

19AD602

FileHomeShareView

←→↕↑

This PC > aid-commonarea (\\172.16.16.220) (X:) > Jenefa > Subjects > 19AD602 >

↻

Search 19AD602

★ Quick access

Desktop

Downloads

Documents

Pictures

19AD602

DVT

Recommender s

unit 4

OneDrive - Person

This PC

3D Objects

Desktop

Documents

Downloads

Music

Pictures

Videos

Local Disk (C:)

aid-commonare

aid-commonare

aid-commonare

Network

AICTE-STREAM

Name	Date modified	Type	Size
JS fun	08-02-2023 11:11	File folder	
19AD502 - ISD 2022 v1	25-08-2022 16:17	Microsoft Word D...	1,567 KB
19AD602 - FULL STACK DEVELOPMENT	15-03-2023 11:10	Microsoft Word D...	23 KB
19AD602 - ISD 2023	16-03-2023 17:05	Microsoft Word D...	2,258 KB
19AD652 - FULL STACK DEVELOPMENT L...	02-01-2023 14:32	Microsoft Word D...	21 KB
AID-UG-5-S1-CAT1-19AD502-Set1	25-08-2022 11:03	Microsoft Word 9...	217 KB
AID-UG-5-S1-CAT1-19AD502-Set2	25-08-2022 11:03	Microsoft Word 9...	914 KB
AID-UG-6-S1-CAT2-19AD602-Set1	08-03-2023 18:29	Microsoft Word 9...	91 KB
AID-UG-6-S1-CAT2-19AD602-Set2	27-03-2023 17:47	Microsoft Word 9...	93 KB
AID-UG-6-S1-CAT2-19AD602-Set2IMprov...	27-03-2023 17:49	Microsoft Word 9...	93 KB
assign1	28-03-2023 09:45	Microsoft Word D...	2,009 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
Frank Zammetti - Modern Full-Stack Dev...	09-12-2021 11:36	Adobe Acrobat D...	5,778 KB
host objects	25-02-2021 14:55	Microsoft PowerP...	2,747 KB
HTML Element Reference	06-01-2023 18:03	Microsoft Word D...	36 KB
JavaScript Variables	07-02-2023 09:42	Microsoft Word D...	16 KB
JavaScriptFunctions	23-12-2009 14:26	Microsoft PowerP...	670 KB
JavaScriptObjects	28-12-2009 17:08	Microsoft PowerP...	605 KB
Learning JavaScript	01-02-2023 18:07	Microsoft PowerP...	1,803 KB
NodeJS	28-02-2023 17:52	Microsoft Word D...	25 KB
nodejs_at_a_glance v1	09-03-2023 14:55	Microsoft PowerP...	1,270 KB
nodejs_at_a_glance	09-03-2023 11:59	Microsoft PowerP...	1,270 KB
React JS	27-03-2023 15:18	Microsoft Power...	
ReactJS	27-02-2023 17:51	Adobe Acrobat	
reactjs_tutorial	27-02-2023 18:00	Adobe Acrobat	
React-MIT	27-02-2023 17:59	Adobe Acrobat	
react-slides	27-02-2023 17:59	Adobe Acrobat	

38 items

Screen Capture

i

Impartus Screen Capture

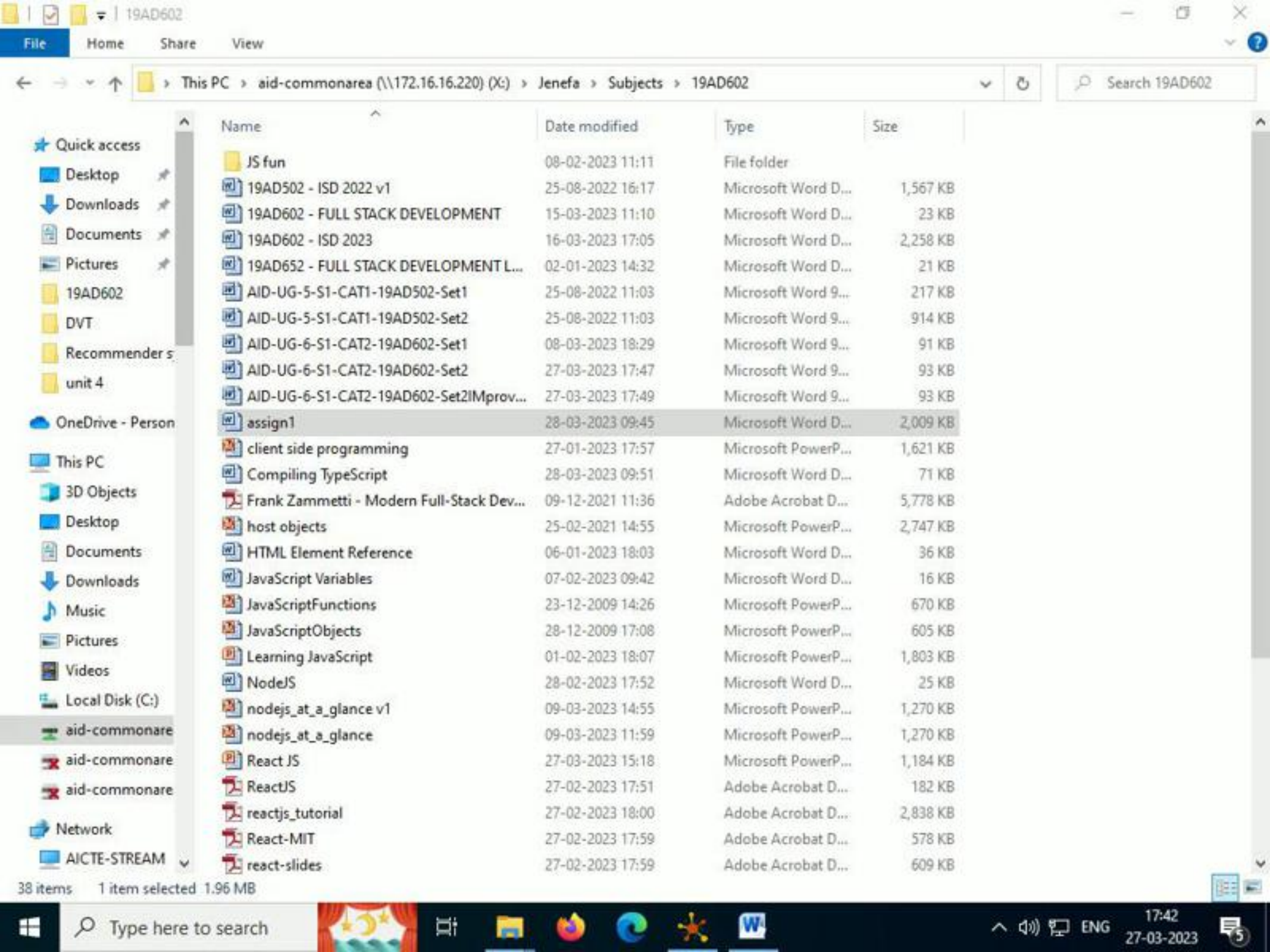
Screen Recording Started

Type here to search

🏠📁🔍🌐🔧

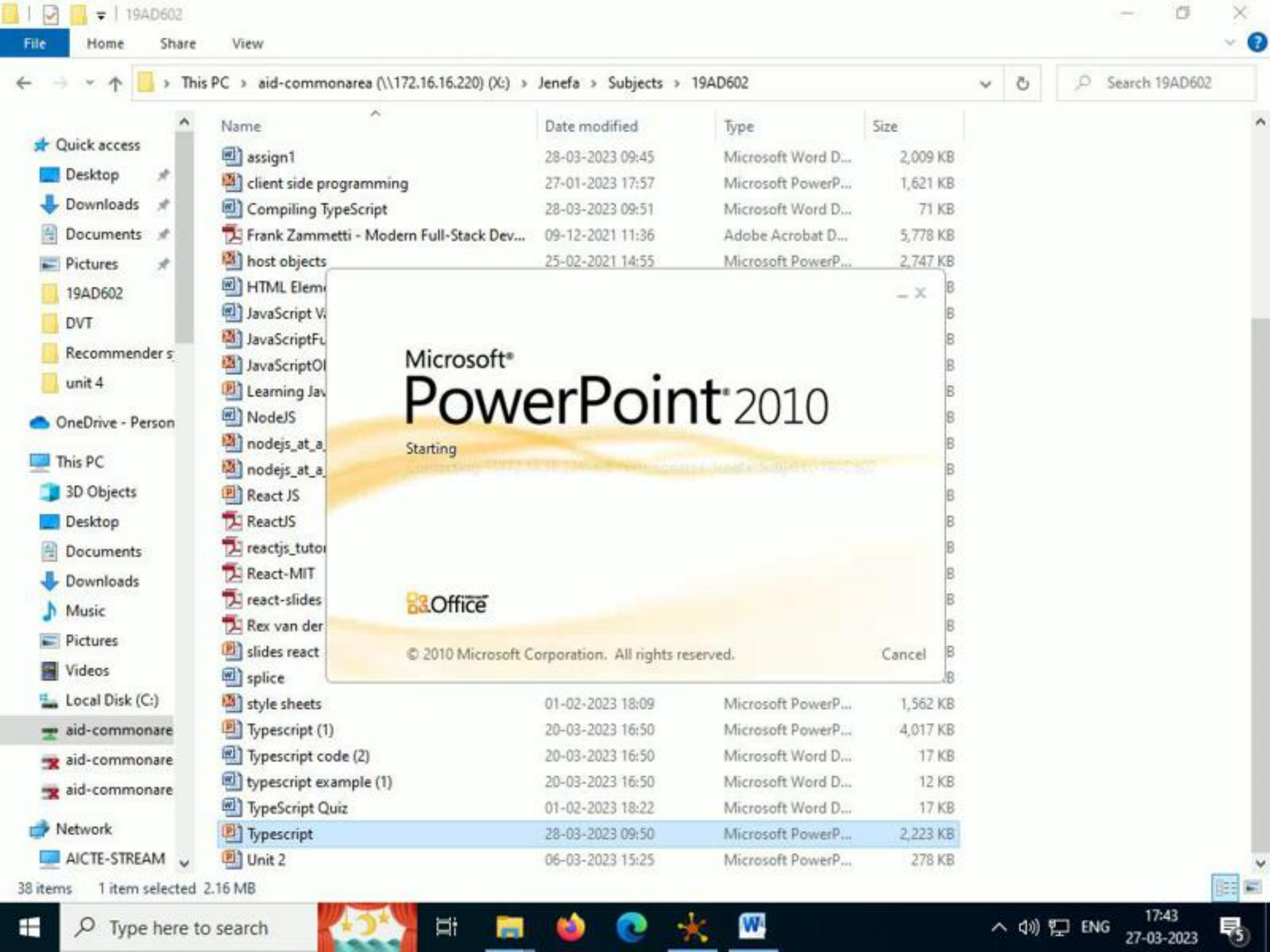
17:4227-03-2023

ENG



- ★ Quick access
 - Desktop
 - Downloads
 - Documents
 - Pictures
 - 19AD602
 - DVT
 - Recommender s
 - unit 4
- OneDrive - Person
- This PC
 - 3D Objects
 - Desktop
 - Documents
 - Downloads
 - Music
 - Pictures
 - Videos
 - Local Disk (C:)
 - aid-commonare
 - aid-commonare
 - aid-commonare
- Network
 - AICTE-STREAM

Name	Date modified	Type	Size
JS fun	08-02-2023 11:11	File folder	
19AD502 - ISD 2022 v1	25-08-2022 16:17	Microsoft Word D...	1,567 KB
19AD602 - FULL STACK DEVELOPMENT	15-03-2023 11:10	Microsoft Word D...	23 KB
19AD602 - ISD 2023	16-03-2023 17:05	Microsoft Word D...	2,258 KB
19AD652 - FULL STACK DEVELOPMENT L...	02-01-2023 14:32	Microsoft Word D...	21 KB
AID-UG-5-S1-CAT1-19AD502-Set1	25-08-2022 11:03	Microsoft Word 9...	217 KB
AID-UG-5-S1-CAT1-19AD502-Set2	25-08-2022 11:03	Microsoft Word 9...	914 KB
AID-UG-6-S1-CAT2-19AD602-Set1	08-03-2023 18:29	Microsoft Word 9...	91 KB
AID-UG-6-S1-CAT2-19AD602-Set2	27-03-2023 17:47	Microsoft Word 9...	93 KB
AID-UG-6-S1-CAT2-19AD602-Set2IMprov...	27-03-2023 17:49	Microsoft Word 9...	93 KB
assign1	28-03-2023 09:45	Microsoft Word D...	2,009 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
Frank Zammetti - Modern Full-Stack Dev...	09-12-2021 11:36	Adobe Acrobat D...	5,778 KB
host objects	25-02-2021 14:55	Microsoft PowerP...	2,747 KB
HTML Element Reference	06-01-2023 18:03	Microsoft Word D...	36 KB
JavaScript Variables	07-02-2023 09:42	Microsoft Word D...	16 KB
JavaScriptFunctions	23-12-2009 14:26	Microsoft PowerP...	670 KB
JavaScriptObjects	28-12-2009 17:08	Microsoft PowerP...	605 KB
Learning JavaScript	01-02-2023 18:07	Microsoft PowerP...	1,803 KB
NodeJS	28-02-2023 17:52	Microsoft Word D...	25 KB
nodejs_at_a_glance v1	09-03-2023 14:55	Microsoft PowerP...	1,270 KB
nodejs_at_a_glance	09-03-2023 11:59	Microsoft PowerP...	1,270 KB
React JS	27-03-2023 15:18	Microsoft PowerP...	1,184 KB
ReactJS	27-02-2023 17:51	Adobe Acrobat D...	182 KB
reactjs_tutorial	27-02-2023 18:00	Adobe Acrobat D...	2,838 KB
React-MIT	27-02-2023 17:59	Adobe Acrobat D...	578 KB
react-slides	27-02-2023 17:59	Adobe Acrobat D...	609 KB



- ★ Quick access
- Desktop
- Downloads
- Documents
- Pictures
- 19AD602
- DVT
- Recommender s
- unit 4
- OneDrive - Person
- This PC
- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Local Disk (C:)
- aid-commonare
- aid-commonare
- aid-commonare
- Network
- AICTE-STREAM

Name	Date modified	Type	Size
assign1	28-03-2023 09:45	Microsoft Word D...	2,009 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
Frank Zammetti - Modern Full-Stack Dev...	09-12-2021 11:36	Adobe Acrobat D...	5,778 KB
host objects	25-02-2021 14:55	Microsoft PowerP...	2,747 KB
HTML Elements			
JavaScript Variables			
JavaScript Functions			
JavaScript Objects			
Learning JavaScript			
NodeJS			
nodejs_at_a			
nodejs_at_a			
React JS			
ReactJS			
reactjs_tutorial			
React-MIT			
react-slides			
Rex van der			
slides react			
splice			
style sheets	01-02-2023 18:09	Microsoft PowerP...	1,562 KB
Typescript (1)	20-03-2023 16:50	Microsoft PowerP...	4,017 KB
Typescript code (2)	20-03-2023 16:50	Microsoft Word D...	17 KB
typescript example (1)	20-03-2023 16:50	Microsoft Word D...	12 KB
TypeScript Quiz	01-02-2023 18:22	Microsoft Word D...	17 KB
Typescript	28-03-2023 09:50	Microsoft PowerP...	2,223 KB
Unit 2	06-03-2023 15:25	Microsoft PowerP...	278 KB

Microsoft®
PowerPoint® 2010

Starting

Office

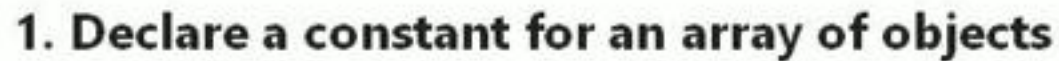
© 2010 Microsoft Corporation. All rights reserved.

