COMPANY:-INVICTUS

NAME:-ADARSH GOUR

EDUCATION:-B-TECH CSE

SEMESTER:-6th

UNIVERSITY:-LOVELY PROFESSIONAL UNIVERSITY

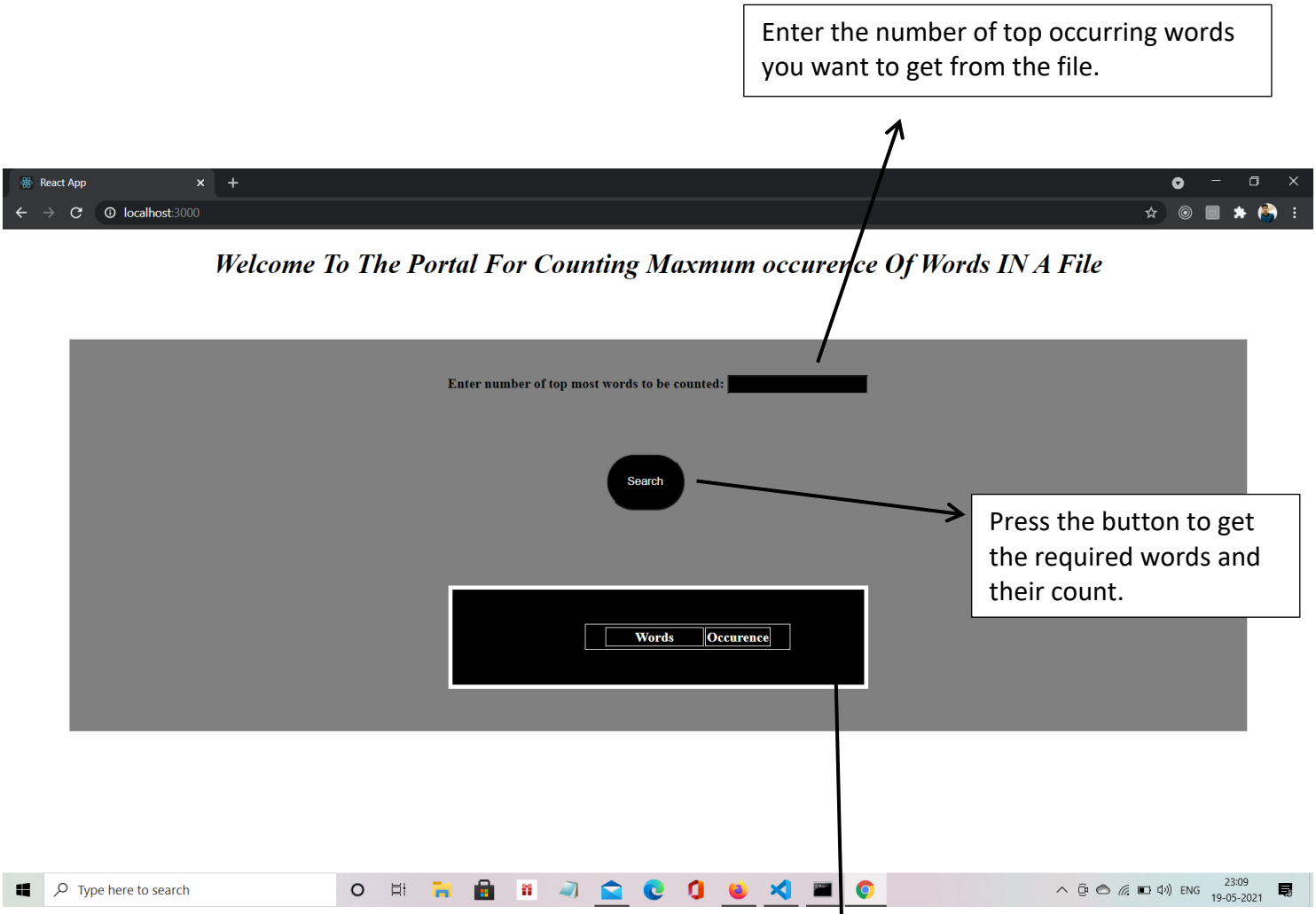
REG.NO:- 11806478

MAIL ID:- rudransh3067@gmail.com

