonModule": true, /* Enable importing .json files. */
```

Activate Windows
Go to Settings to activate Windows.

Open Package.json file and update as below:

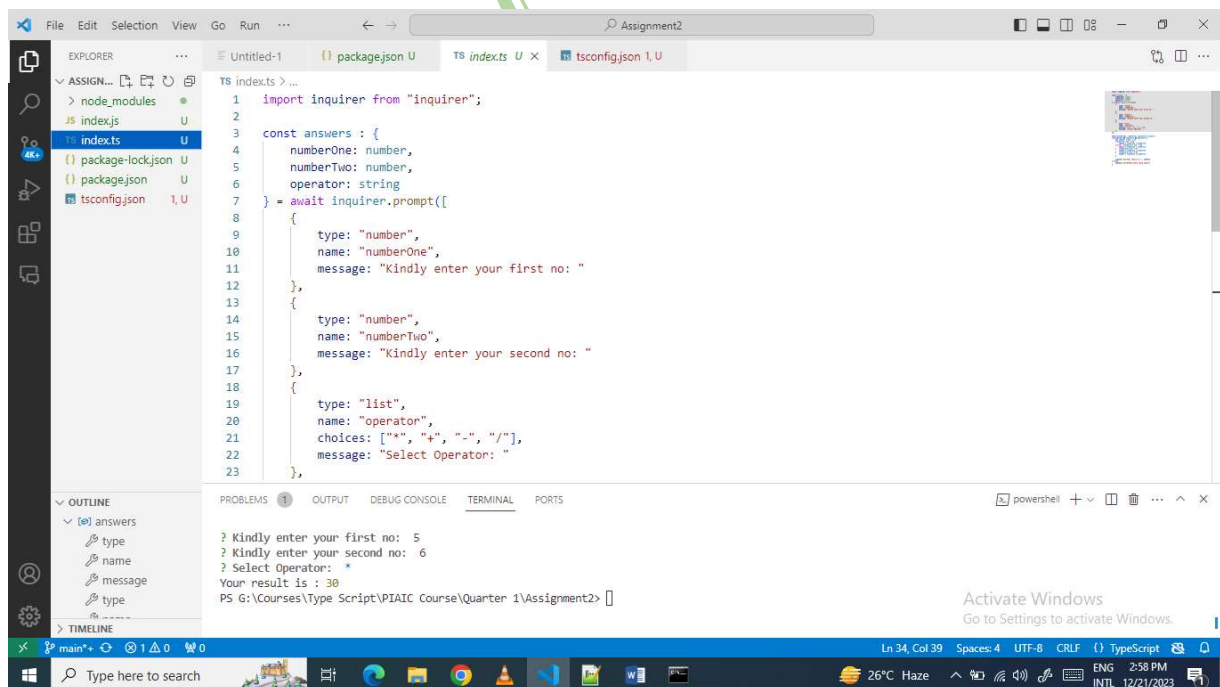


The screenshot shows the Visual Studio Code editor with the 'package.json' file open. The file contains the following JSON structure:

```
{
  "name": "assignment2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "inquirer": "^9.2.12"
  },
  "devDependencies": {
    "@types/inquirer": "^9.0.7"
  }
}
```

The Explorer sidebar on the left shows the project structure with files like 'index.js', 'package-lock.json', 'package.json', and 'tsconfig.json'. The Output window at the bottom shows the command prompt output: 'Kindly enter your first no: 5', 'Kindly enter your second no: 6', 'Select Operator: *', and 'Your result is : 30'.

Now create a file as .ts for example index.ts and write code as per your requirement below code is for calculator .



The screenshot shows the Visual Studio Code editor with the 'index.ts' file open. The file contains the following TypeScript code:

```
import inquirer from "inquirer";

const answers : {
  numberOne: number,
  numberTwo: number,
  operator: string
} = await inquirer.prompt([
  {
    type: "number",
    name: "numberOne",
    message: "Kindly enter your first no: "
  },
  {
    type: "number",
    name: "numberTwo",
    message: "Kindly enter your second no: "
  },
  {
    type: "list",
    name: "operator",
    choices: ["*", "+", "-", "/"],
    message: "Select Operator: "
  }
]);
```

The Explorer sidebar on the left shows the project structure with files like 'index.ts', 'package-lock.json', 'package.json', and 'tsconfig.json'. The Output window at the bottom shows the command prompt output: 'Kindly enter your first no: 5', 'Kindly enter your second no: 6', 'Select Operator: *', and 'Your result is : 30'.