Cancel

1. Display a list of all cars with details and images.
2. Filter car list by years ie displays all cars with release_year 2020.
3. Filter car list by brand name from the dropdown, ie display all BMW cars



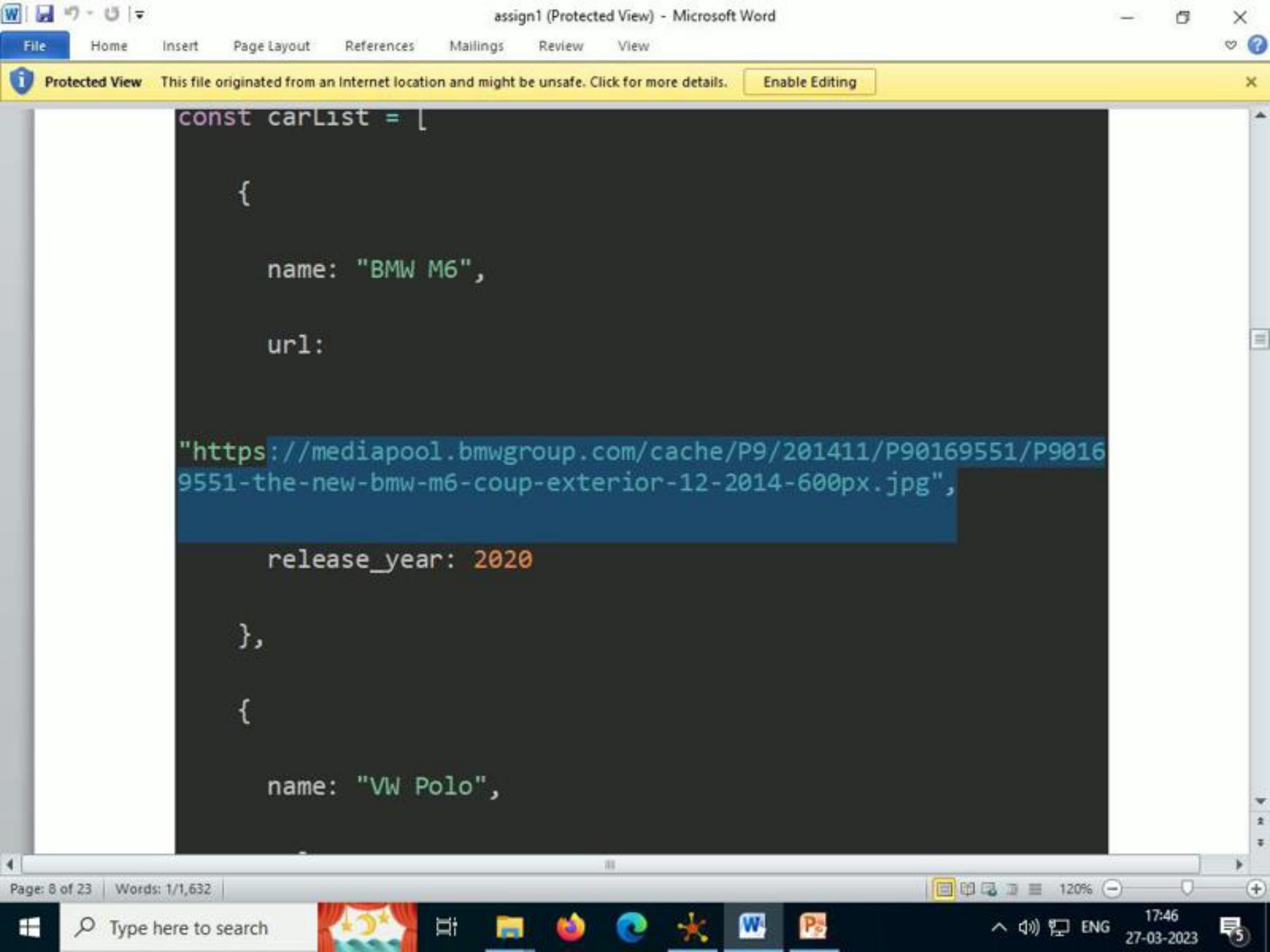
Every car is represented as a JS object with **name**, **url**, and **release_year** properties. The list of car objects is declared as to "carList" JS constant.



Every car is represented as a JS object with **name**, **url**, and **release_year** properties. The list of car objects is declared as an array assigned to **"carList"** JS constant.

Every car is represented as a JS object with **name**, **url**, and **release_year** properties. The list of car objects is declared as an array assigned to “**carList**” JS constant.

```
const carList = [
  {
    name: "BMW M6",
    url:
```



```
const carList = [  
  
  {  
  
    name: "BMW M6",  
  
    url:  
  
    "https://mediapool.bmwgroup.com/cache/P9/201411/P90169551/P90169551-the-new-bmw-m6-coup-exterior-12-2014-600px.jpg",  
  
    release_year: 2020  
  
  },  
  
  {  
  
    name: "VW Polo",
```

1. **filteredList**: State for list of all car objects that satisfy all the filter conditions.
2. **selectedBrand**: State for the brand_name value based on which filteredList state has to be filtered.
3. **selectedYear**: State for the year which is used to filter car object based on release_year property.

```
// List of all cars satisfying all the filters

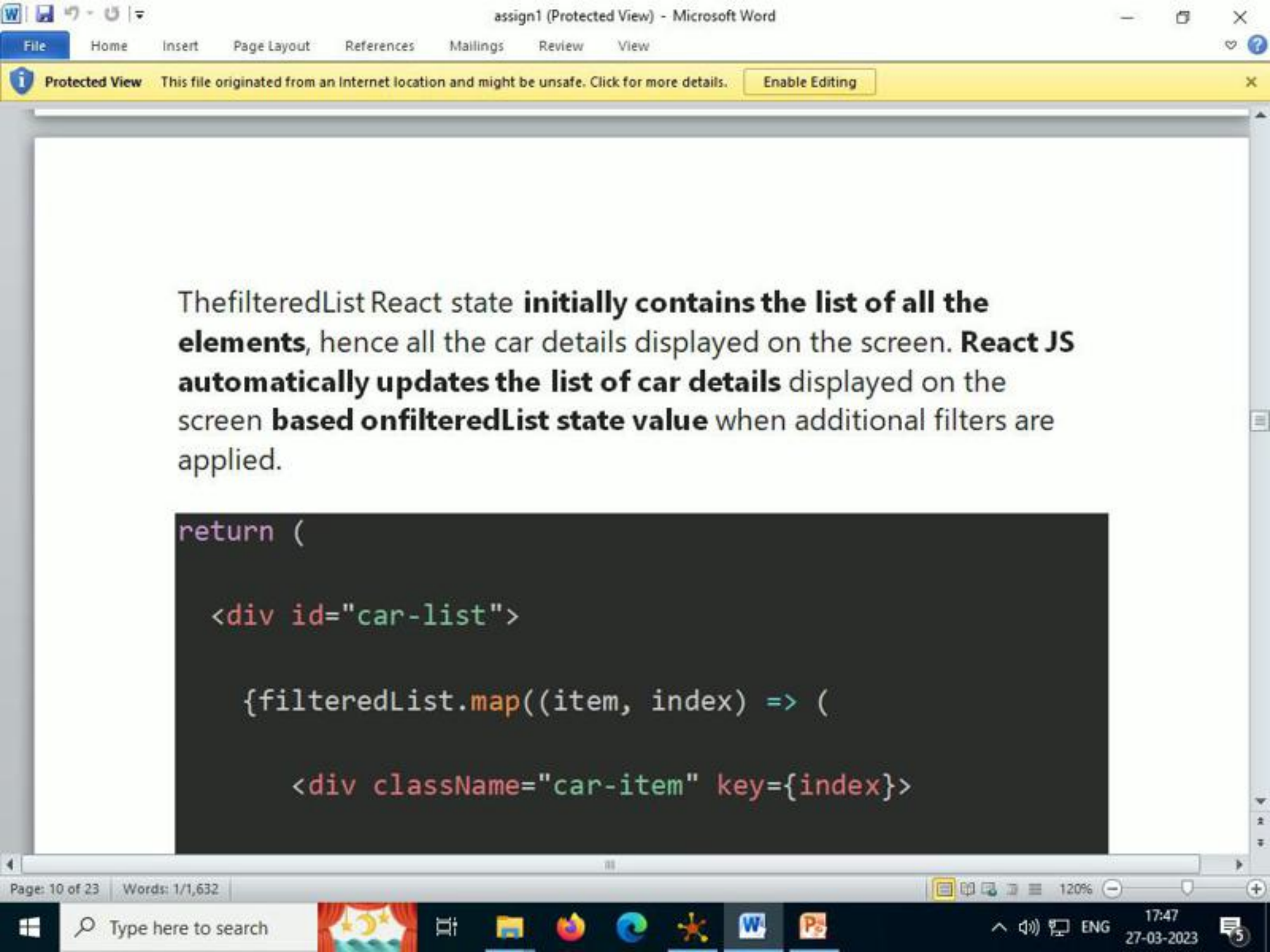
const [filteredList, setFilteredList] = useState(carList);

// Selected Brand name filter

const [selectedBrand, setSelectedBrand] = useState("");

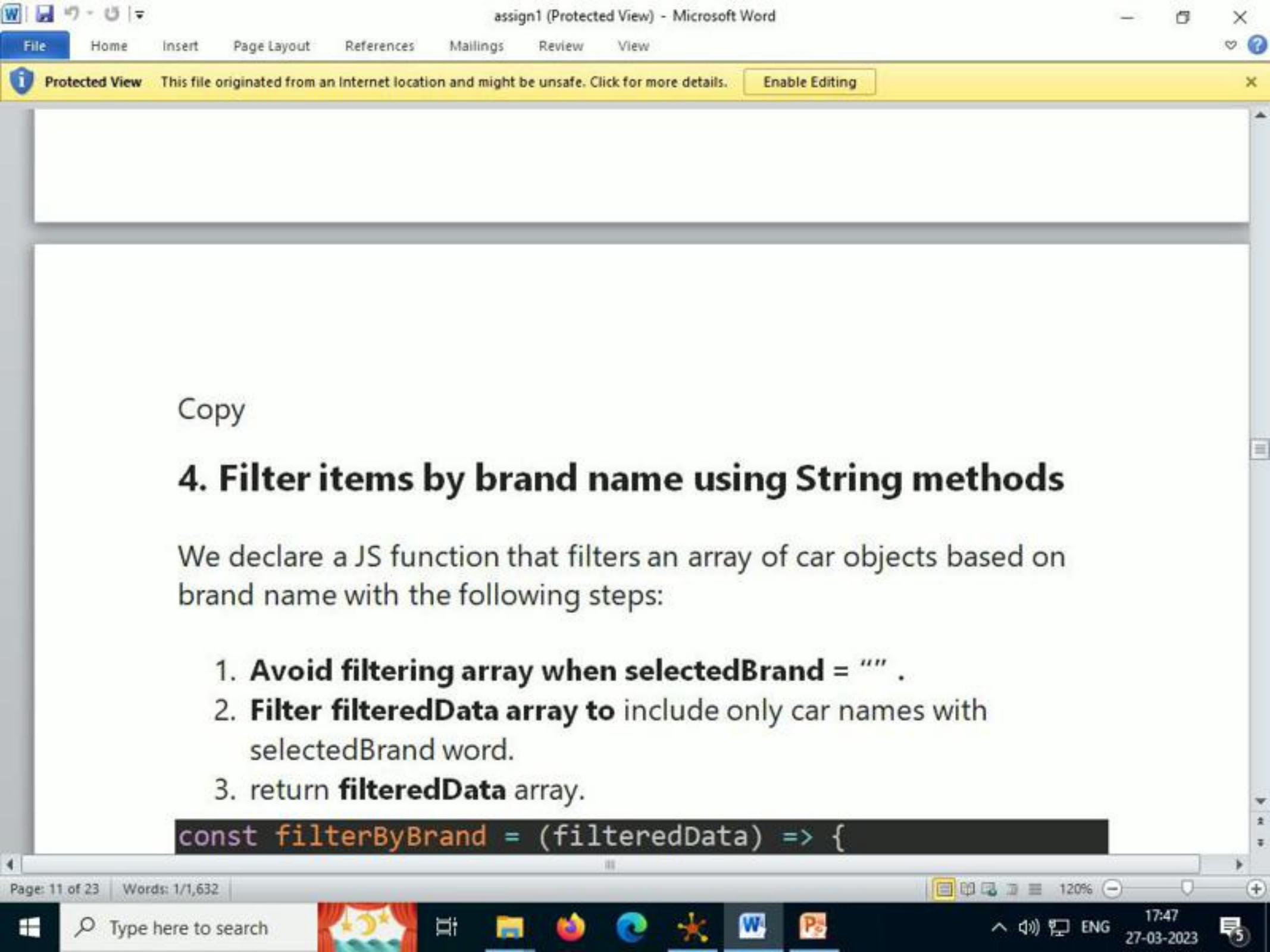
// Selected Year filter

const [selectedYear, setSelectedYear] = useState();
```

ThefilteredList React state **initially contains the list of all the elements**, hence all the car details displayed on the screen. **React JS automatically updates the list of car details** displayed on the screen **based onfilteredList state value** when additional filters are applied.

```
return (  
  
  <div id="car-list">  
  
    {filteredList.map((item, index) => (  
  
      <div className="car-item" key={index}>
```



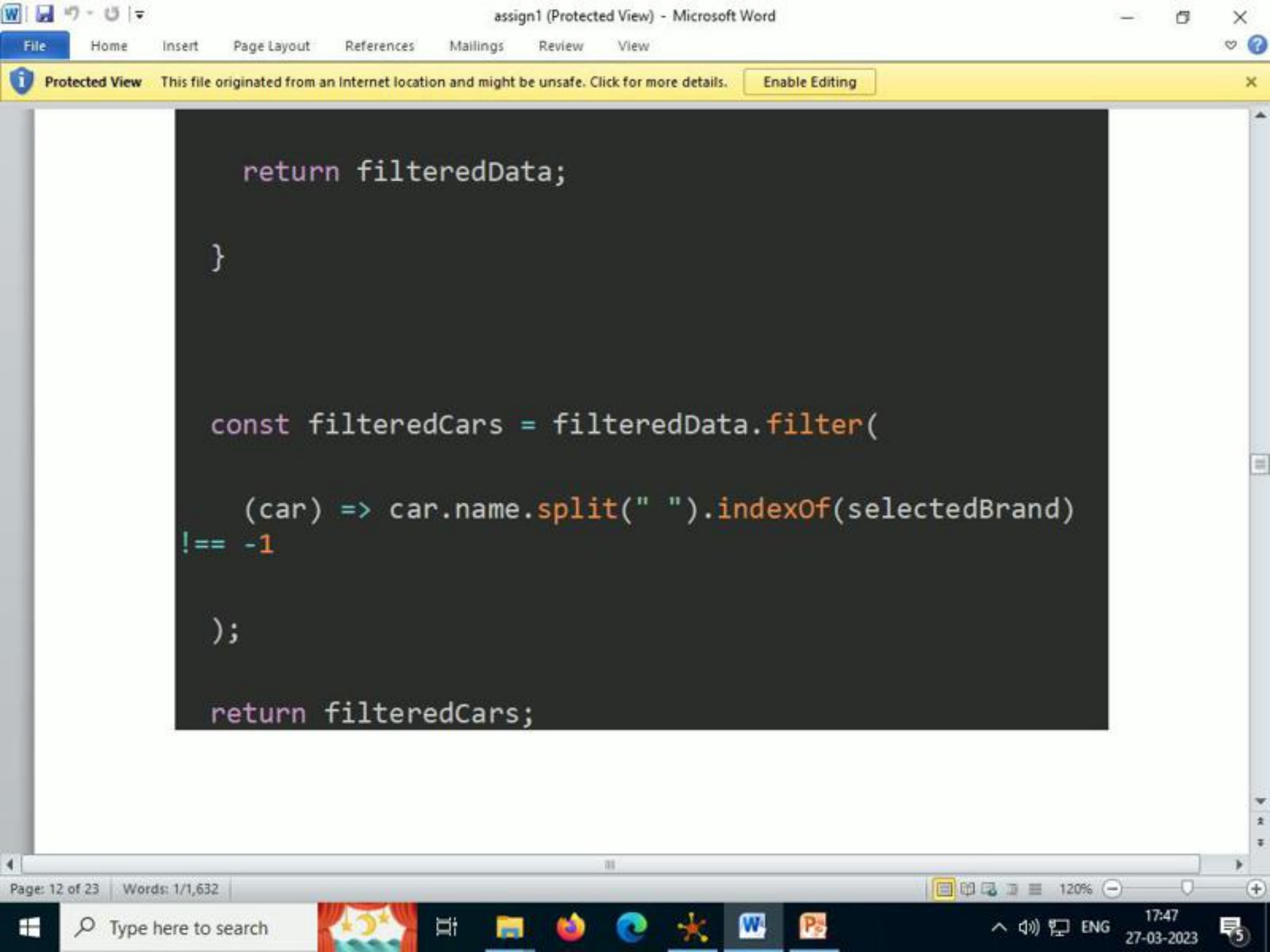
Copy

4. Filter items by brand name using String methods

We declare a JS function that filters an array of car objects based on brand name with the following steps:

1. **Avoid filtering array when selectedBrand = "" .**
2. **Filter filteredData array to** include only car names with selectedBrand word.
3. return **filteredData** array.

```
const filterByBrand = (filteredData) => {
```



```
return filteredData;
```

```
}
```

```
const filteredCars = filteredData.filter(
```

```
  (car) => car.name.split(" ").indexOf(selectedBrand)  
  !== -1
```

```
);
```

```
return filteredCars;
```


W

assign1 (Protected View) - Microsoft Word

FileHomeInsertPage LayoutReferencesMailingsReviewView

Protected ViewThis file originated from an Internet location and might be unsafe. Click for more details.Enable Editing

};

5. Filter items by Year using the Comparison operator

We declare JS function that filters array of car objects by release_year property with the following steps:

1. Avoid filtering array when **selectedYear == null**
2. Filter array to include all the car objects where **release_year property matches selectedYear.**
3. return **filteredData** array.

```
const filterByYear = (filteredData) => {  
  
    // Avoid filter for null value
```

Page: 12 of 23Words: 1/1,632

120%

17:4827-03-2023

```
const inputYear = Number(event.target.id);

if (inputYear === selectedYear) {

    setSelectedYear("");

}
```

}

selecting the brand name filter values. The options for the filter by year is displayed using `<div></div>` containers.

```
<div className="brand-filter">

  <div>Filter by Brand :</div>

  <select

    id="brand-input"

    value={selectedBrand}

    onChange={handleBrandChange}

  >
```


To **allow users to trigger filter functions**, we need to add HTML code for both filters by brand name and year. We will create an **HTML dropdown for selecting the brand name** filter values. The options for the filter by year is displayed using `<div></div>` containers.

```
<div className="brand-filter">
  <div>Filter by Brand :</div>
  <select
    id="brand-input"
    value={selectedBrand}
    onChange={handleBrandChange}
  >
```

```
id="brand-input"
```

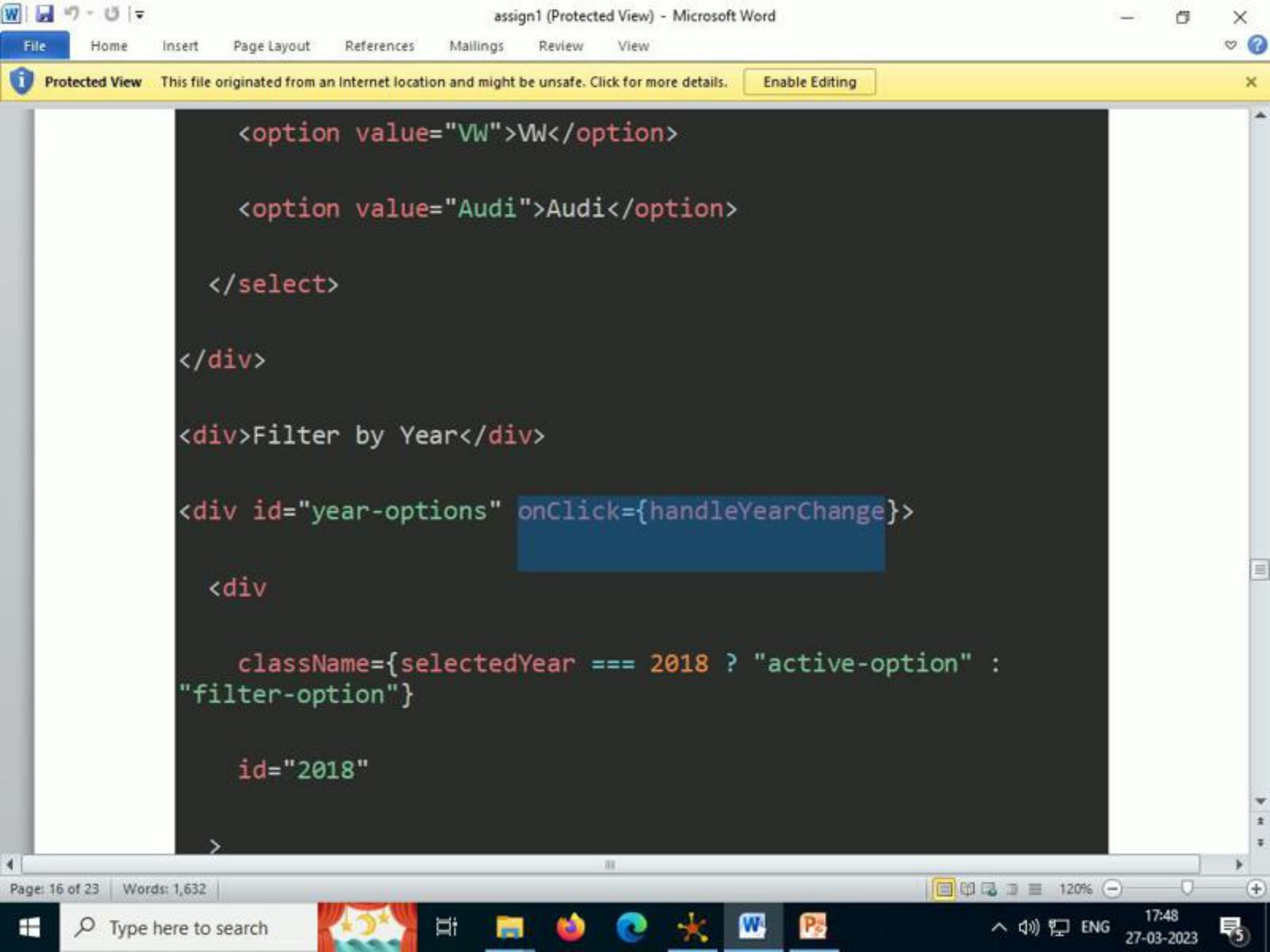
```
value={selectedBrand}
```

```
onChange={handleBrandChange}
```

```
>
```

```
<option value="">All</option>
```

```
<option value="BMW">BMW</option>
```



```
<option value="VW">VW</option>
```

```
<option value="Audi">Audi</option>
```

```
</select>
```

```
</div>
```

```
<div>Filter by Year</div>
```

```
<div id="year-options" onClick={handleYearChange}>
```

```
<div
```

```
  className={selectedYear === 2018 ? "active-option" :
  "filter-option"}
```

```
  id="2018"
```

```
>
```


states are added as the dependencies for `useEffect()` to ensure filter functionality is triggered every time any one of the values is changed.

```
useEffect(() => {

    var filteredData = filterByBrand(carList);

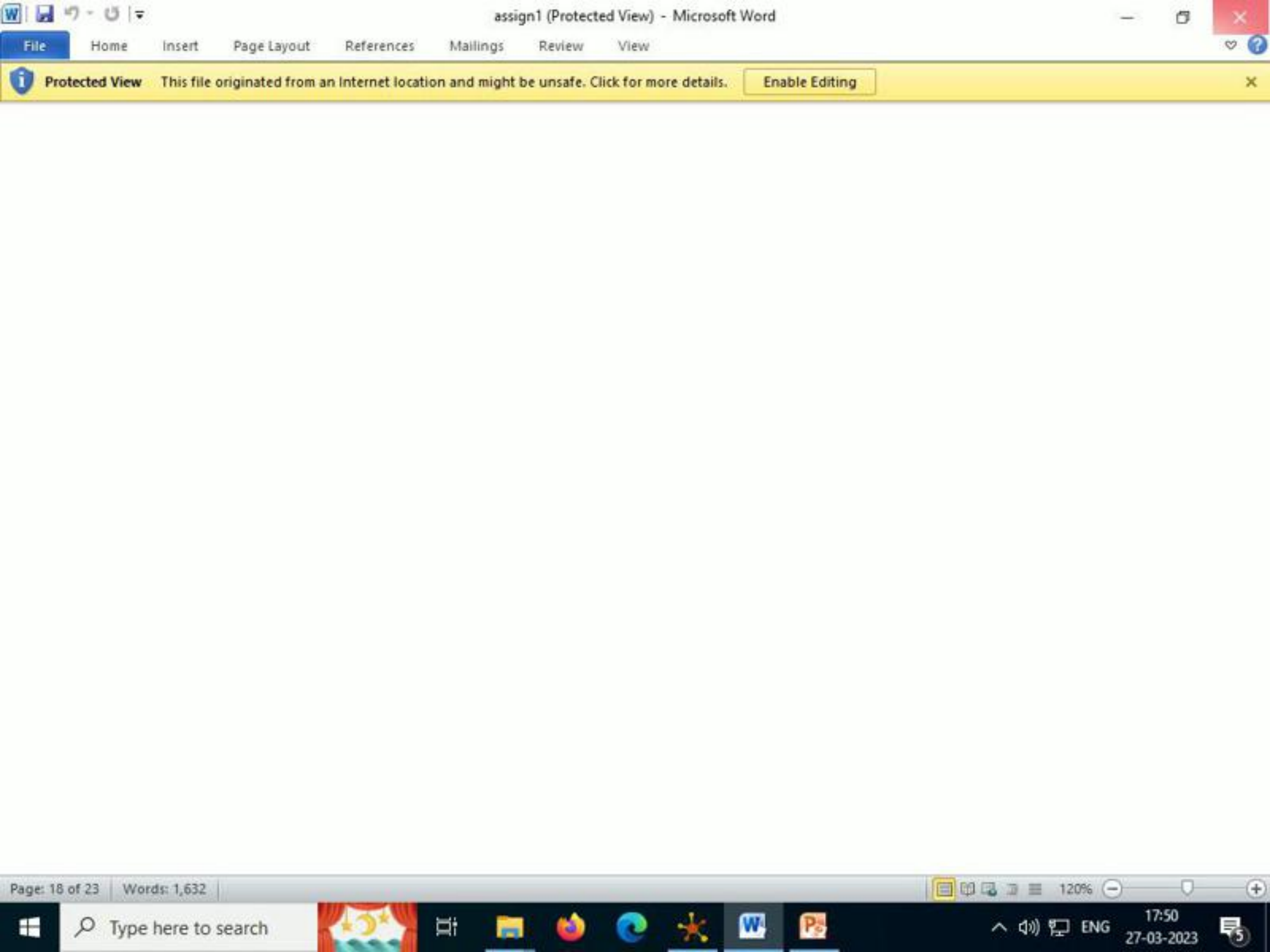
    filteredData = filterByYear(filteredData);

    setFilteredList(filteredData);

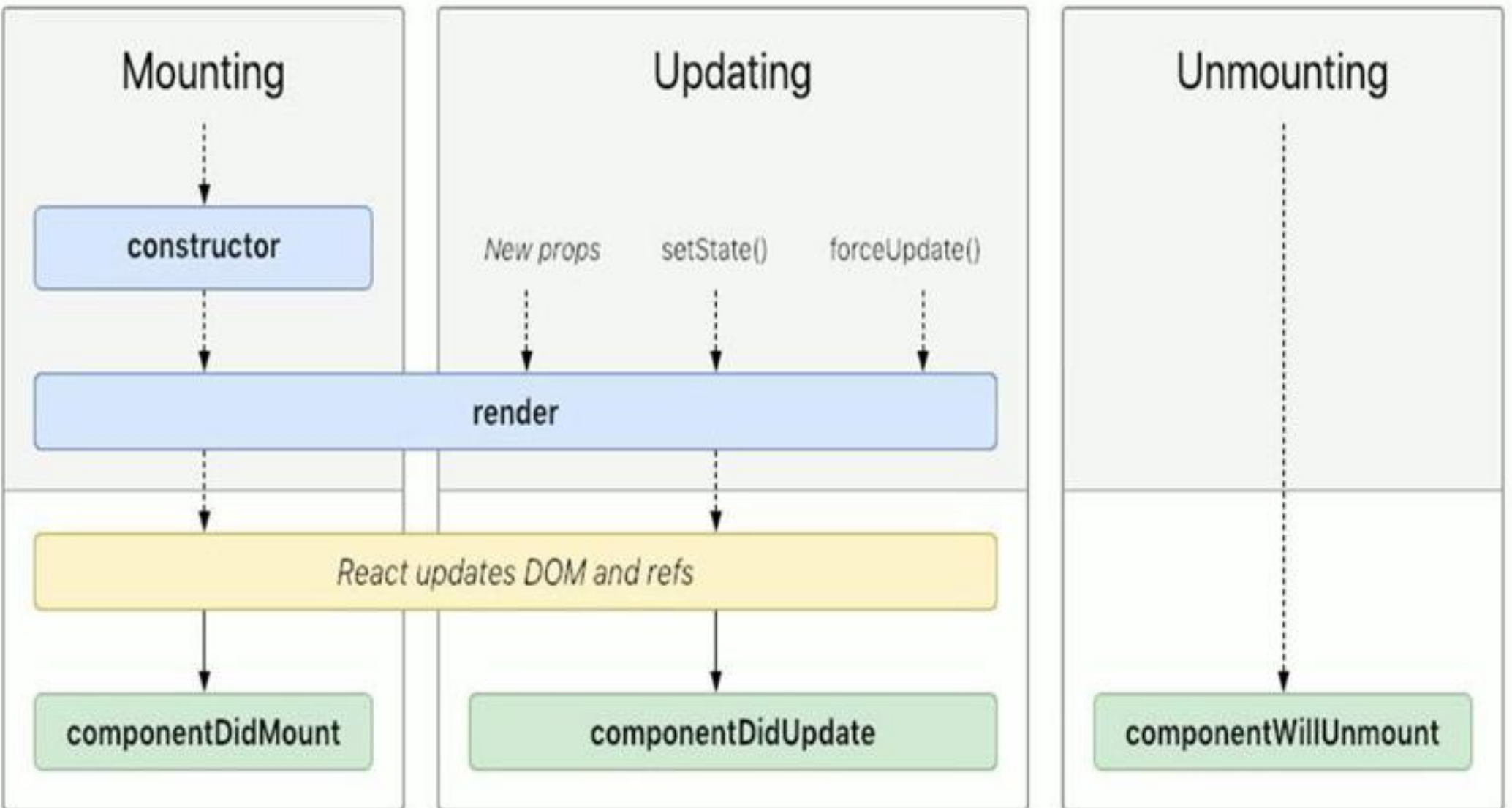
},

[selectedBrand, selectedYear]);
```

Copy



Component Lifecycle




U don't get error instead o/p will be 25

```
JS no-typescript.js X index.html
1  function add(a, b) {
2    |   return a + b;
3  }
4
5  const result = add('2', '5');
6
7  console.log(result);
```

th-typescript.ts • index.html

```
function add(a: number, b: number) {  
  return a + b;  
}  
  
const result = add('2', '5');  
  
console.log(result);
```



Argument of type 'string' is not assignable to parameter of type 'number'. ts(2345)

Primitive types

```
// Primitives

let age: number;

age = 12;

let userName: string;

userName = 'Max';

let isInstructor: boolean;

isInstructor = true;
```

TypeScript Vs JavaScript

- you can't run TypeScript code in a browser or Node
- TypeScript must be compiled, or *transpiled*
- TypeScript gets compiled to JavaScript!
- TypeScript's home page at typescriptlang.org called "the playground"

Example code

The screenshot shows the TypeScript Playground interface. At the top, there's a header with the TypeScript logo and navigation links: v3.6.3, Config, Examples, What's new, Run, Shortcuts, and About. A modal dialog is open in the center, displaying the message "Hello, [object Object]!" and an "OK" button. The dialog also shows the source "www.typescriptlang.org says". Below the header, there are two code editors side-by-side. The left editor contains the following code:

```
1 function sayHi(humanName) {  
2   alert(`Hello, ${humanName}!`);  
3 }  
4 sayHi({ humanName : "Luke Skywalker" });  
5
```

The right editor contains the following code:

```
1 "use strict";  
2 function sayHi(humanName) {  
3   alert(`Hello, ${humanName}!`);  
4 }  
5 sayHi({ humanName: "Luke Skywalker" });  
6
```


- `function sayHi(humanName: string) {`
- `sayHi("Luke Skywalker");`

Example code

The screenshot displays the TypeScript Playground interface. At the top, a dark blue header contains the 'TypeScript' logo and navigation links: 'v3.6.3', 'Config', 'Examples', 'What's new', 'Run', 'Shortcuts', and 'About'. A modal alert box is centered on the screen, showing the text 'www.typescriptlang.org says' and 'Hello, [object Object]!' with an 'OK' button. Below the header, the editor is split into two panes. The left pane contains the following code:

```
1 function sayHi(humanName) {  
2   alert(`Hello, ${humanName}!`);  
3 }  
4 sayHi({ humanName : "Luke Skywalker" });  
5
```

The right pane contains the following code:

```
1 "use strict";  
2 function sayHi(humanName) {  
3   alert(`Hello, ${humanName}!`);  
4 }  
5 sayHi({ humanName: "Luke Skywalker" });  
6
```