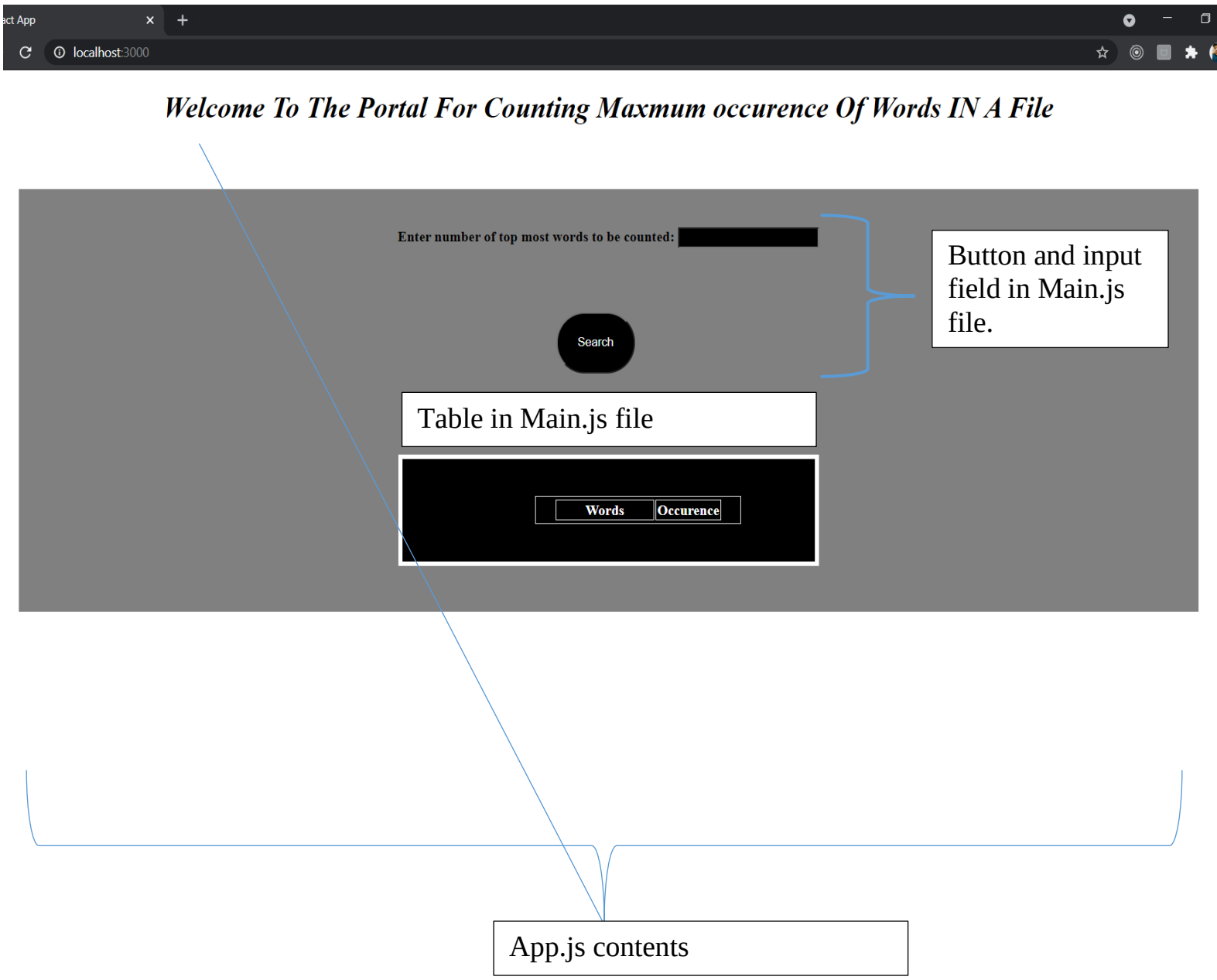
TASK:- TO CREATE A REACT BASED APPLICATION FOR COUNTING THE ICCURENCE OF PARTICULAR WORD AND FETCHING IT AND DISPLAYING IT ON CALL SCREEN.

Introduction:-

Hello there ! Let me take to the portal.



