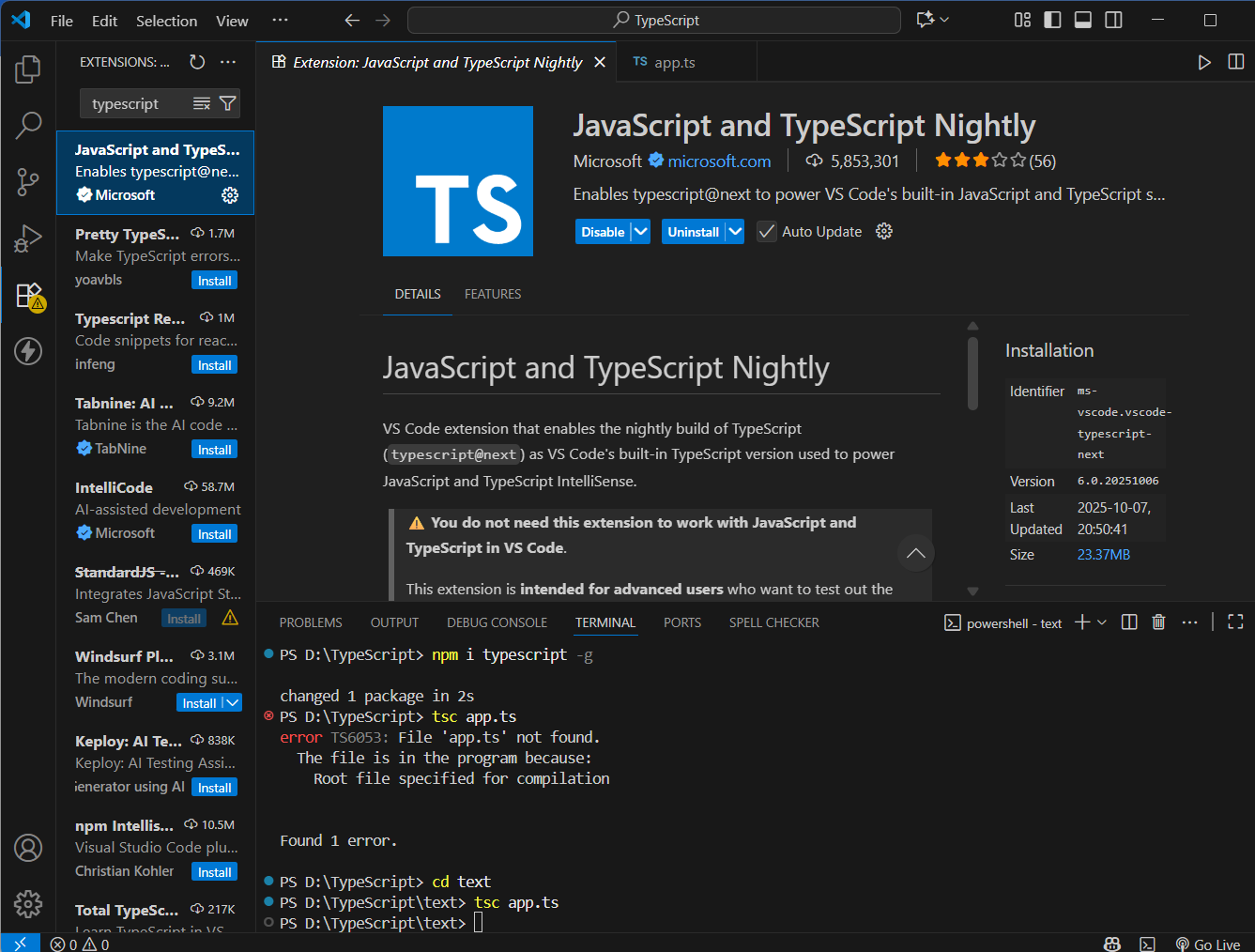
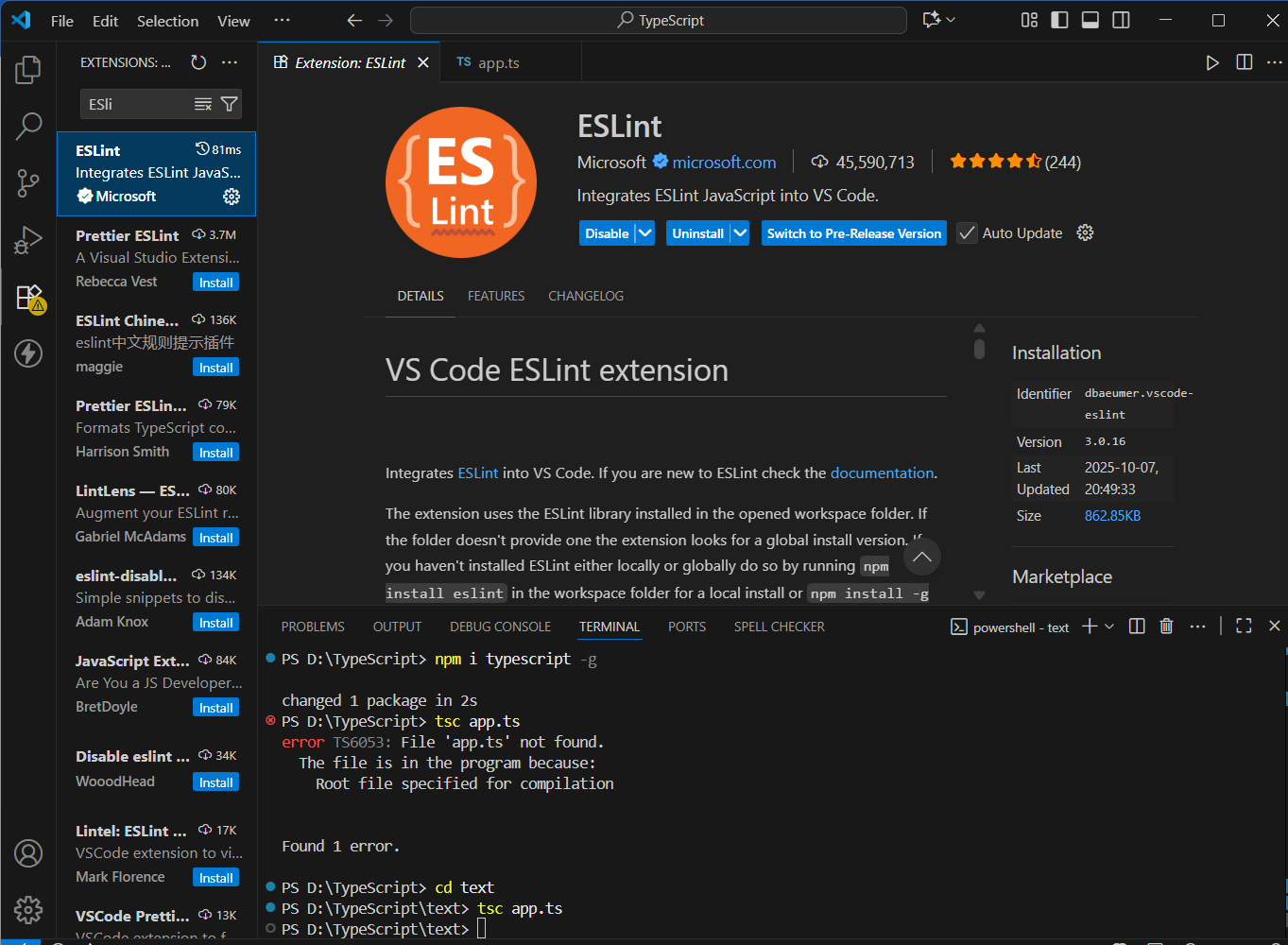
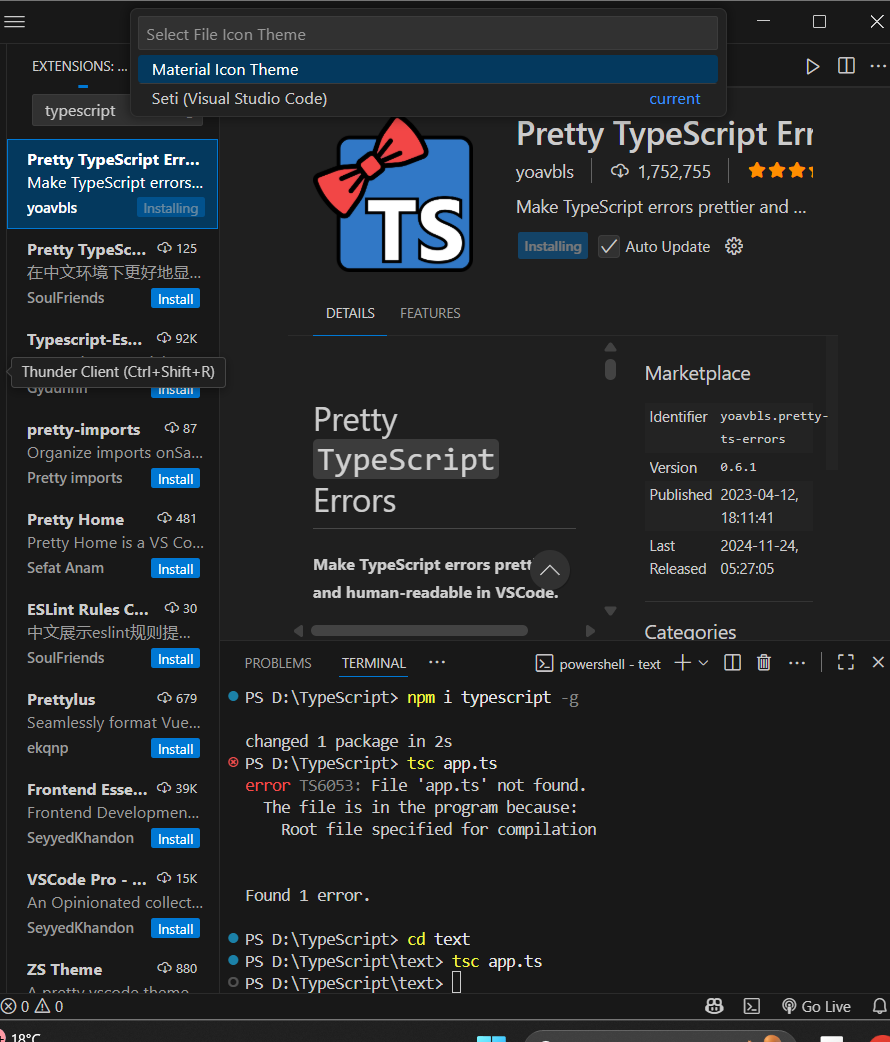
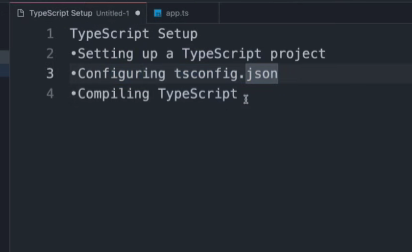
* npm i typescript -g //package installed globally
* installs he compiler of typescript
* ESlint help us to find error during development