- `function sayHi(humanName: string) {`
- `sayHi("Luke Skywalker");`

19AD602

FileHomeShareView

←→↕↑

This PC > aid-commonarea (\\172.16.16.220) (X:) > Jenefa > Subjects > 19AD602

↻

Search 19AD602

★ Quick access

Desktop

Downloads

Documents

Pictures

19AD602

DVT

Recommender s

unit 4

OneDrive - Person

This PC

3D Objects

Desktop

Documents

Downloads

Music

Pictures

Videos

Local Disk (C:)

aid-commonare

aid-commonare

aid-commonare

Network

AICTE-STREAM

Name	Date modified	Type	Size
assign1	28-03-2023 09:45	Microsoft Word D...	2,009 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
Frank Zammetti - Modern Full-Stack Dev...	09-12-2021 11:36	Adobe Acrobat D...	5,778 KB
host objects	25-02-2021 14:55	Microsoft PowerP...	2,747 KB
HTML Element Reference	06-01-2023 18:03	Microsoft Word D...	36 KB
JavaScript Variables	07-02-2023 09:42	Microsoft Word D...	16 KB
JavaScriptFunctions	23-12-2009 14:26	Microsoft PowerP...	670 KB
JavaScriptObjects	28-12-2009 17:08	Microsoft PowerP...	605 KB
Learning JavaScript	01-02-2023 18:07	Microsoft PowerP...	1,803 KB
NodeJS	28-02-2023 17:52	Microsoft Word D...	25 KB
nodejs_at_a_glance v1	09-03-2023 14:55	Microsoft PowerP...	1,270 KB
nodejs_at_a_glance	09-03-2023 11:59	Microsoft PowerP...	1,270 KB
React JS	27-03-2023 15:18	Microsoft PowerP...	1,184 KB
ReactJS	27-02-2023 17:51	Adobe Acrobat D...	182 KB
reactjs_tutorial	27-02-2023 18:00	Adobe Acrobat D...	2,838 KB
React-MIT	27-02-2023 17:59	Adobe Acrobat D...	578 KB
react-slides	27-02-2023 17:59	Adobe Acrobat D...	609 KB
Rex van der Spuy - Foundation game des...	09-12-2021 16:30	Adobe Acrobat D...	15,779 KB
slides react	27-02-2023 17:55	Microsoft PowerP...	1,689 KB
splice	16-02-2017 13:40	Microsoft Word D...	382 KB
style sheets	01-02-2023 18:09	Microsoft PowerP...	1,562 KB
Typescript (1)	20-03-2023 16:50	Microsoft PowerP...	4,017 KB
Typescript code (2)	20-03-2023 16:50	Microsoft Word D...	17 KB
typescript example (1)	20-03-2023 16:50	Microsoft Word D...	12 KB
TypeScript Quiz	01-02-2023 18:22	Microsoft Word D...	17 KB
Typescript	28-03-2023 09:50	Microsoft PowerP...	2,223 KB
Unit 2	06-03-2023 15:25	Microsoft PowerP...	278 KB

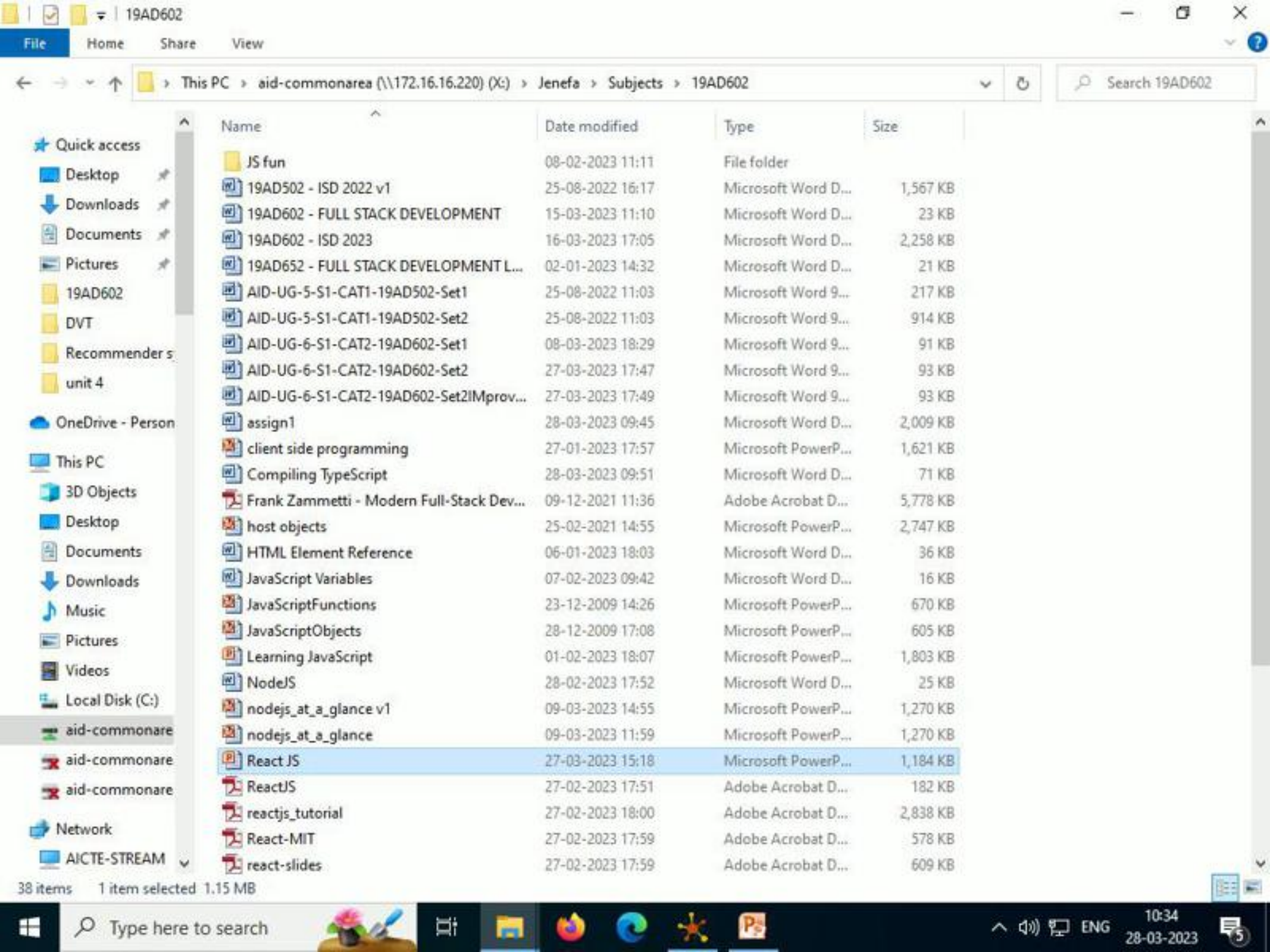
38 items1 item selected1.15 MB

Windows Taskbar

Search: Type here to search

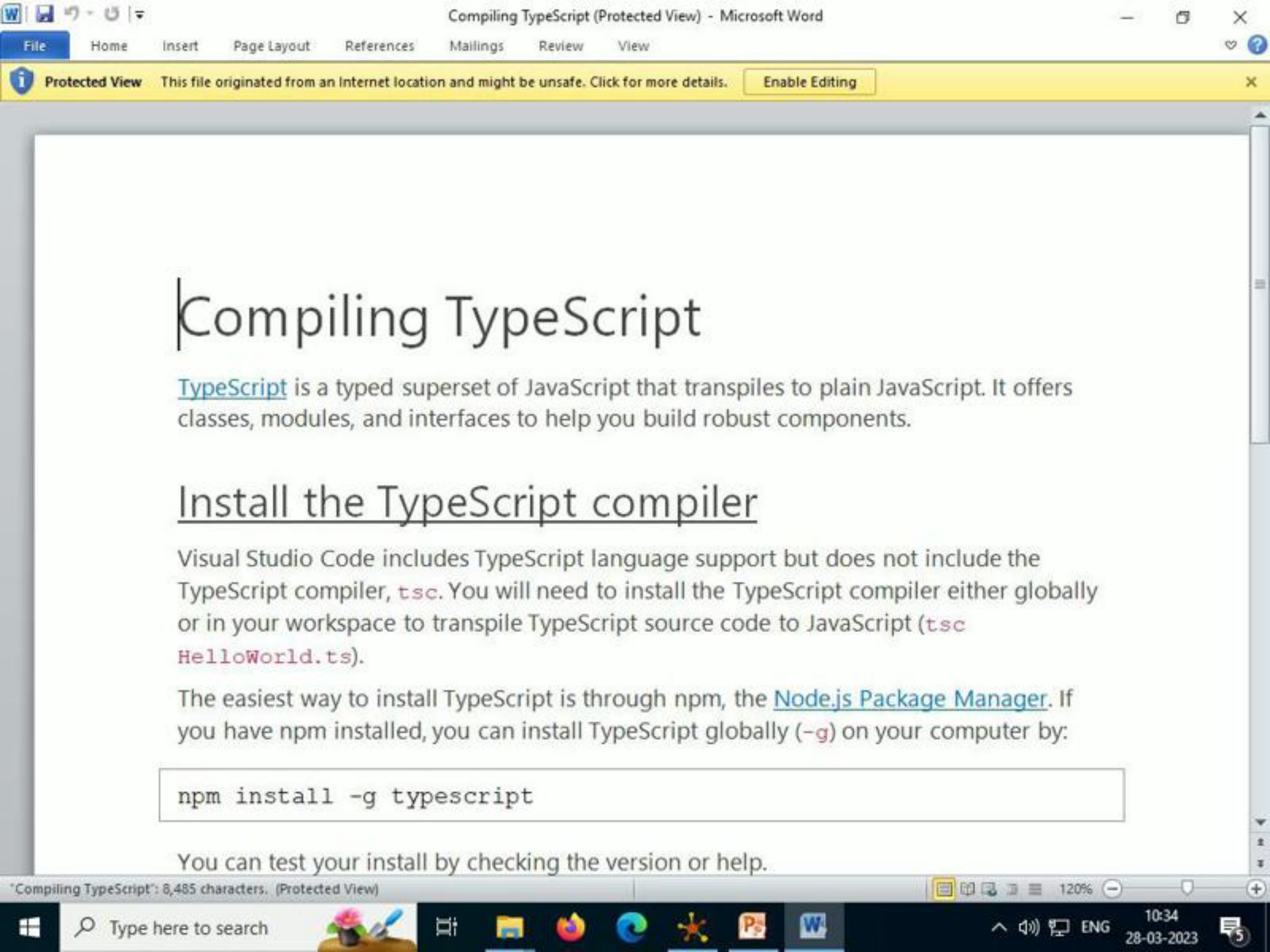
Taskbar Icons: File Explorer, Firefox, Edge, VS Code, PowerPoint

System Tray: Volume, Network, Language (ENG), Date/Time (10:34 28-03-2023)



- ★ Quick access
 - Desktop
 - Downloads
 - Documents
 - Pictures
 - 19AD602
 - DVT
 - Recommender s
 - unit 4
- OneDrive - Person
- This PC
 - 3D Objects
 - Desktop
 - Documents
 - Downloads
 - Music
 - Pictures
 - Videos
 - Local Disk (C:)
 - aid-commonare
 - aid-commonare
 - aid-commonare
- Network
 - AICTE-STREAM

Name	Date modified	Type	Size
JS fun	08-02-2023 11:11	File folder	
19AD502 - ISD 2022 v1	25-08-2022 16:17	Microsoft Word D...	1,567 KB
19AD602 - FULL STACK DEVELOPMENT	15-03-2023 11:10	Microsoft Word D...	23 KB
19AD602 - ISD 2023	16-03-2023 17:05	Microsoft Word D...	2,258 KB
19AD652 - FULL STACK DEVELOPMENT L...	02-01-2023 14:32	Microsoft Word D...	21 KB
AID-UG-5-S1-CAT1-19AD502-Set1	25-08-2022 11:03	Microsoft Word 9...	217 KB
AID-UG-5-S1-CAT1-19AD502-Set2	25-08-2022 11:03	Microsoft Word 9...	914 KB
AID-UG-6-S1-CAT2-19AD602-Set1	08-03-2023 18:29	Microsoft Word 9...	91 KB
AID-UG-6-S1-CAT2-19AD602-Set2	27-03-2023 17:47	Microsoft Word 9...	93 KB
AID-UG-6-S1-CAT2-19AD602-Set2IMprov...	27-03-2023 17:49	Microsoft Word 9...	93 KB
assign1	28-03-2023 09:45	Microsoft Word D...	2,009 KB
client side programming	27-01-2023 17:57	Microsoft PowerP...	1,621 KB
Compiling TypeScript	28-03-2023 09:51	Microsoft Word D...	71 KB
Frank Zammetti - Modern Full-Stack Dev...	09-12-2021 11:36	Adobe Acrobat D...	5,778 KB
host objects	25-02-2021 14:55	Microsoft PowerP...	2,747 KB
HTML Element Reference	06-01-2023 18:03	Microsoft Word D...	36 KB
JavaScript Variables	07-02-2023 09:42	Microsoft Word D...	16 KB
JavaScriptFunctions	23-12-2009 14:26	Microsoft PowerP...	670 KB
JavaScriptObjects	28-12-2009 17:08	Microsoft PowerP...	605 KB
Learning JavaScript	01-02-2023 18:07	Microsoft PowerP...	1,803 KB
NodeJS	28-02-2023 17:52	Microsoft Word D...	25 KB
nodejs_at_a_glance v1	09-03-2023 14:55	Microsoft PowerP...	1,270 KB
nodejs_at_a_glance	09-03-2023 11:59	Microsoft PowerP...	1,270 KB
React JS	27-03-2023 15:18	Microsoft PowerP...	1,184 KB
ReactJS	27-02-2023 17:51	Adobe Acrobat D...	182 KB
reactjs_tutorial	27-02-2023 18:00	Adobe Acrobat D...	2,838 KB
React-MIT	27-02-2023 17:59	Adobe Acrobat D...	578 KB
react-slides	27-02-2023 17:59	Adobe Acrobat D...	609 KB



Compiling TypeScript

[TypeScript](#) is a typed superset of JavaScript that transpiles to plain JavaScript. It offers classes, modules, and interfaces to help you build robust components.