Components:-

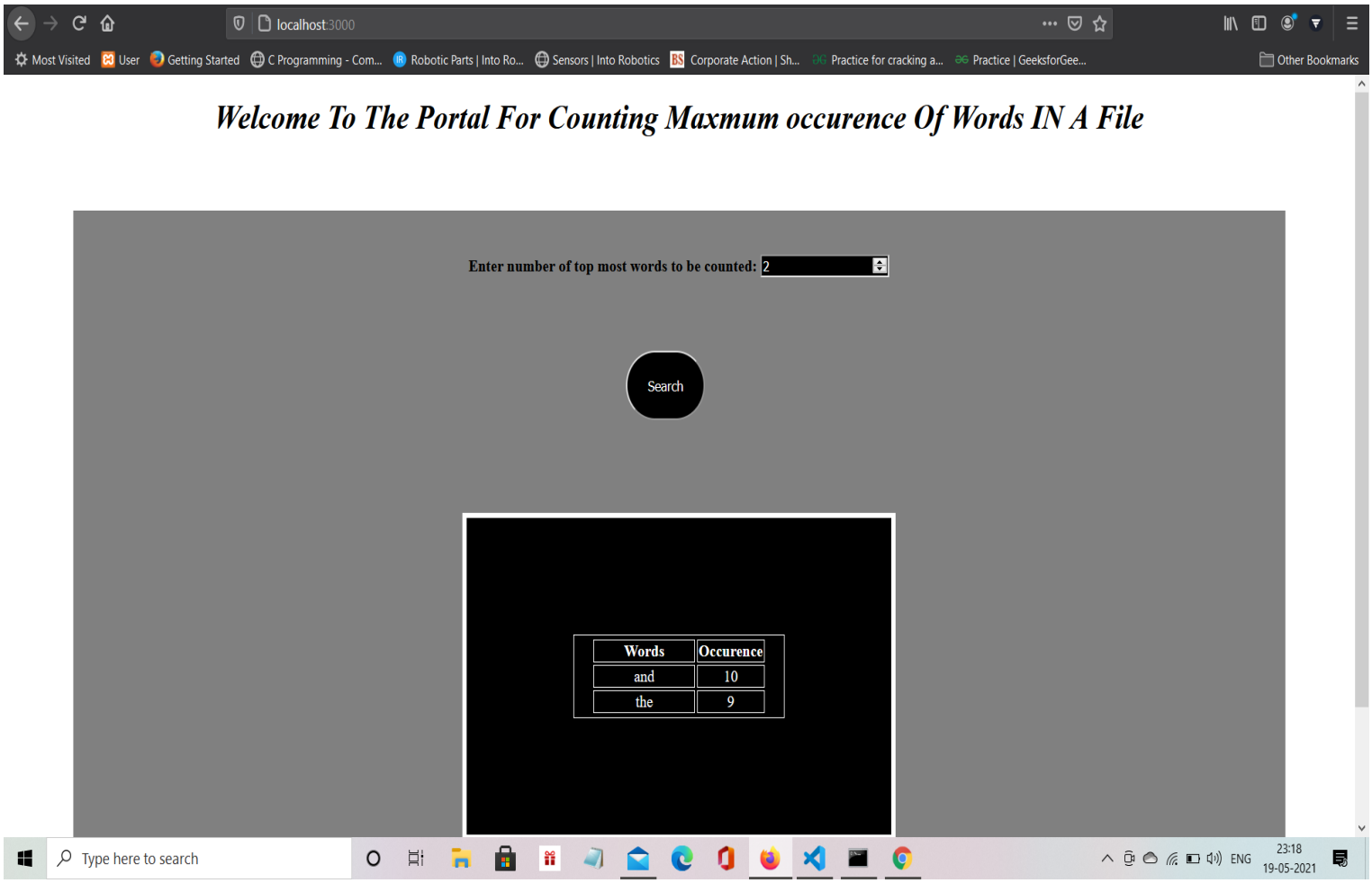

act App
localhost:3000
Welcome To The Portal For Counting Maxmum occurence Of Words IN A File
Enter number of top most words to be counted:
Search
Button and input field in Main.js file.
Table in Main.js file
Words
Occurence
App.js contents

- Onto input of any number in the input field acceptance.
- On clicking the “Search” button getting to the content file and fetching content.
- Deleting special character and spaces.
- Catching Words and counting occurrence.
- Sorting words in descending order of occurrence.
- Creating list of these keys and values of words and occurrence.
- Displaying the number of words from this list requested by user in input field.