```

import inquirer from "inquirer";

const answers : {
  numberOne: number,
  numberTwo: number,
  operator: string
} = await inquirer.prompt([
  {
    type: "number",
    name: "numberOne",
    message: "Kindly enter your first no: "
  },
  {
    type: "number",
    name: "numberTwo",
    message: "Kindly enter your second no: "
  },
  {
    type: "list",
    name: "operator",
    choices: ["*", "+", "-", "/"],
    message: "Select Operator: "
  }
]);

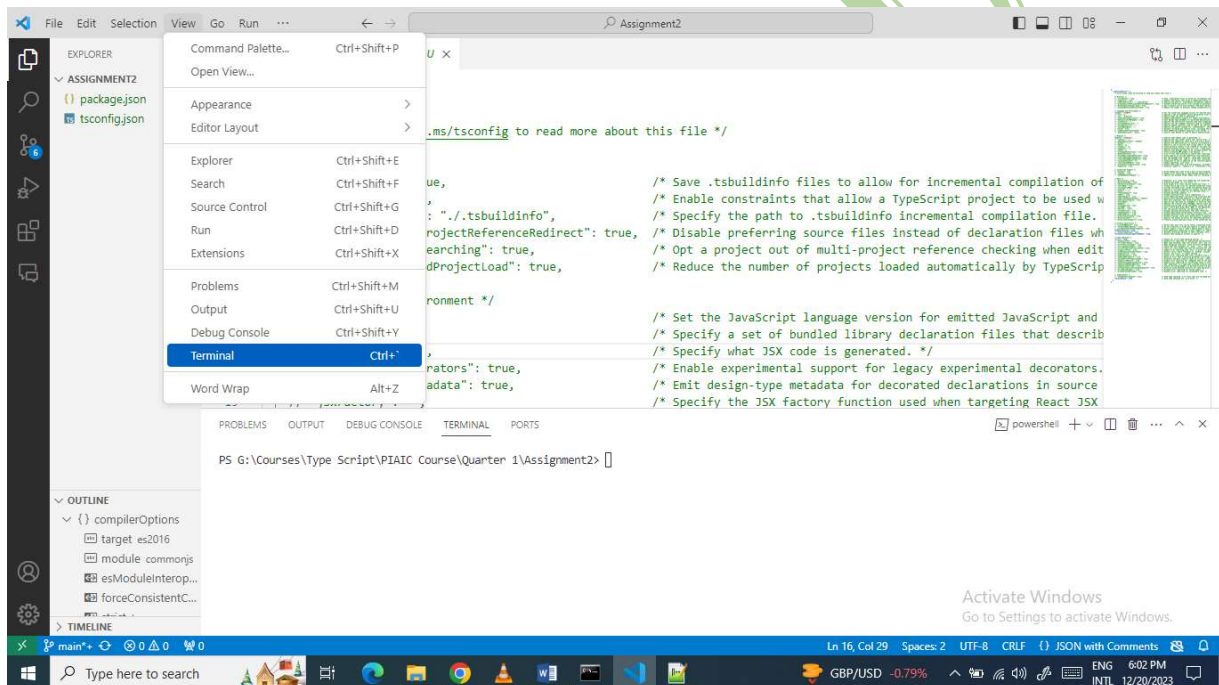
const {numberOne, numberTwo, operator} = answers;
if(numberOne && numberTwo && operator) {
  let result: number = 0;
  if(operator === "+"){
    result = numberOne + numberTwo
  } else if(operator === "-"){
    result = numberOne - numberTwo
  } if(operator === "/"){