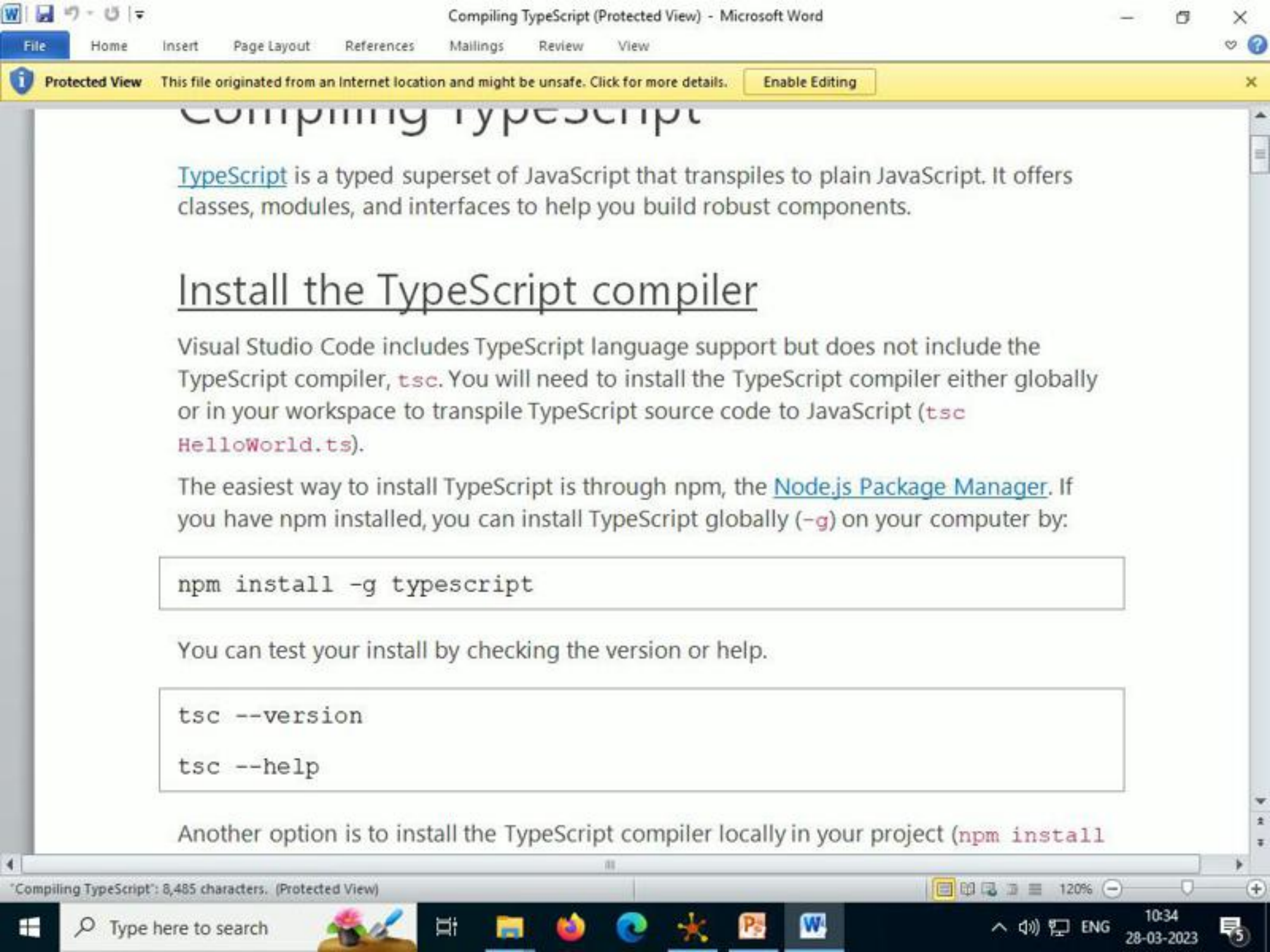
Install the TypeScript compiler

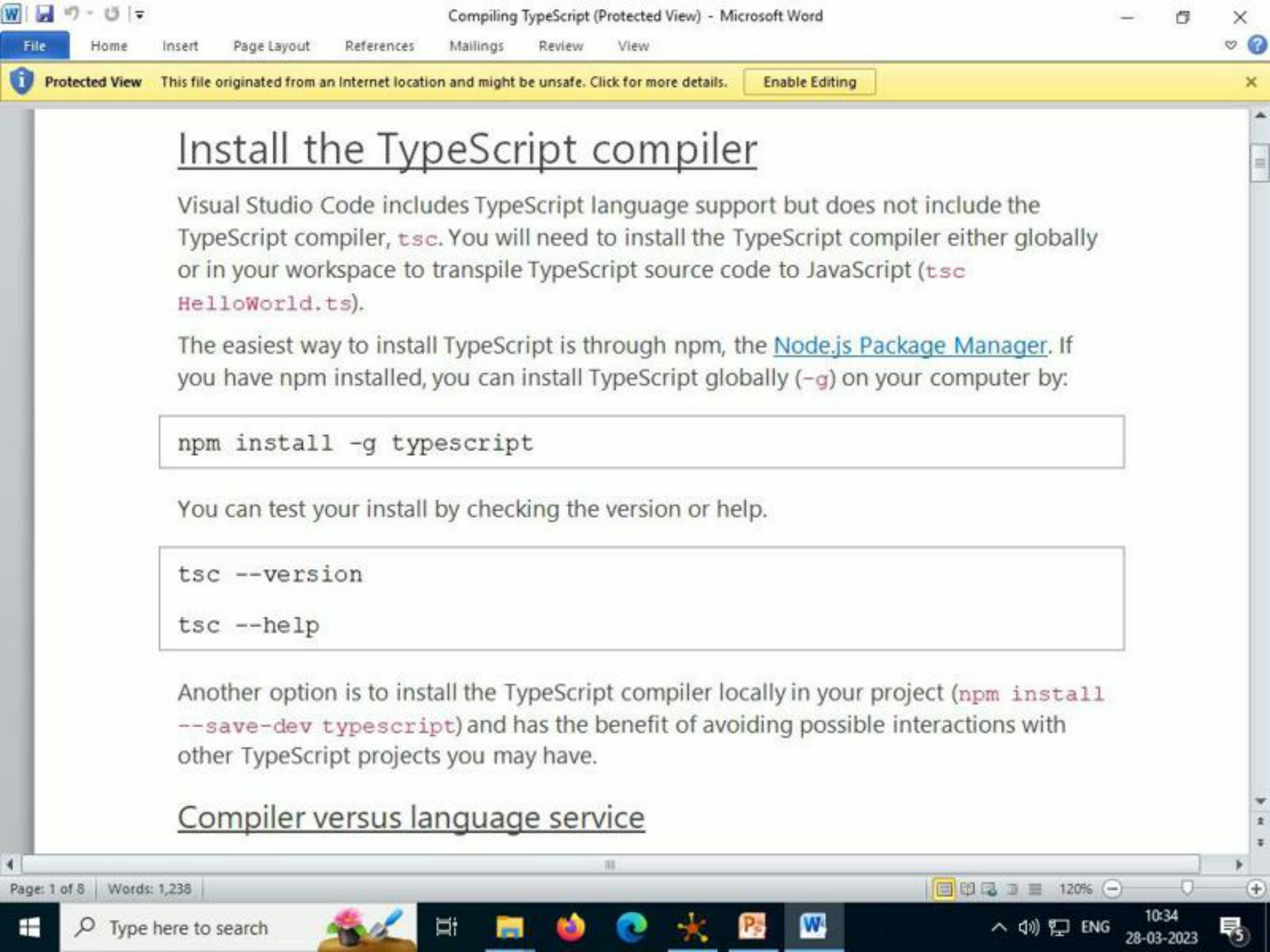
Visual Studio Code includes TypeScript language support but does not include the TypeScript compiler, `tsc`. You will need to install the TypeScript compiler either globally or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```

You can test your install by checking the version or help.





Install the TypeScript compiler

Visual Studio Code includes TypeScript language support but does not include the TypeScript compiler, `tsc`. You will need to install the TypeScript compiler either globally or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```

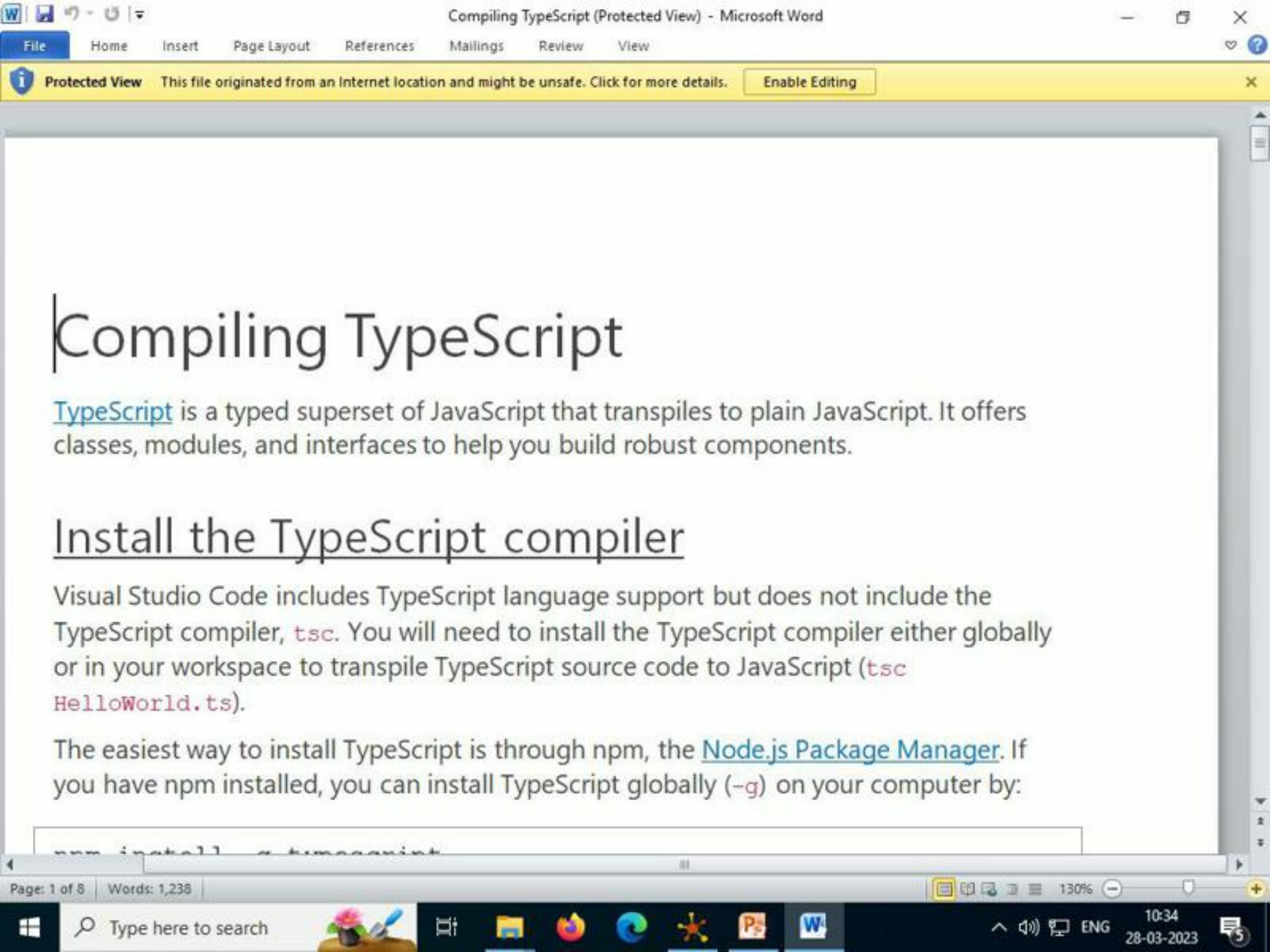
You can test your install by checking the version or help.

```
tsc --version
```

```
tsc --help
```

Another option is to install the TypeScript compiler locally in your project (`npm install --save-dev typescript`) and has the benefit of avoiding possible interactions with other TypeScript projects you may have.

Compiler versus language service



Compiling TypeScript

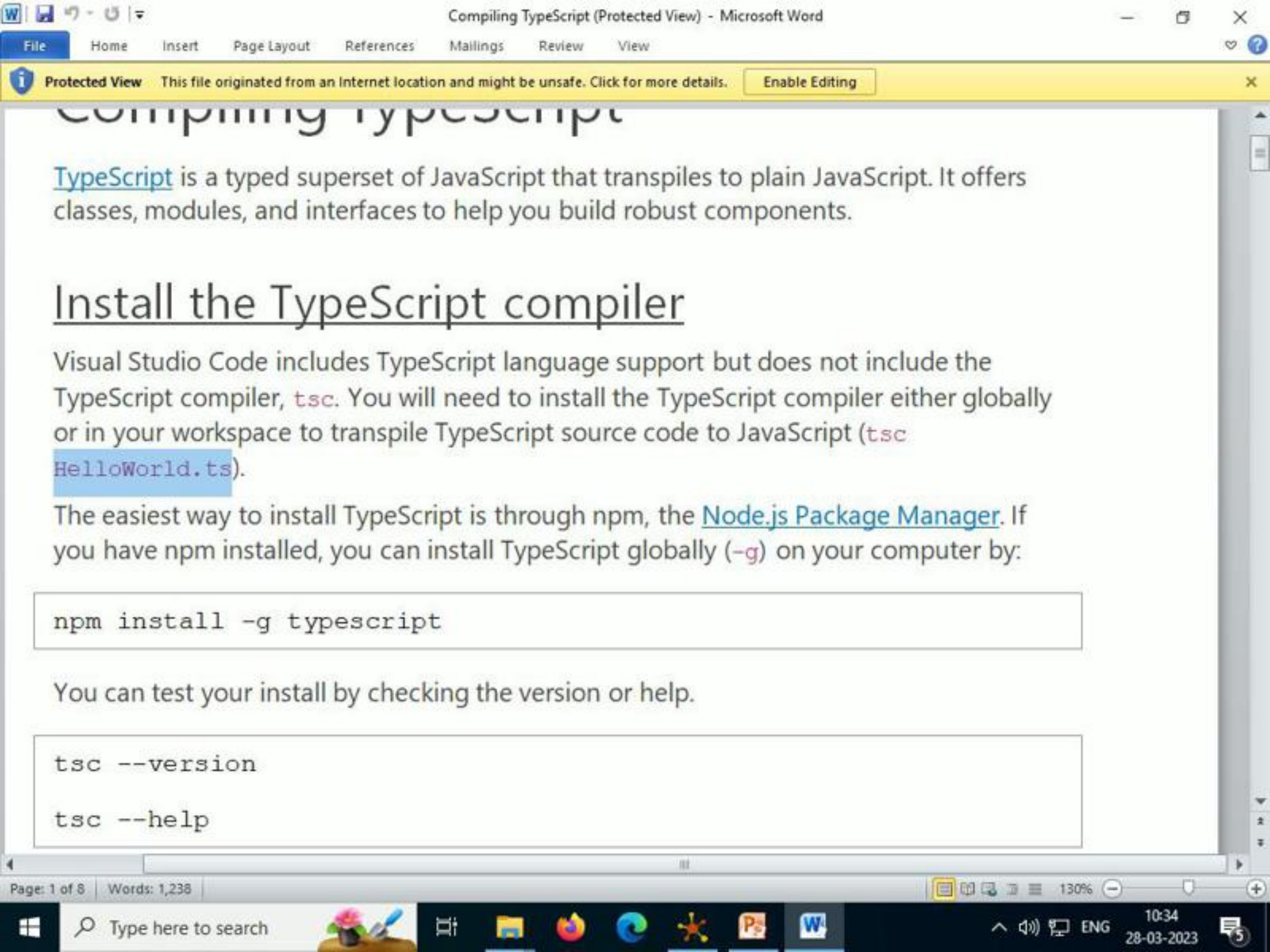
[TypeScript](#) is a typed superset of JavaScript that transpiles to plain JavaScript. It offers classes, modules, and interfaces to help you build robust components.

Install the TypeScript compiler

Visual Studio Code includes TypeScript language support but does not include the TypeScript compiler, `tsc`. You will need to install the TypeScript compiler either globally or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```



Compiling TypeScript

[TypeScript](#) is a typed superset of JavaScript that transpiles to plain JavaScript. It offers classes, modules, and interfaces to help you build robust components.

Install the TypeScript compiler

Visual Studio Code includes TypeScript language support but does not include the TypeScript compiler, `tsc`. You will need to install the TypeScript compiler either globally or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

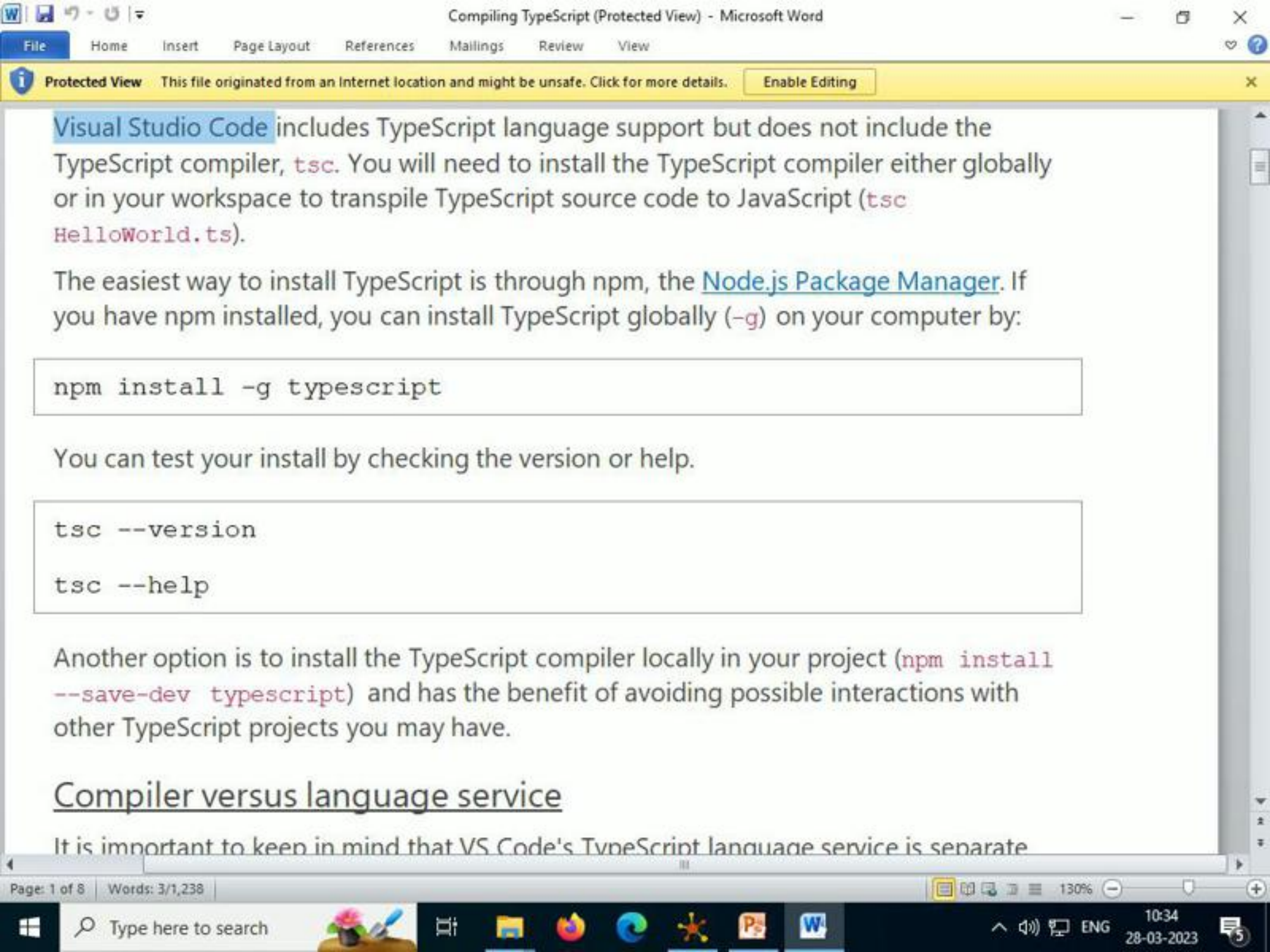
The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```

You can test your install by checking the version or help.

```
tsc --version
```

```
tsc --help
```

Visual Studio Code includes TypeScript language support but does not include the TypeScript compiler, `tsc`. You will need to install the TypeScript compiler either globally or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```

You can test your install by checking the version or help.