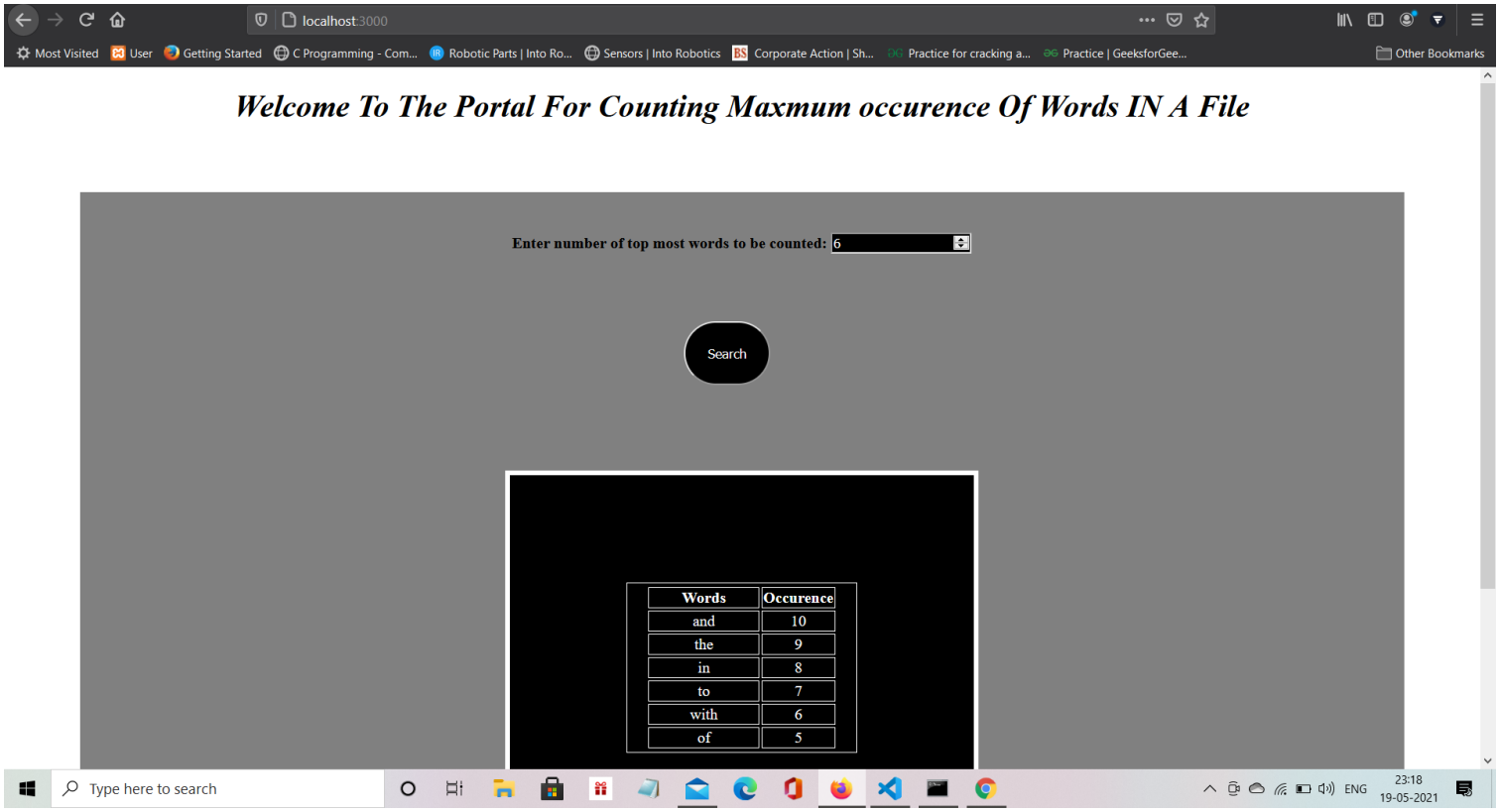
Usages:-

- .tostring()
- .toLowerCase()
- .replace()
- .split()
- parseInt
- Fetch()
- .then()
- .isNAN()
- .sort()
- Object.fromEntries()
- Object.entries()
- Object.keys()
- .map([keys,values])

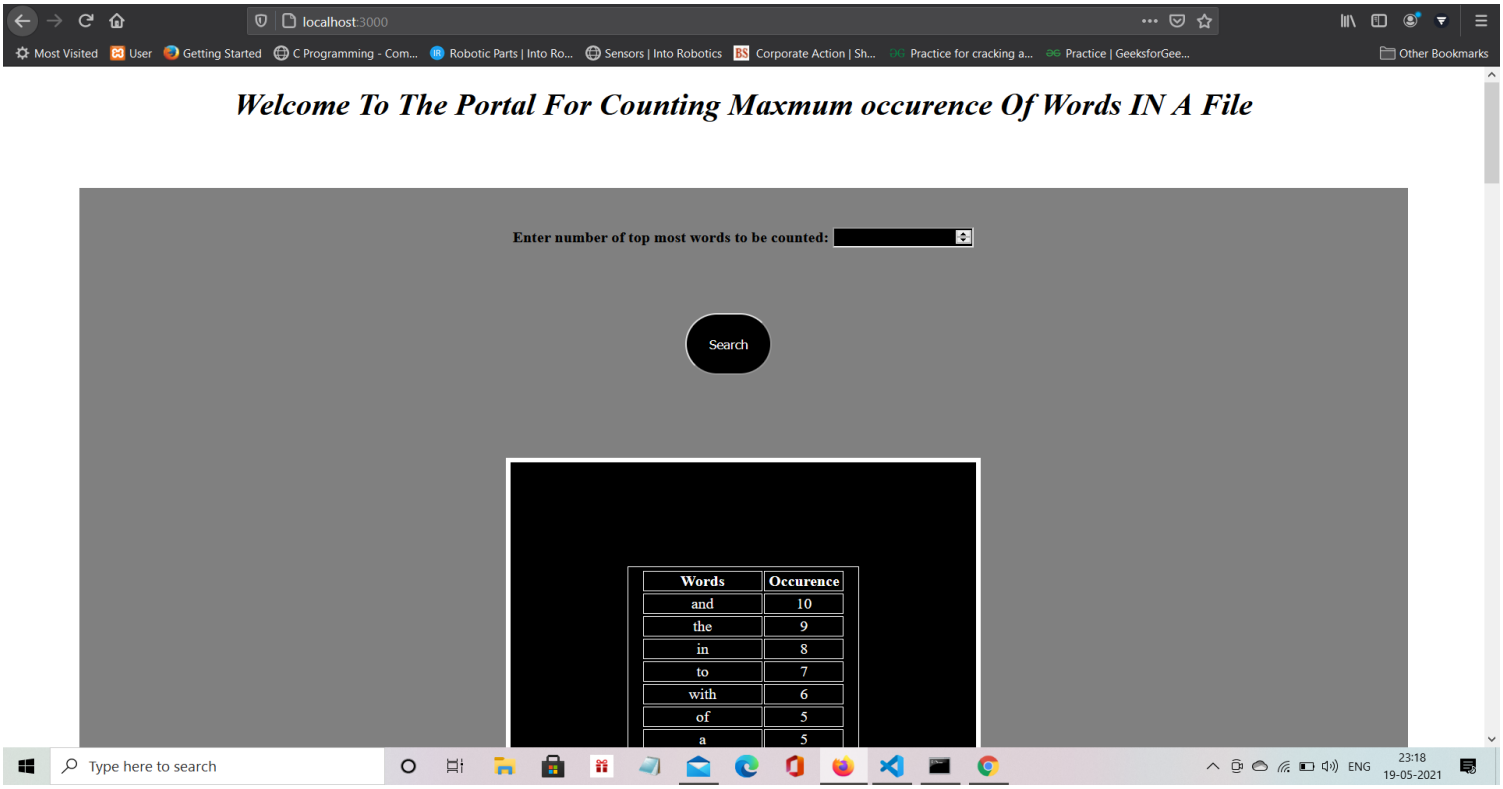
Testcases



On passing 2 in input field we get the output of two top most occurring words (i.e “and -10 & the - 9 ”)



Here we passed - 6 as input field giving top 6 occurring words from the file.