Extension

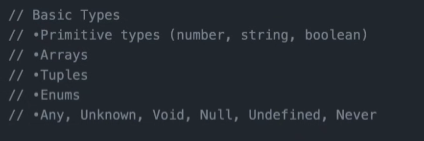








1. npm i typescript -g //installed the compiler of typescript that will convert ts to js
2. tsc -- init// configuring tsconfigu.json whioch will help us set up on hwere it sgould throw an error
3. tsc filename and tsc --watch so that code compiler as we write



**1. Definition**

The **map()** function is an **array method** in JavaScript that creates a **new array** by applying a given **function to each element** of the original array.

It **does not modify** the original array — it returns a **new one**.

array.map(function(currentValue, index, arr), thisValue)

**1. Definition**

The **filter()** method in JavaScript is used to **create a new array** that includes only the **elements that pass a specific test (condition)** defined by a callback function.

It does **not modify the original array** — instead, it returns a **new filtered array**.

**2. Syntax**

array.filter(function(currentValue, index, arr), thisValue)

**Parameters**

* currentValue → The current element being processed.
* index *(optional)* → Index of the current element.
* arr *(optional)* → The array that filter() was called upon.
* thisValue *(optional)* → Value to use as this inside the callback.

**3. Example 1: Filtering Numbers**

const numbers = [10, 25, 30, 45, 50];

const above30 = numbers.filter(num => num > 30);

console.log(above30); // Output: [4

**1. Definition**

The **reduce()** method executes a **callback function** on each element of an array to **reduce it to a single value** — such as a **sum, product, object, or string**.

It processes elements from **left to right** and **accumulates** the result.

**2. Syntax**

array.reduce(function(accumulator, currentValue, index