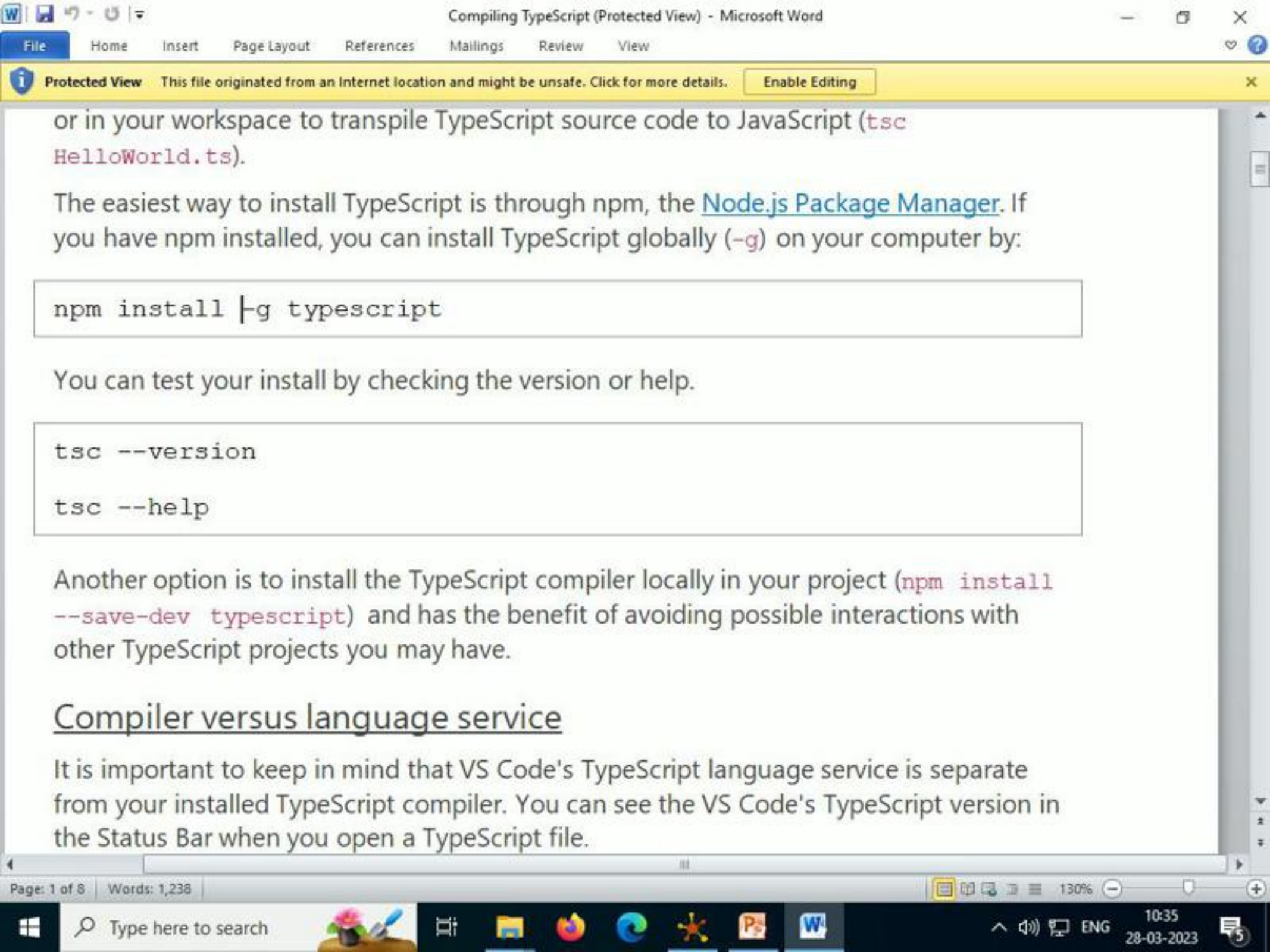
```
tsc --version
```

```
tsc --help
```

Another option is to install the TypeScript compiler locally in your project (`npm install --save-dev typescript`) and has the benefit of avoiding possible interactions with other TypeScript projects you may have.

Compiler versus language service

It is important to keep in mind that VS Code's TypeScript language service is separate



or in your workspace to transpile TypeScript source code to JavaScript (`tsc HelloWorld.ts`).

The easiest way to install TypeScript is through npm, the [Node.js Package Manager](#). If you have npm installed, you can install TypeScript globally (`-g`) on your computer by:

```
npm install -g typescript
```

You can test your install by checking the version or help.

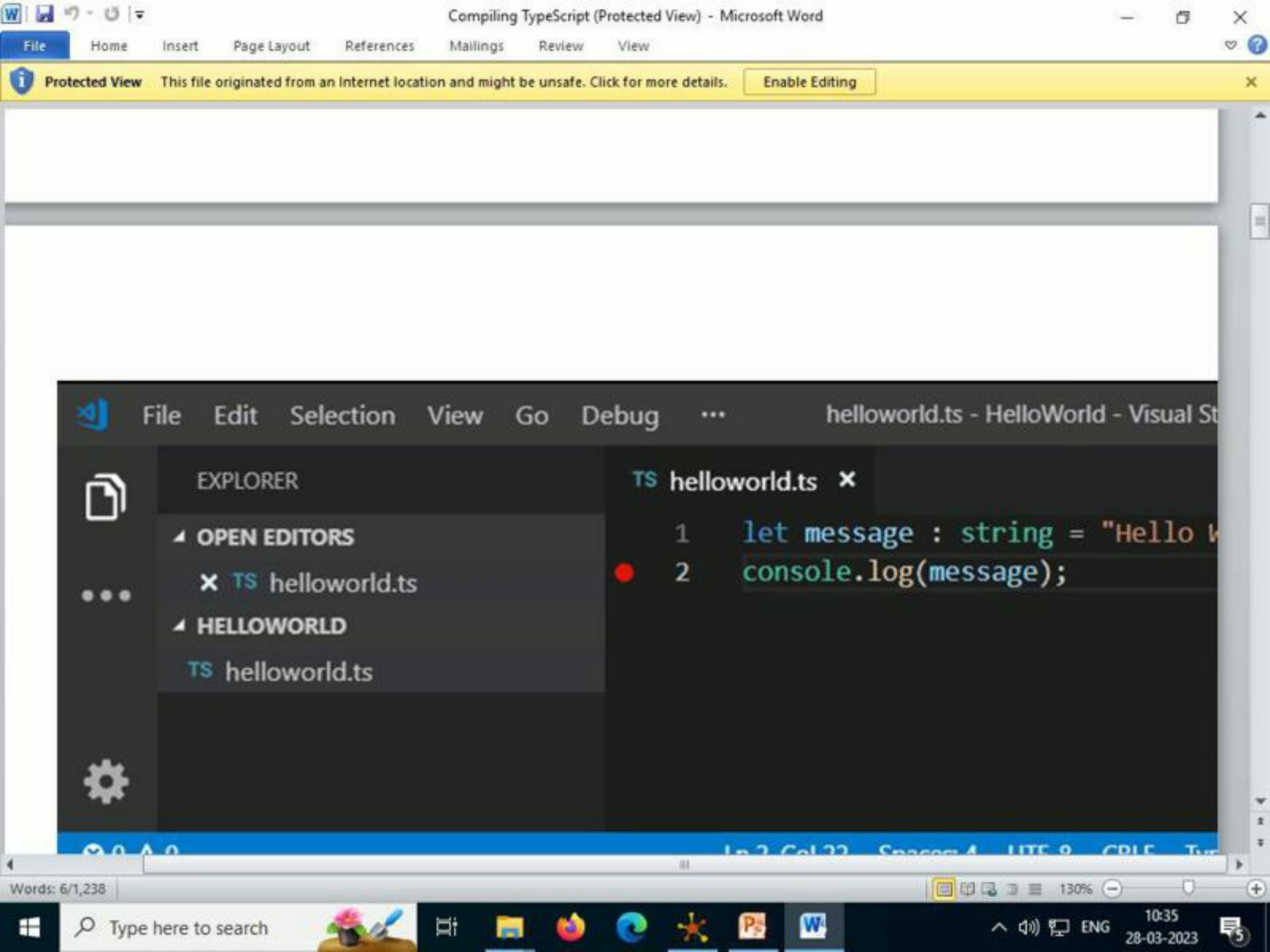
```
tsc --version
```

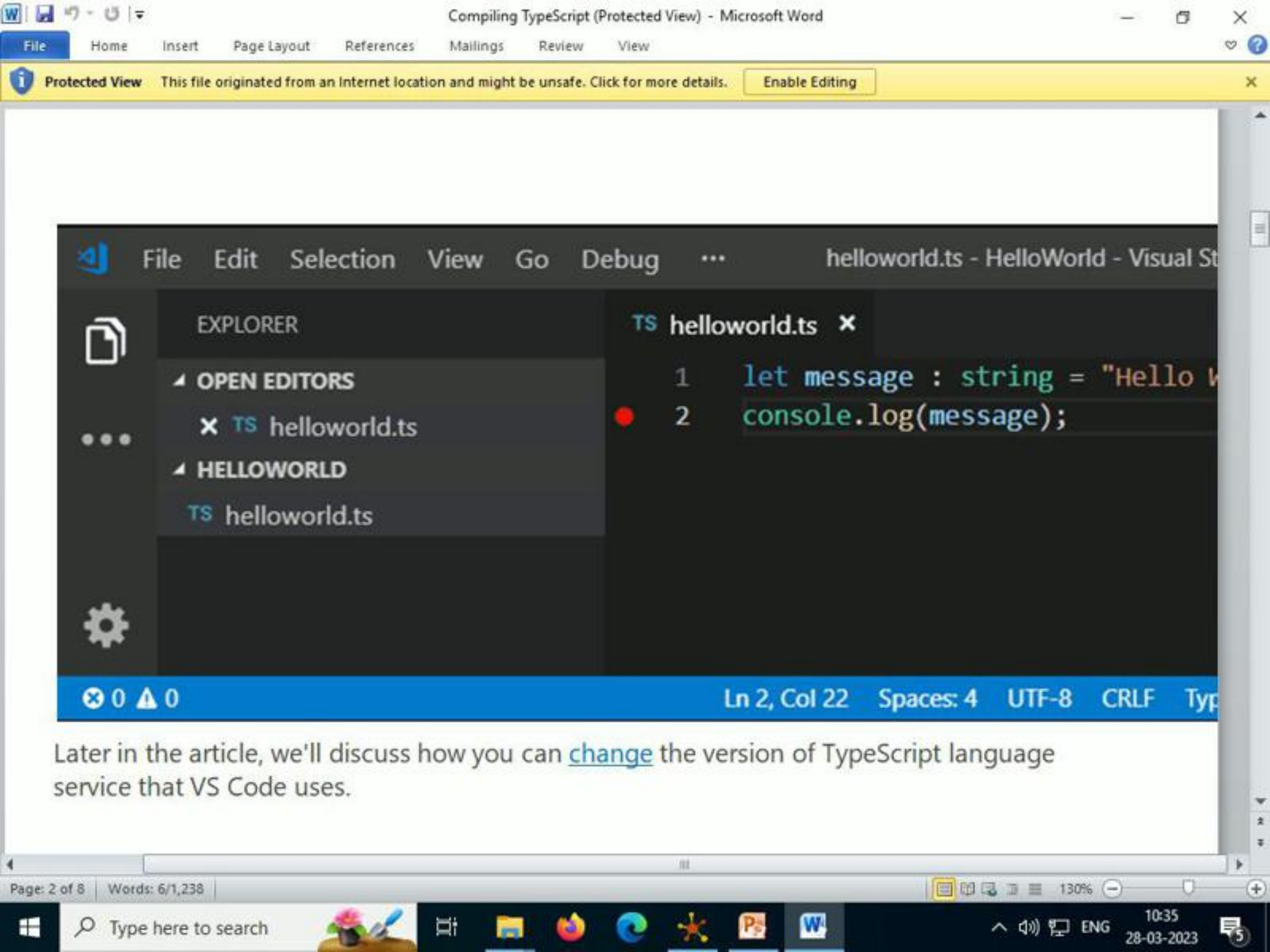
```
tsc --help
```

Another option is to install the TypeScript compiler locally in your project (`npm install --save-dev typescript`) and has the benefit of avoiding possible interactions with other TypeScript projects you may have.

Compiler versus language service

It is important to keep in mind that VS Code's TypeScript language service is separate from your installed TypeScript compiler. You can see the VS Code's TypeScript version in the Status Bar when you open a TypeScript file.