If we enter nothing in the input field and go for search- by default it will give the whole list of words and occurrences.

App.js:-

```
import Main from './Main'


function App() {
    return ( <
        div className = "App" >
```

```
        <
        h1 style = {
            {
                color: "black",
                fontStyle: "italic"
            }
        } > < center > Welcome To The Portal For Counting Maxmum occurence Of Words IN A File
< /center > < /
        h1 > <
        Main / >
        <
        /div>
    );
}
```

```
Export default App;
```

Main.js :-

```
import React, { useState } from 'react'

function Main() {
    const url = 'https://raw.githubusercontent.com/invictustech/test/main/README.md'
    const [data, setData] = useState([]);
```

```
    const [n, f] = useState([]);
    const apiGet = () => {
        fetch(url)
            .then((response) => response.text())
            .then((text) => {
                setData(text);
            });
        f(document.getElementById("input").value);
    }
```

```
    const correction = data.toString().toLowerCase().replace(/  [^a-zA-Z ]/g, "").split(" ");
    const output = []
```

```
    //occurrence of each word
    for (var i = 0; i < correction.length; i++) {
        var word = correction[i];
        if (isNaN(word)) {
            if (output[word] === undefined) {
```

```
                output[word] = 1;
            } else {
                output[word] += 1;
            }
        }
    }
```

```
    var a = parseInt(n);
    //sorting  in decreasing order
    const sortable = Object.fromEntries(
        Object.entries(output).sort(([, a], [, b]) => b - a));
```

```
    //creating object of keys and values(words and occurrences)
    const pp = []
    var c = 0;
```

```
    for (const e of Object.keys(sortable)) {
        if (c !== a) {
            pp[e] = sortable[e];
            c += 1;
        } else {
            break;
        }
    }
```

```
    const s1 = {
        backgroundColor: "black",
        color: "white",
        width: "30%",
        border: "5px solid white",
        padding: "40px",
        margin: "30px",
    };
```

```
    const s2 = {
```

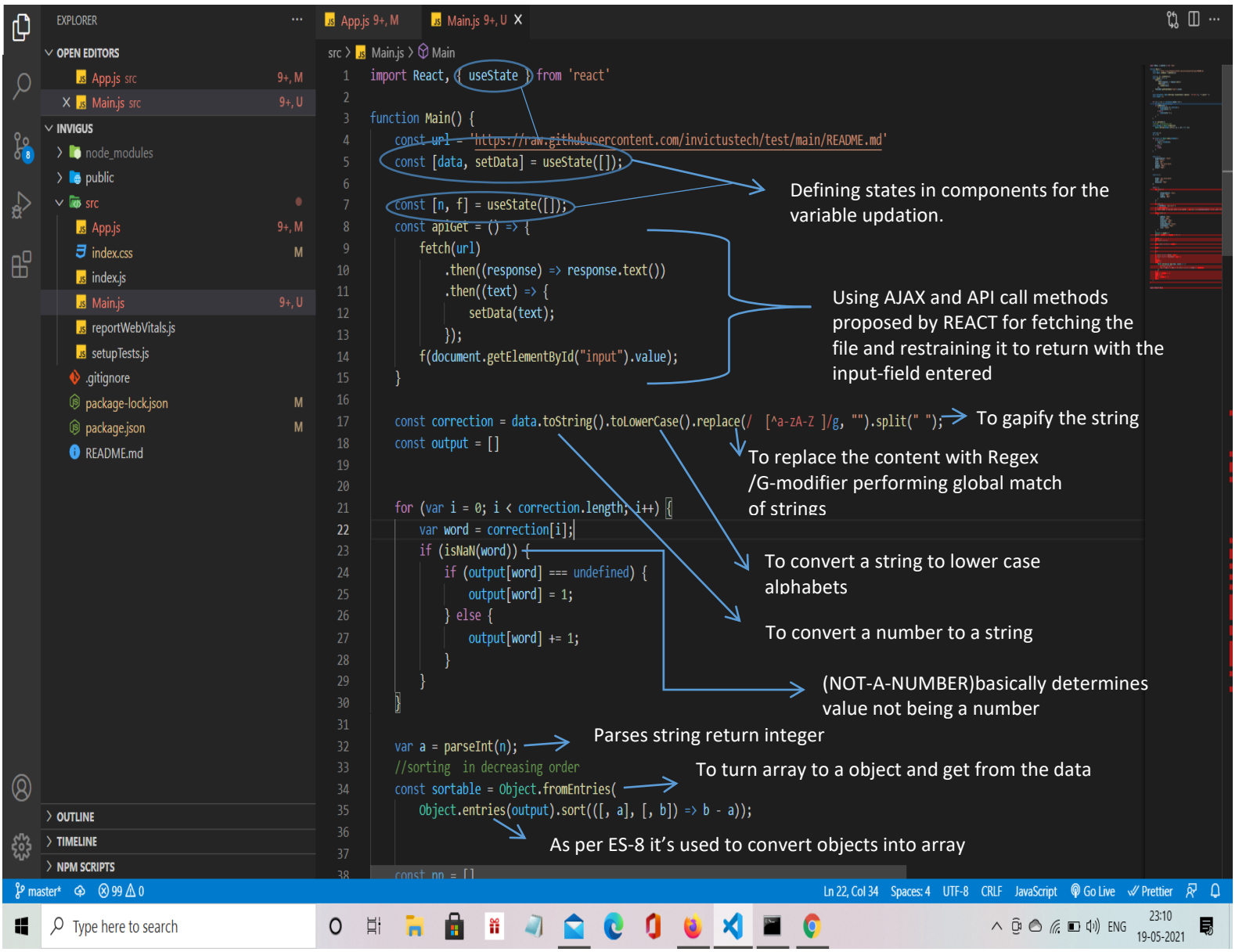
```
        border: "1px solid white",
        width: "60%",
        margin: "70px"
    };
```