    result = numberOne / numberTwo
  } if(operator === "*"){
    result = numberOne * numberTwo
  }
}

```



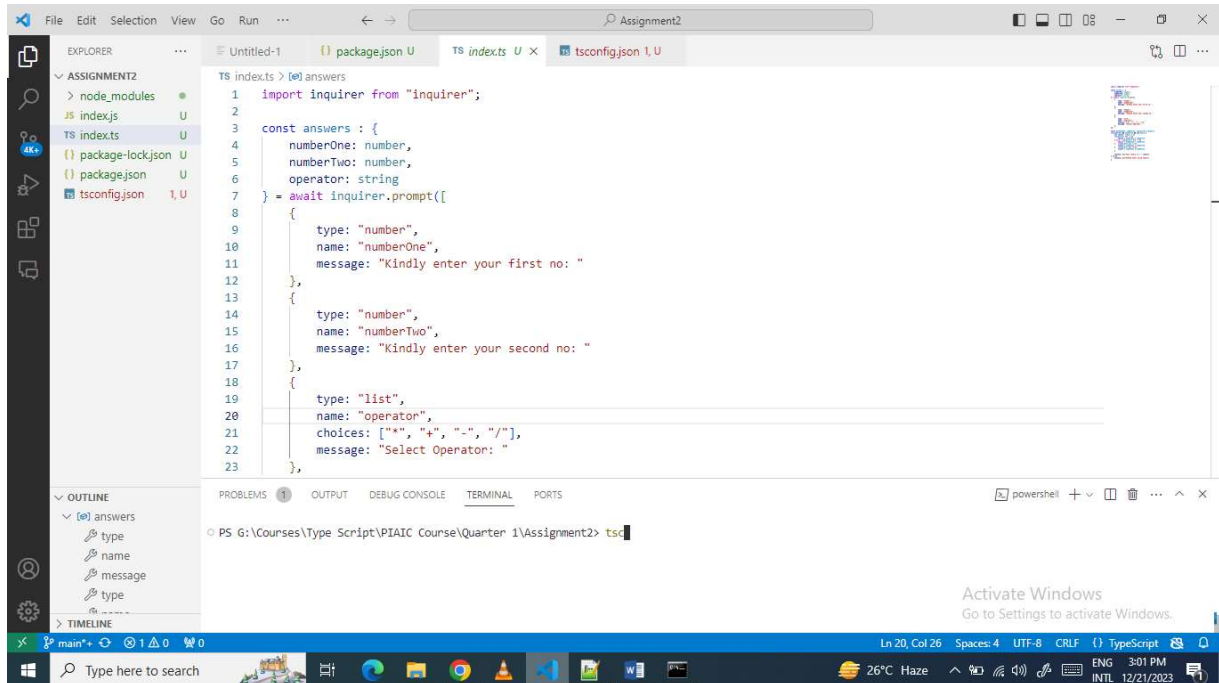
```
console.log("Your result is :", result)
} else{
console.log("Kindly enter valid input")
}
```

Open terminal window in VS Code , go to View menu and click on Terminal



Write command as below to create .js file it will create the .js file.

tsc



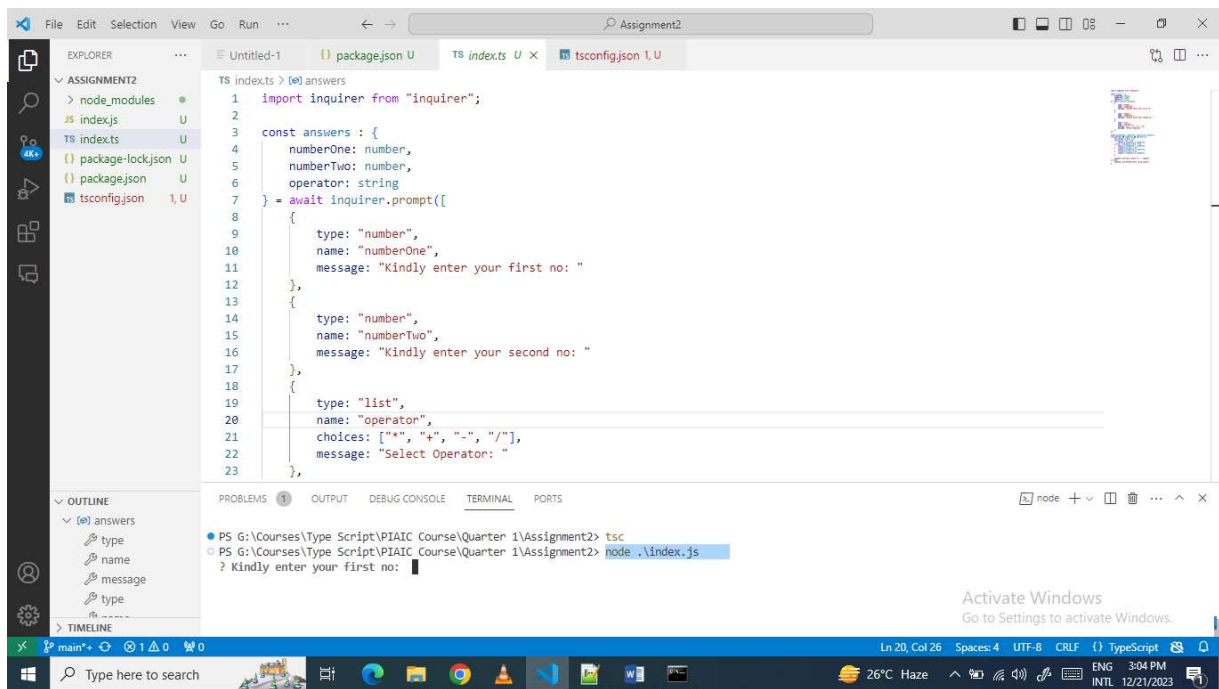
The screenshot shows the Visual Studio Code (VS Code) editor interface. The Explorer sidebar on the left displays the project structure for 'ASSIGNMENT2', including files like 'node_modules', 'index.js', 'package-lock.json', 'package.json', and 'tsconfig.json'. The main editor area shows a TypeScript file named 'answers.ts' with the following code:

```
1 import inquirer from "inquirer";
2
3 const answers : {
4     numberOne: number,
5     numberTwo: number,
6     operator: string
7 } = await inquirer.prompt([
8     {
9         type: "number",
10        name: "numberOne",
11        message: "Kindly enter your first no: "
12    },
13    {
14        type: "number",
15        name: "numberTwo",
16        message: "Kindly enter your second no: "
17    },
18    {
19        type: "list",
20        name: "operator",
21        choices: ["*", "+", "-", "/"],
22        message: "Select Operator: "
23    }
24 ])
```

Below the editor, the TERMINAL panel is open, showing the command prompt at the path 'PS G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2> tsc'. The status bar at the bottom indicates the current file is 'main.ts' at line 20, column 26, with a UTF-8 encoding and CRLF line endings.

Now type below command and press enter key, and index.js file will be run

node .\index.js



The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows a project named 'ASSIGNMENT2' with files: 'node_modules', 'index.js', 'package-lock.json', 'package.json', and 'tsconfig.json'. The main editor area displays a TypeScript file named 'index.ts' with the following code:

```
1 import inquirer from "inquirer";
2
3
4 const answers : {
5     numberOne: number,
6     numberTwo: number,
7     operator: string
8 } = await inquirer.prompt([
9     {
10         type: "number",
11         name: "numberOne",
12         message: "Kindly enter your first no: "
13     },
14     {
15         type: "number",
16         name: "numberTwo",
17         message: "Kindly enter your second no: "
18     },
19     {
20         type: "list",
21         name: "operator",
22         choices: ["*", "+", "-", "/"],
23         message: "Select Operator: "
```

The OUTPUT panel at the bottom shows the command 'node .\index.js' being executed, followed by the prompt '? Kindly enter your first no: '.

Now enter two number one by one and with down arrow select the operator for calculation and press enter key, it will show the result.

```
1 import inquirer from "inquirer";
2
3 const answers : {
4     numberOne: number,
5     numberTwo: number,
6     operator: string
7 } = await inquirer.prompt([
8     {
9         type: "number",
10        name: "numberOne",
11        message: "Kindly enter your first no: "
12    },
13    {
14        type: "number",
15        name: "numberTwo",
16        message: "Kindly enter your second no: "
17    },
18    {
19        type: "list",
20        name: "operator",
21        choices: ["*", "+", "-", "/"],
22        message: "Select Operator: "
23    }
24 ])
```

? Kindly enter your first no: 5
? Kindly enter your second no: 6
? Select Operator: *
? *

```
1 import inquirer from "inquirer";
2
3 const answers : {
4     numberOne: number,
5     numberTwo: number,
6     operator: string
7 } = await inquirer.prompt([
8     {
9         type: "number",
10        name: "numberOne",
11        message: "Kindly enter your first no: "
12    },
13    {
14        type: "number",
15        name: "numberTwo",
16        message: "Kindly enter your second no: "
17    },
18    {
19        type: "list",
20        name: "operator",
21        choices: ["*", "+", "-", "/"],
22        message: "Select Operator: "
23    }
24 ])
```

? Kindly enter your first no: 5
? Kindly enter your second no: 6
? Select Operator: *
Your result is : 30
PS G:\Courses\Type Script\PIAIC Course\Quarter 1\Assignment2

Thanks