File Edit Selection View Go Debug ... helloworld.ts - HelloWorld - Visual S



EXPLORER

OPEN EDITORS

x TS helloworld.ts

HELLOWORLD

TS helloworld.ts



TS helloworld.ts x

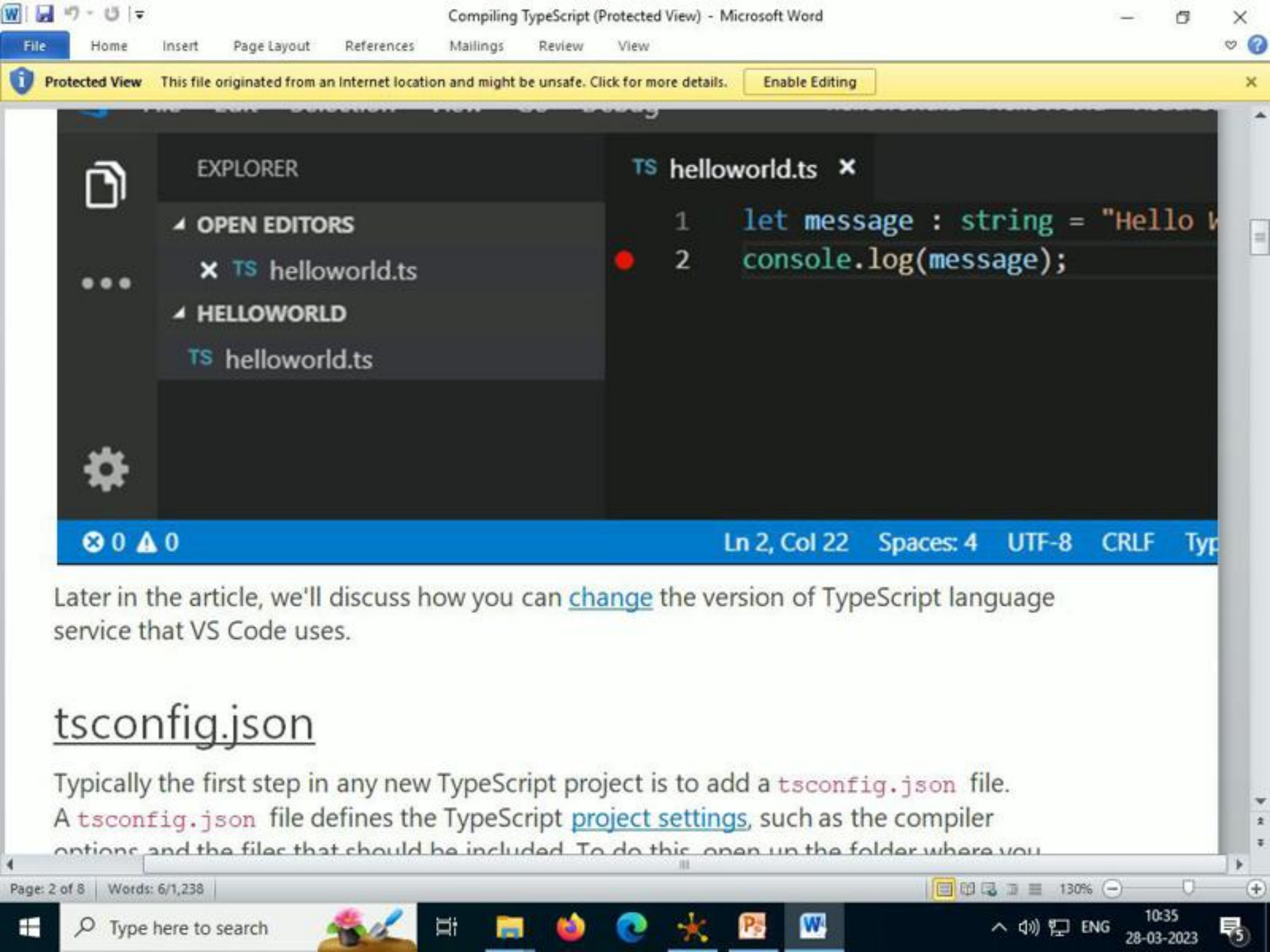
```
1 let message : string = "Hello W
2 console.log(message);
```

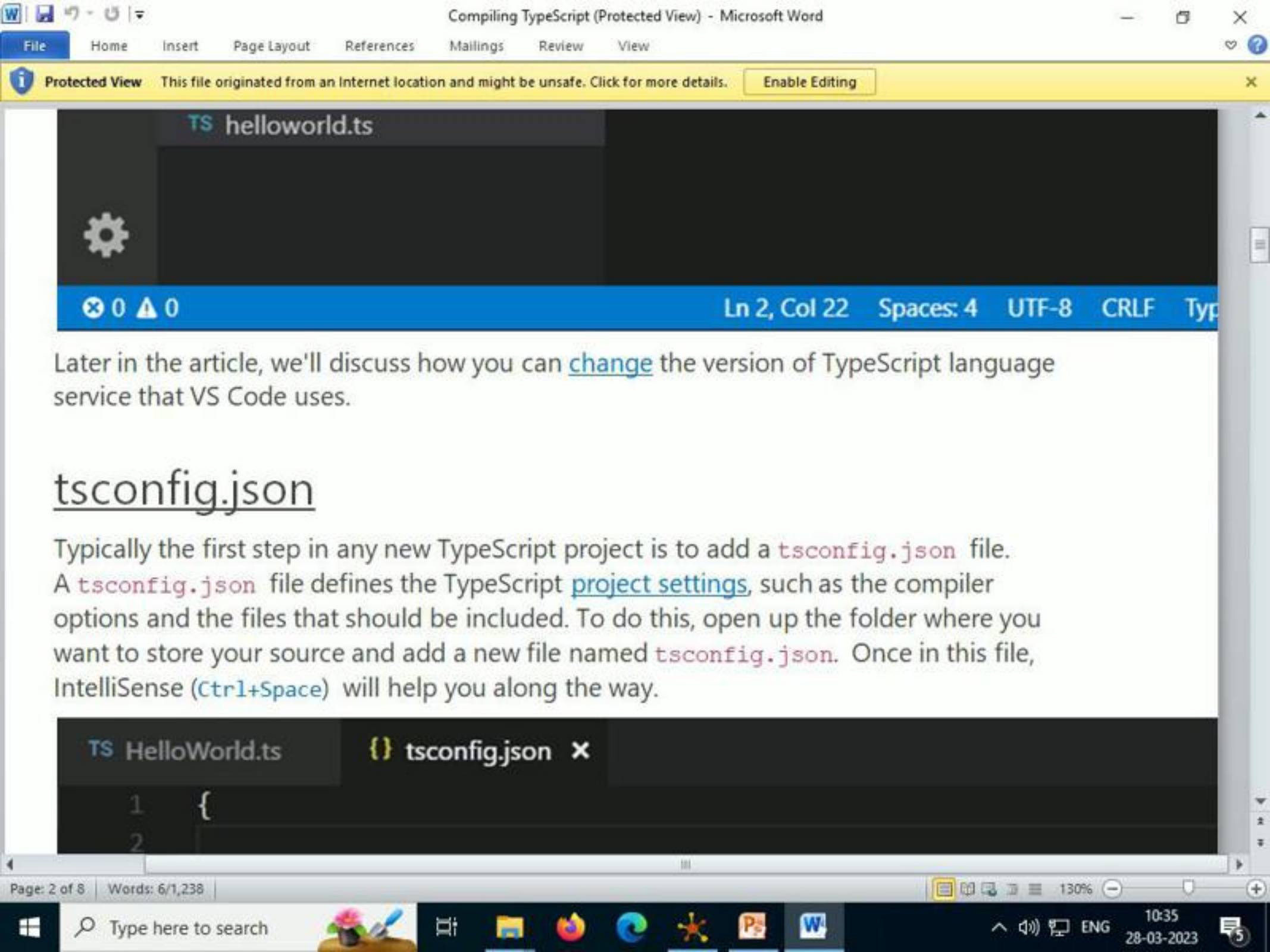
0 0

Ln 2, Col 22 Spaces: 4 UTF-8 CRLF Typ

Later in the article, we'll discuss how you can [change](#) the version of TypeScript language service that VS Code uses.







TS helloworld.ts



0 0

Ln 2, Col 22 Spaces: 4 UTF-8 CRLF Typ

Later in the article, we'll discuss how you can [change](#) the version of TypeScript language service that VS Code uses.

tsconfig.json

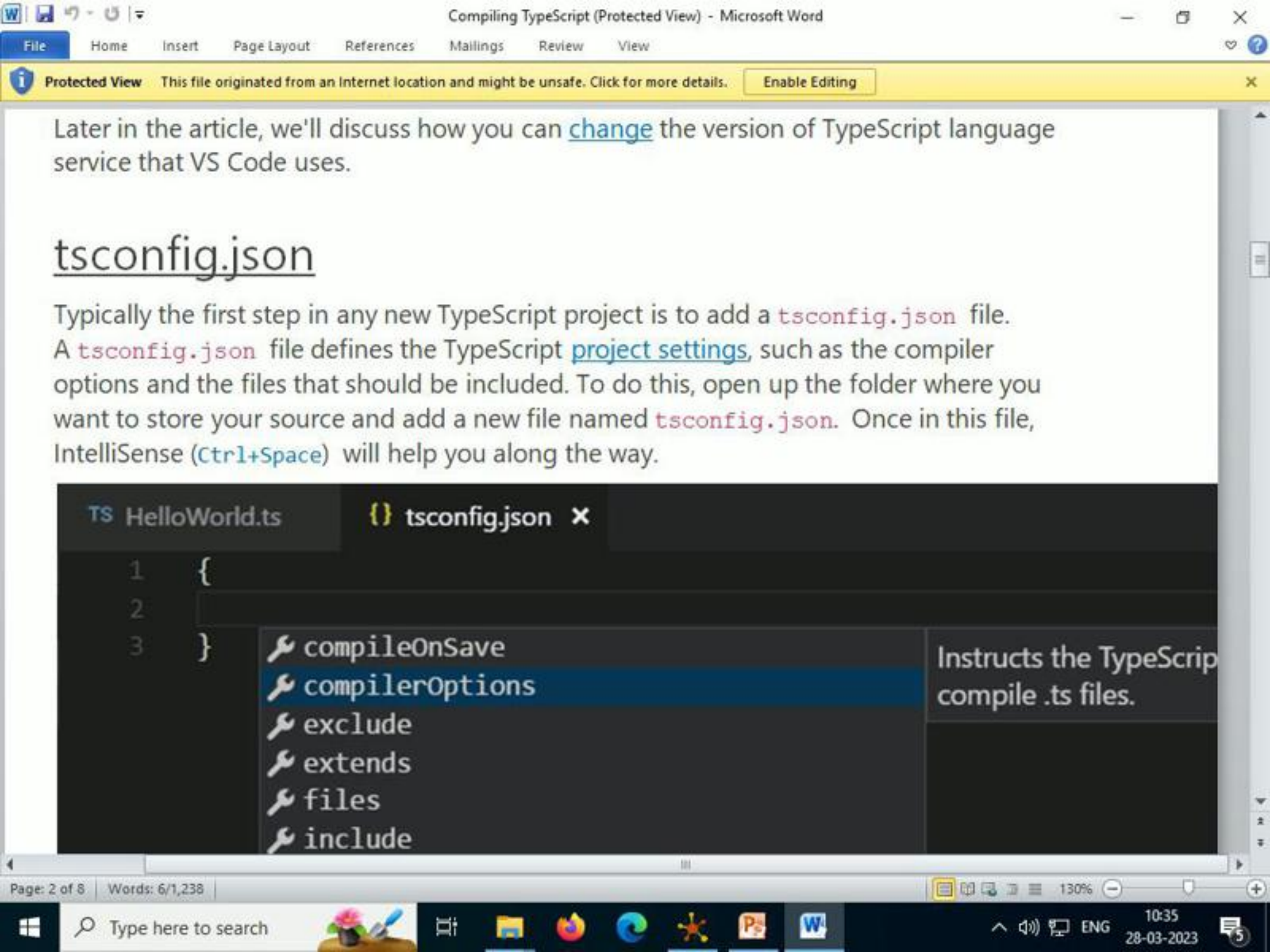
Typically the first step in any new TypeScript project is to add a `tsconfig.json` file. A `tsconfig.json` file defines the TypeScript [project settings](#), such as the compiler options and the files that should be included. To do this, open up the folder where you want to store your source and add a new file named `tsconfig.json`. Once in this file, IntelliSense (`Ctrl+Space`) will help you along the way.

TS HelloWorld.ts

{ } tsconfig.json x

```
1 {  
2
```





Later in the article, we'll discuss how you can [change](#) the version of TypeScript language service that VS Code uses.

tsconfig.json

Typically the first step in any new TypeScript project is to add a `tsconfig.json` file. A `tsconfig.json` file defines the TypeScript [project settings](#), such as the compiler options and the files that should be included. To do this, open up the folder where you want to store your source and add a new file named `tsconfig.json`. Once in this file, IntelliSense (`Ctrl+Space`) will help you along the way.

TS HelloWorld.ts

{ } tsconfig.json X

```
1 {  
2  
3 }
```

- ⚙ compileOnSave
- ⚙ compilerOptions
- ⚙ exclude
- ⚙ extends
- ⚙ files
- ⚙ include

Instructs the TypeScript compile .ts files.

W

File

Home

Insert

Page Layout

References

Mailings

Review

View

Protected View

This file originated from an Internet location and might be unsafe. Click for more details.

Enable Editing

want to store your source and add a new file named `tsconfig.json`. Once in this file, IntelliSense (`Ctrl+Space`) will help you along the way.

TS HelloWorld.ts

1 {

2

3 }

⚙️ compileOnSave

⚙️ compilerOptions

⚙️ exclude

⚙️ extends

⚙️ files

⚙️ include

⚙️ typeAcquisition

Instructs the TypeScript compiler to compile .ts files.

A simple `tsconfig.json` looks like this for ES5, **CommonJS** [modules](#) and source maps:

Page: 2 of 8

Words: 1,238

130%

10:36

28-03-2023

Typically, the first step in any new TypeScript project is to add a `tsconfig.json` file. A `tsconfig.json` file defines the TypeScript [project settings](#), such as the compiler options and the files that should be included. To do this, open up the folder where you want to store your source and add a new file named `tsconfig.json`. Once in this file, IntelliSense ([Ctrl+Space](#)) will help you along the way.

TS HelloWorld.ts
{} tsconfig.json x

```

1  {
2
3  }
```

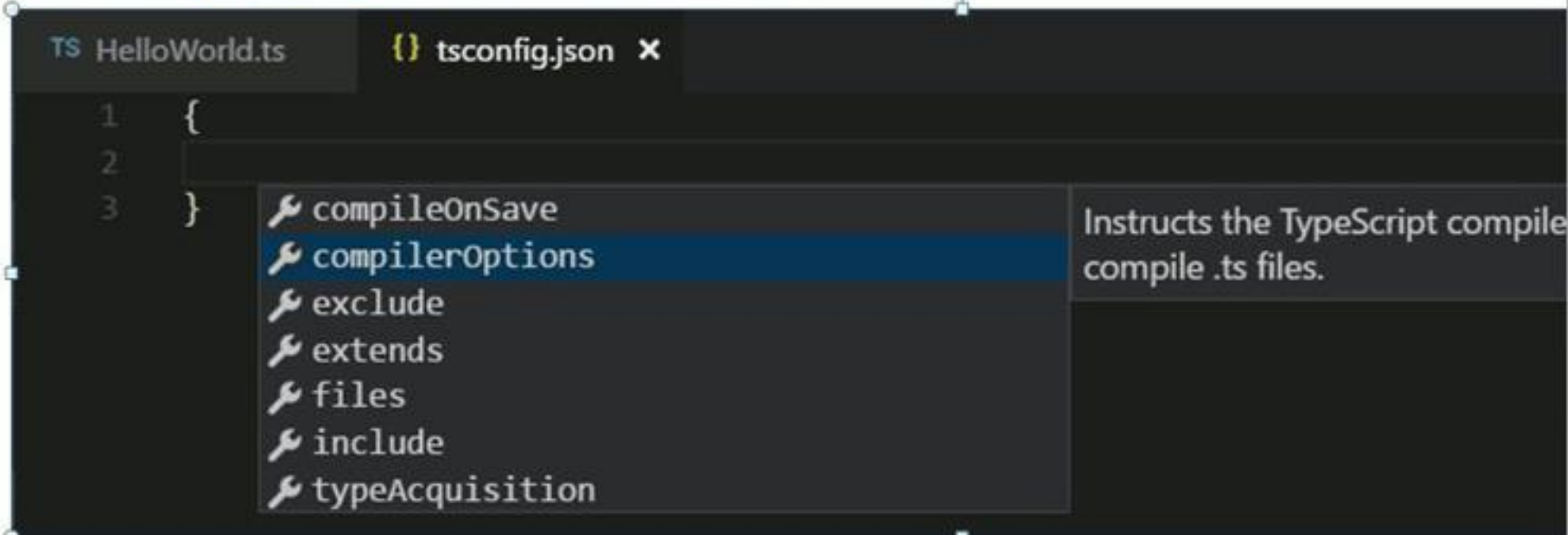
⚙️ compileOnSave
⚙️ compilerOptions
⚙️ exclude
⚙️ extends
⚙️ files
⚙️ include
⚙️ typeAcquisition

Instructs the TypeScript compiler to compile .ts files.

A simple `tsconfig.json` looks like this for ES5, [CommonJS modules](#) and source maps:

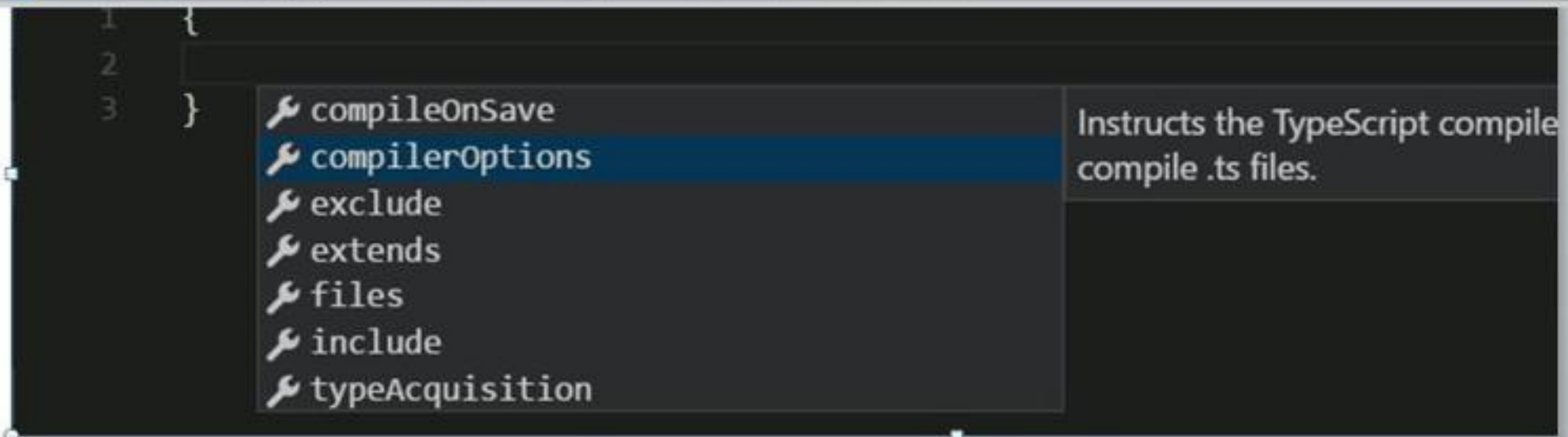
```
{
  "compilerOptions": {
```

A `tsconfig.json` file defines the TypeScript [project settings](#), such as the compiler options and the files that should be included. To do this, open up the folder where you want to store your source and add a new file named `tsconfig.json`. Once in this file, IntelliSense ([Ctrl+Space](#)) will help you along the way.