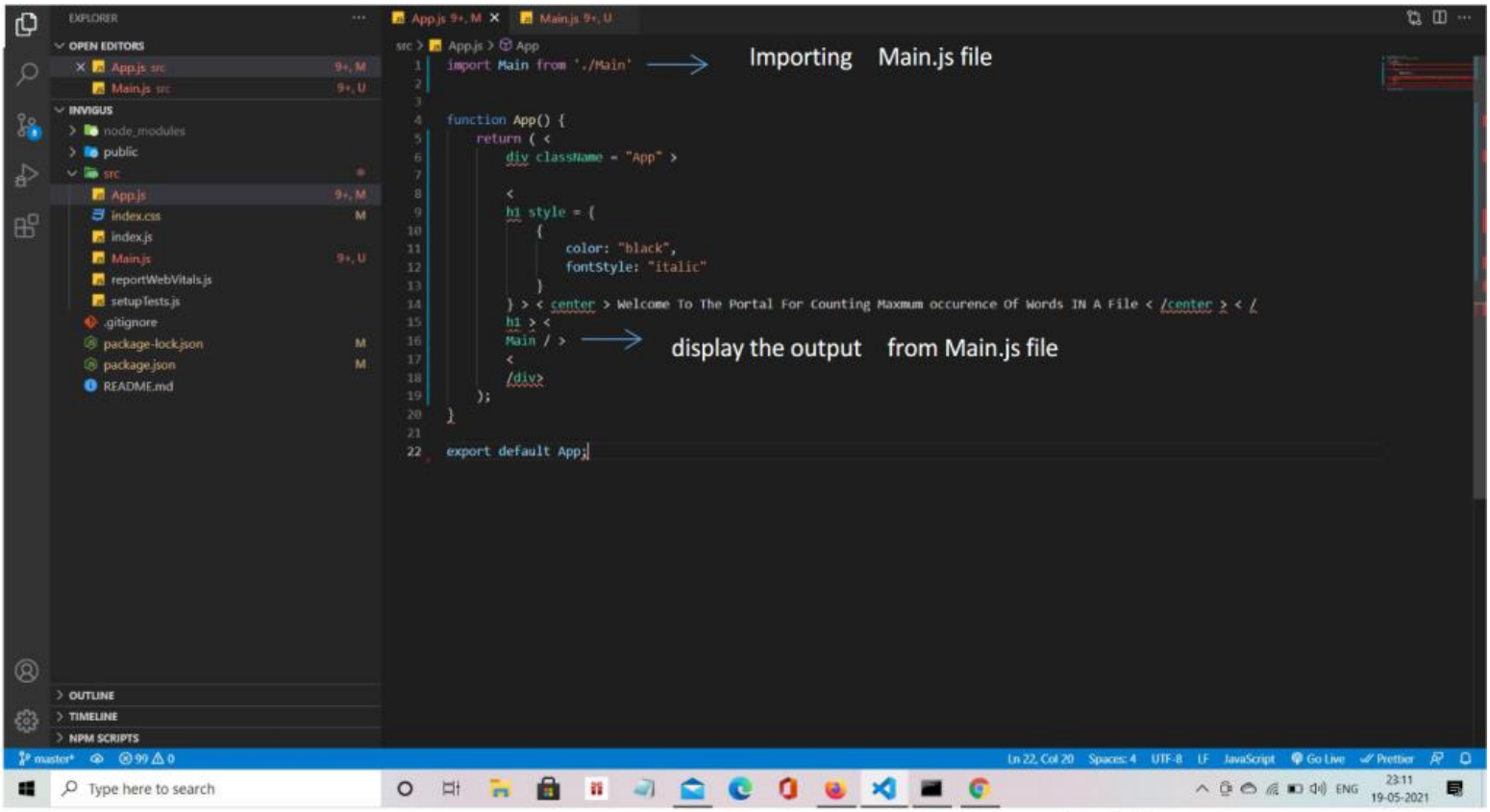
```
    return ( <
        div style = {
            {
                backgroundColor: "grey",
                margin: "70px",
                padding: "20px"
            }
        } >
        <
        h1 style = {
            { fontFamily: "sans-serif" }
        } > < /h1><center> <
        b > Enter number of top most words to be counted: < /b><input style={{ backgroundColor
: "black", color: "white", }} type="number" id="input" name="number"></input > < br / >
        <
        button style = {
            {
```

```
                padding: "23px",
                margin: "40px",
                marginLeft: "10px",
                marginTop: "70px",
                backgroundColor: "black",
                borderRadius: "70px",
                color: "white"
            }
        }
        onClick = { apiGet } >
        Search < /button > < /center > < br / >
        <
        center > <
        div style = { s1 } >
        <
        table style = { s2 } > < center >
        <
        thead >
        <
        tr >
        <
        th style = { s2 } > Words < /th> <
        th style = { s2 } > Occurence < /th> < /
        tr > <
        /thead> <
        tbody > {
            Object.entries(pp).map(([key, value]) => ( <
                tr style = { s2 }
                key = { key } > < td style = { s2 } > { key } < /td> < td style={s2} > {value}
 < /td > < /tr >
            ))
        } < /
        tbody > < /center > < /
        table > < /
        div > < /center > < /
        div >
    )
}
```