A simple `tsconfig.json` looks like this for ES5, [CommonJS modules](#) and source maps:

```
{  
  
  "compilerOptions": {  
  
    "target": "es5",
```



A simple [`tsconfig.json`](#) looks like this for ES5, [`CommonJS`](#) [`modules`](#) and source maps:

```
{  
  "compilerOptions": {  
    "target": "es5",  
    "module": "commonjs",  
    "sourceMap": true  
  }  
}
```


Compiling TypeScript [Compatibility Mode] - Microsoft Word

File Home Insert Page Layout References Mailings Review View

compilerOptions

- exclude
- extends
- files
- include
- typeAcquisition

Instructs the TypeScript compiler to compile .ts files.

A simple `tsconfig.json` looks like this for ES5, **CommonJS** `modules` and source maps:

```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "sourceMap": true
  }
}
```

Now when you create a `.ts` file as part of the project we will offer up rich editing


```
"compilerOptions": {  
  "target": "es5",  
  "module": "commonjs",  
  "sourceMap": true  
}
```

Now when you create a `.ts` file as part of the project we will offer up rich editing experiences and syntax validation.

Transpile TypeScript into JavaScript

VS Code integrates with `tsc` through our integrated [task runner](#). We can use this to [transpile](#) `.ts` files into `.js` files. Another benefit of using VS Code tasks is that you get integrated error and warning detection displayed in the [Problems](#) panel. Let's walk through [transpiling](#) a simple [TypeScript](#) Hello World program.

Step 1: Create a simple TS file

Open VS Code on an empty folder and create a `helloworld.ts` file, place the following

```
}  
  
}
```

Now when you create a file as part of the project we will offer up rich editing experiences and

<https://code.visualstudio.com/docs/typescript/typescript-compiling>
Ctrl+Click to follow link

Transpile TypeScript into JavaScript

VS Code integrates with `tsc` through our integrated [task runner](#). We can use this to transpile `.ts` files into `.js` files. Another benefit of using VS Code tasks is that you get integrated error and warning detection displayed in the [Problems](#) panel. Let's walk through transpiling a simple TypeScript Hello World program.

Step 1: Create a simple TS file

Open VS Code on an empty folder and create a `helloworld.ts` file, place the following code in that file...

```
let message: string = 'Hello World';  
  
console.log(message);
```

To test that you have the TypeScript compiler `tsc` installed correctly and a working

Transpile TypeScript into JavaScript

VS Code integrates with `tsc` through our integrated [task runner](#). We can use this to [transpile](#) `.ts` files into `.js` files. Another benefit of using VS Code tasks is that you get integrated error and warning detection displayed in the [Problems](#) panel. Let's walk through [transpiling](#) a simple [TypeScript](#) Hello World program.

Step 1: Create a simple TS file

Open VS Code on an empty folder and create a `helloworld.ts` file, place the following code in that file...

```
let message: string = 'Hello World';  
console.log(message);
```

To test that you have the [TypeScript](#) compiler `tsc` installed correctly and a working Hello World program, open a terminal and type `tsc helloworld.ts`. You can use the Integrated Terminal (`Ctrl+``) directly in VS Code.

You should now see the [transpiled](#) `helloworld.js` JavaScript file, which you can run if you have [Node.js](#) installed, by typing `node helloworld.js`.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: powershell

Integrated error and warning detection displayed in the [Problems](#) panel. Let's walk through [transpiling](#) a simple [TypeScript](#) Hello World program.

Step 1: Create a simple TS file

Open VS Code on an empty folder and create a [helloworld.ts](#) file, place the following code in that file...

```
let message: string = 'Hello World';  
console.log(message);
```

To test that you have the [TypeScript](#) compiler [tsc](#) installed correctly and a working Hello World program, open a terminal and type [tsc helloworld.ts](#). You can use the Integrated Terminal ([Ctrl+`](#)) directly in VS Code.

You should now see the [transpiled helloworld.js](#) JavaScript file, which you can run if you have [Node.js](#) installed, by typing [node helloworld.js](#).

The screenshot shows the VS Code Integrated Terminal with the 'TERMINAL' tab selected. The terminal title is '1: powershell'. The command prompt shows the following sequence of commands and output:

```
PS C:\HelloWorld> tsc HelloWorld.ts  
PS C:\HelloWorld> node HelloWorld.js  
Hello World  
PS C:\HelloWorld>
```

The bottom of the image shows the Windows taskbar with the search bar and several application icons. The system tray in the bottom right corner displays the time as 10:37 and the date as 28-03-2023.

Step 1: Create a simple TS file

Open VS Code on an empty folder and create a `helloworld.ts` file, place the following code in that file...

```
let message: string = 'Hello World';  
console.log(message);
```

To test that you have the TypeScript compiler `tsc` installed correctly and a working Hello World program, open a terminal and type `tsc helloworld.ts`. You can use the Integrated Terminal (`Ctrl+``) directly in VS Code.

You should now see the transpiled `helloworld.js` JavaScript file, which you can run if you have Node.js installed, by typing `node helloworld.js`.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    1: powershell ▼  
  
PS C:\HelloWorld> tsc HelloWorld.ts  
PS C:\HelloWorld> node HelloWorld.js  
Hello World  
PS C:\HelloWorld> |
```

```
let message: string = 'Hello World';  
  
console.log(message);
```

To test that you have the TypeScript compiler tsc installed correctly and a working Hello World program, open a terminal and type tsc helloworld.ts. You can use the Integrated Terminal (Ctrl+`) directly in VS Code.

You should now see the transpiled helloworld.js JavaScript file, which you can run if you have Node.js installed, by typing node helloworld.js.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

1: powershell ▾

```
PS C:\HelloWorld> tsc HelloWorld.ts  
PS C:\HelloWorld> node HelloWorld.js  
Hello World  
PS C:\HelloWorld> █
```

Step 2: Run the TypeScript build

Execute **Run Build Task** (Ctrl+Shift+B) from the global **Terminal** menu. If you created a tsconfig.json file in the earlier section, this should present the following picker:

You should now see the transpiled `helloworld.js` JavaScript file, which you can run if you have Node.js installed, by typing `node helloworld.js`.

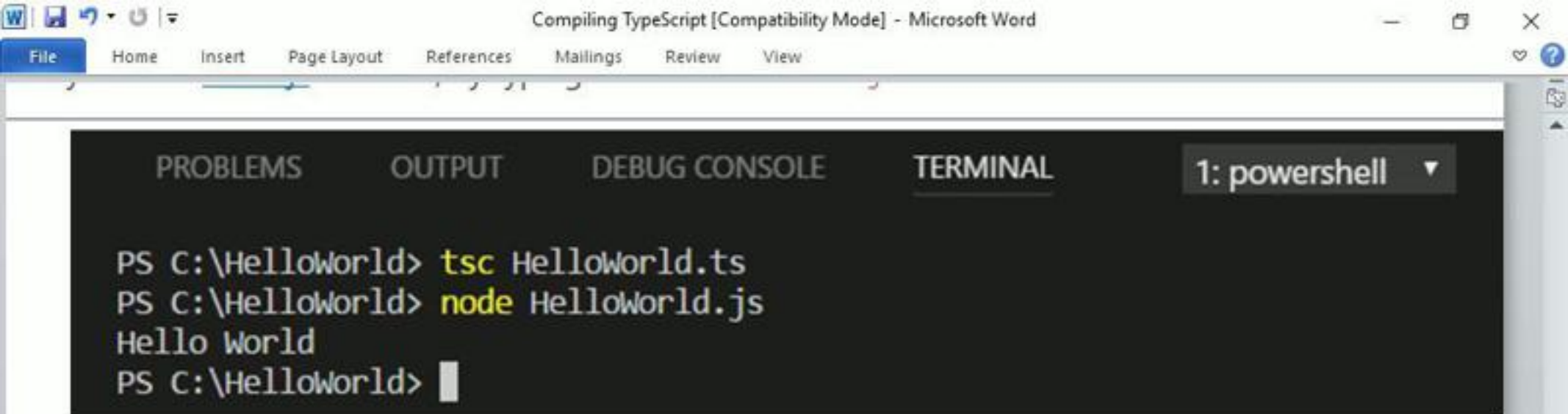
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell ▼  
  
PS C:\HelloWorld> tsc HelloWorld.ts  
PS C:\HelloWorld> node HelloWorld.js  
Hello World  
PS C:\HelloWorld> |
```

Step 2: Run the TypeScript build

Execute **Run Build Task** (Ctrl+Shift+B) from the global **Terminal** menu. If you created a `tsconfig.json` file in the earlier section, this should present the following picker:

```
Select the build task to run  
  
tsc: build - tsconfig.json  
tsc: watch - tsconfig.json
```

Select the **tsc: build** entry. This will produce



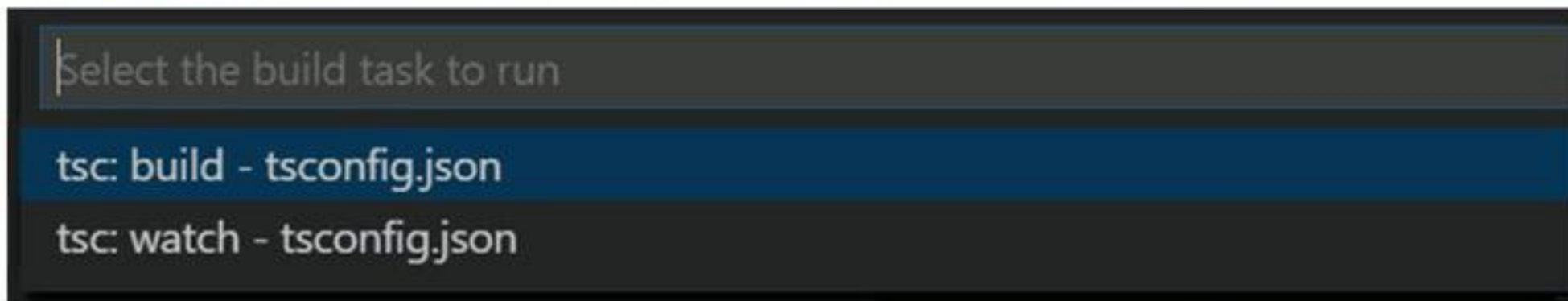
The screenshot shows a Visual Studio Code window with the title bar "Compiling TypeScript [Compatibility Mode] - Microsoft Word". The interface includes a menu bar with "File", "Home", "Insert", "Page Layout", "References", "Mailings", "Review", and "View". Below the menu bar is a toolbar. The main area is a dark-themed terminal window with tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "TERMINAL" tab is active, showing a PowerShell session with the following commands and output:

```
PS C:\HelloWorld> tsc HelloWorld.ts
PS C:\HelloWorld> node HelloWorld.js
Hello World
PS C:\HelloWorld> |
```

The terminal window also shows a dropdown menu for "1: powershell" in the top right corner.

Step 2: Run the TypeScript build

Execute **Run Build Task** ([Ctrl+Shift+B](#)) from the global **Terminal** menu. If you created a [tsconfig.json](#) file in the earlier section, this should present the following picker:



Select the **tsc: build** entry. This will produce a [HelloWorld.js](#) and [HelloWorld.js.map](#) file in the workspace.

If you selected **tsc: watch**, the TypeScript compiler watches for changes to your

```
PS C:\HelloWorld> node HelloWorld.js
Hello World
PS C:\HelloWorld> 
```

Step 2: Run the TypeScript build

Execute **Run Build Task** (Ctrl+Shift+B) from the global **Terminal** menu. If you created a tsconfig.json file in the earlier section, this should present the following picker:

```
Select the build task to run

tsc: build - tsconfig.json
tsc: watch - tsconfig.json
```

Select the tsc: build entry. This will produce a HelloWorld.js and HelloWorld.js.map file in the workspace.

If you selected tsc: watch, the TypeScript compiler watches for changes to your TypeScript files and runs the transpiler on each change.

Under the covers, we run the TypeScript compiler as a task. The command we use is: tsc -p .

PS C:\HelloWorld>

Step 2: Run the TypeScript build

Execute **Run Build Task** (Ctrl+Shift+B) from the global **Terminal** menu. If you created a tsconfig.json file in the earlier section, this should present the following picker:

Select the build task to run

tsc: build - tsconfig.json

tsc: watch - tsconfig.json

Select the tsc: build entry. This will produce a HelloWorld.js and HelloWorld.js.map file in the workspace.

If you selected tsc: watch, the TypeScript compiler watches for changes to your TypeScript files and runs the transpiler on each change.

Under the covers, we run the TypeScript compiler as a task. The command we use is: tsc -p .

Step 3: Make the TypeScript Build the default

You can also define the TypeScript build task as the default build task so that it is

Select the tsc: build entry. This will produce a HelloWorld.js and HelloWorld.js.map file in the workspace.

If you selected tsc: watch, the TypeScript compiler watches for changes to your TypeScript files and runs the transpiler on each change.

Under the covers, we run the TypeScript compiler as a task. The command we use is: tsc -p .

Step 3: Make the TypeScript Build the default

You can also define the TypeScript build task as the default build task so that it is executed directly when triggering **Run Build Task** (Ctrl+Shift+B). To do so, select **Configure Default Build Task** from the global **Terminal** menu. This shows you a picker with the available build tasks. Select TypeScript tsc: build, which generates the following tasks.json file in a .vscode folder:

```
{  
  
  // See https://go.microsoft.com/fwlink/?LinkId=733558  
  // for the documentation about the tasks.json format  
  
  "version": "2.0.0",  
}
```

Under the covers, we run the TypeScript compiler as a task. The command we use is, tsc -p .

Step 3: Make the TypeScript Build the default

You can also define the TypeScript build task as the default build task so that it is executed directly when triggering **Run Build Task** (Ctrl+Shift+B). To do so, select **Configure Default Build Task** from the global **Terminal** menu. This shows you a picker with the available build tasks. Select TypeScript tsc: build, which generates the following tasks.json file in a .vscode folder:

```
{  
  
  // See https://go.microsoft.com/fwlink/?LinkId=733558  
  // for the documentation about the tasks.json format  
  "version": "2.0.0",  
  
  "tasks": [  
    {  
      "type": "typescript",  
      "tsconfig": "tsconfig.json",  
    }  
  ]  
}
```

{

// See <https://go.microsoft.com/fwlink/?LinkId=733558>

// for the documentation about the tasks.json format

"version": "2.0.0",

"tasks": [
 {

{

"type": "typescript",

"tsconfig": "tsconfig.json",

"problemMatcher": [
 "\$tsc"

],

"group": {

"kind": "build",

"isDefault": true




```
// See https://go.microsoft.com/fwlink/?LinkId=733558  
// for the documentation about the tasks.json format  
"version": "2.0.0",
```

```
"tasks": [  
  {  
    "type": "typescript",  
    "tsconfig": "tsconfig.json",  
    "problemMatcher": [  
      "$tsc"  
    ],  
    "group": {  
      "kind": "build",  
      "isDefault": true  
    }  
  }  
]
```

```
{  
  "type": "typescript",  
  "tsconfig": "tsconfig.json",  
  "problemMatcher": [  
    "$tsc"  
  ],  
  "group": {  
    "kind": "build",  
    "isDefault": true  
  }  
}
```

Notice that the task has a **group** JSON object that sets the task **kind** to **build** and

```
"type": "typescript",  
"tsconfig": "tsconfig.json",  
"problemMatcher": [  
    "$tsc"  
],  
"group": {  
    "kind": "build",  
    "isDefault": true  
}  
}  
]  
}
```

Notice that the task has a **group** JSON object that sets the task **kind** to **build** and makes it the default. Now when you select the **Run Build Task** command or press ([Ctrl+Shift+B](#)), you are not prompted to select a task and your compilation starts.