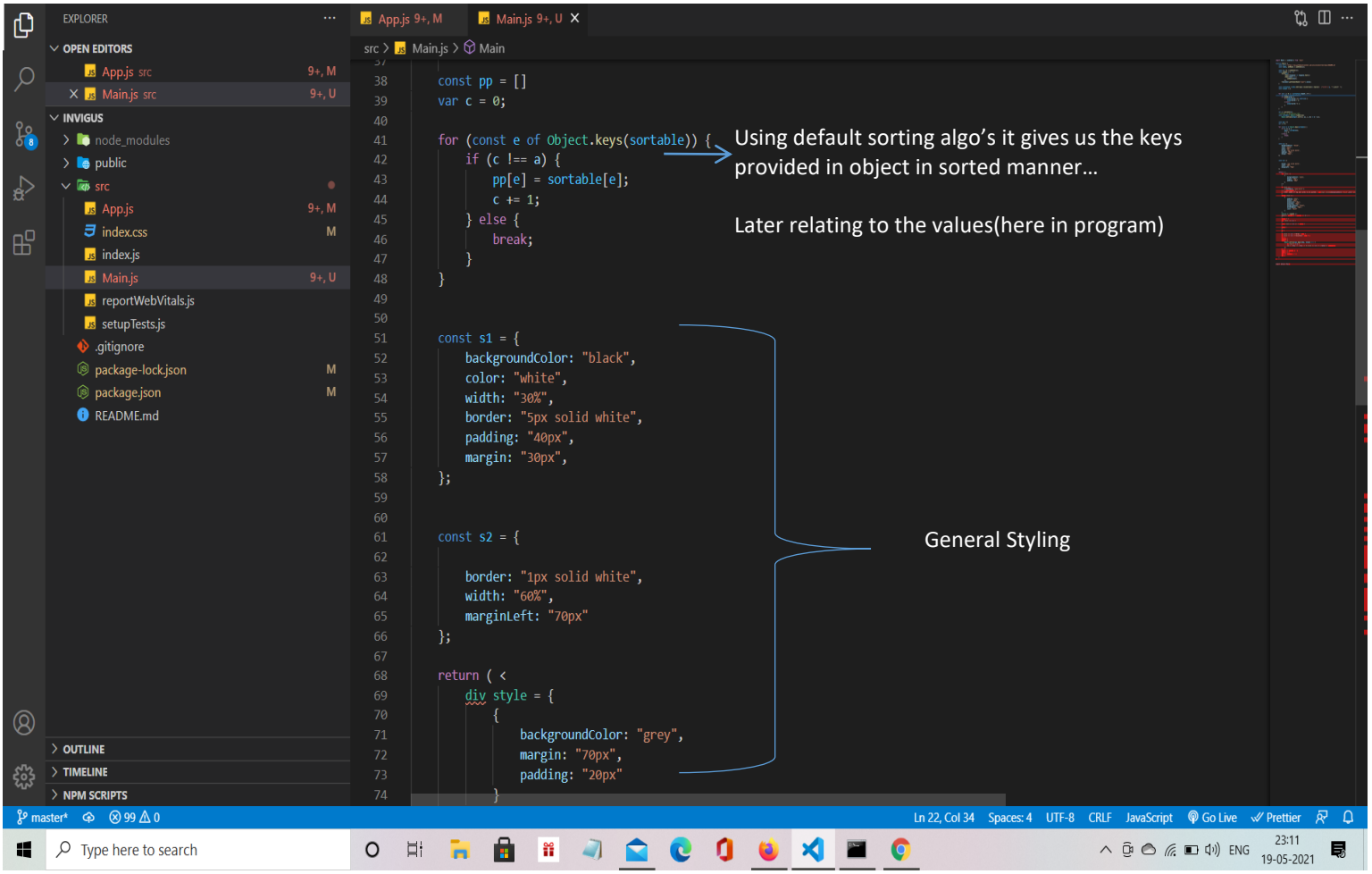
```
export default Main;
```

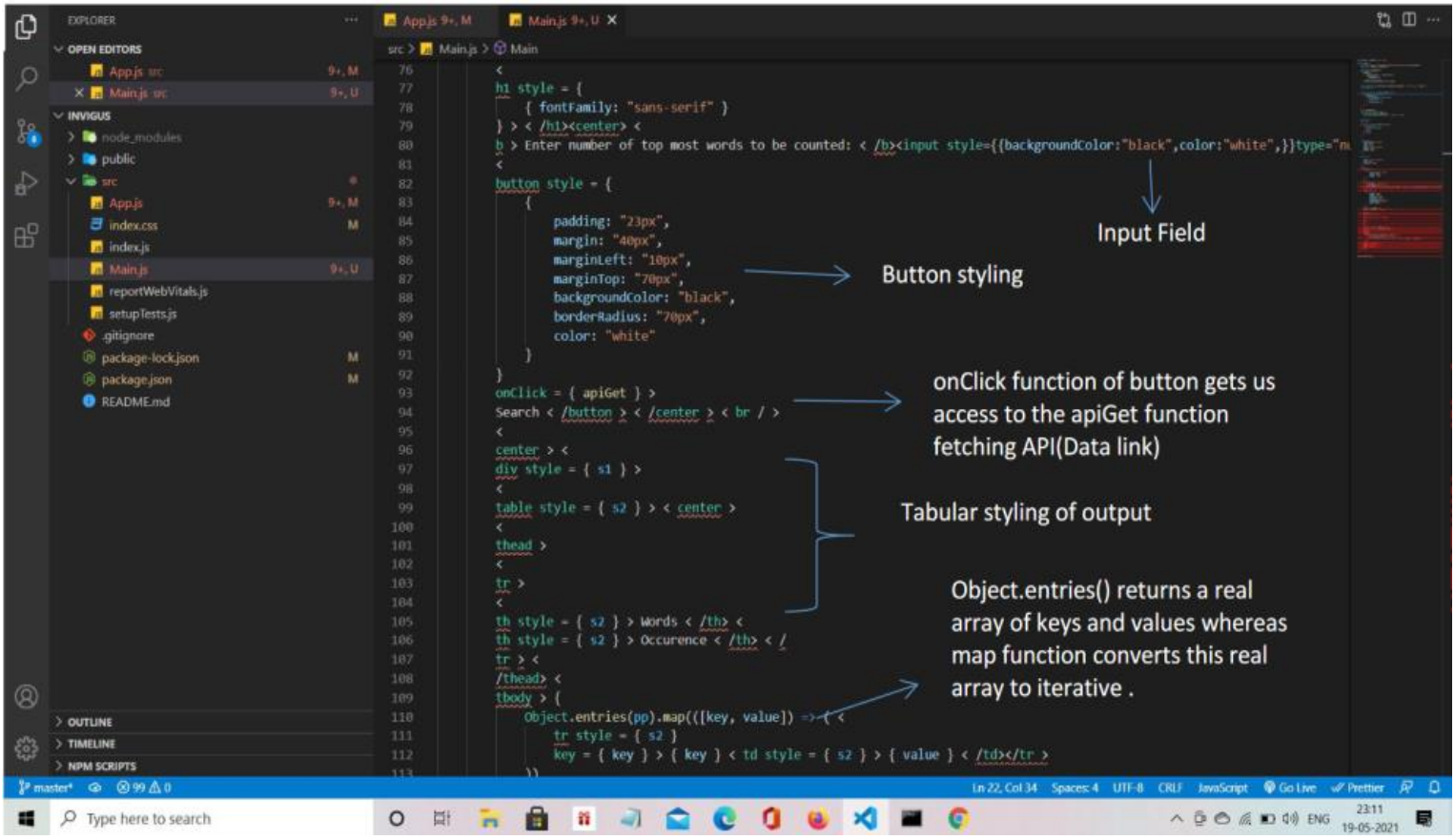
Code :-


EXPLORER
OPEN EDITORS
App.js src 9+, M
Main.js src 9+, U
INVIGUS
node_modules
public
src
App.js 9+, M
index.css M
index.js
Main.js 9+, U
reportWebVitals.js
setupTests.js
.gitignore
package-lock.json M
package.json M
README.md
OUTLINE
TIMELINE
NPM SCRIPTS
App.js 9+, M
Main.js 9+, U
src > Main.js > Main
import React, { useState } from 'react'
function Main() {
const url = 'https://raw.githubusercontent.com/invictustech/test/main/README.md'
const [data, setData] = useState([]);
const [n, f] = useState([]);
const apiGet = () => {
fetch(url)
.then((response) => response.text())
.then((text) => {
setData(text);
});
f(document.getElementById("input").value);
}
const correction = data.toString().toLowerCase().replace(/ [^a-zA-Z]/g, "").split(" ");
const output = []
for (var i = 0; i < correction.length; i++) {
var word = correction[i];
if (isNaN(word)) {
if (output[word] === undefined) {
output[word] = 1;
} else {
output[word] += 1;
}
}
}
var a = parseInt(n);
//sorting in decreasing order
const sortable = Object.fromEntries(
Object.entries(output).sort(([, a], [, b]) => b - a));
Defining states in components for the variable updation.
Using AJAX and API call methods proposed by REACT for fetching the file and restraining it to return with the input-field entered
To gapify the string
To replace the content with Regex /G-modifier performing global match of strings
To convert a string to lower case alphabets
To convert a number to a string
(NOT-A-NUMBER)basically determines value not being a number
Parses string return integer
To turn array to a object and get from the data
As per ES-8 it's used to convert objects into array
master*
Ln 22, Col 34
Spaces: 4
UTF-8
CRLF
JavaScript
Go Live
Prettier
Type here to search
ENG
23:10
19-05-2021



App.js file


Using default sorting algo's it gives us the keys provided in object in sorted manner...
Later relating to the values(here in program)
General Styling


Input Field
Button styling
onClick function of button gets us access to the apiGet function fetching API(Data link)
Tabular styling of output
Object.entries() returns a real array of keys and values whereas map function converts this real array to iterative .