1) under calc.js write 2 functions -1 to calculate product of 2 sumbers 2 - sum of 2 numbers

Export both the functions.

Import and call these functions from another module.

Calculator.js

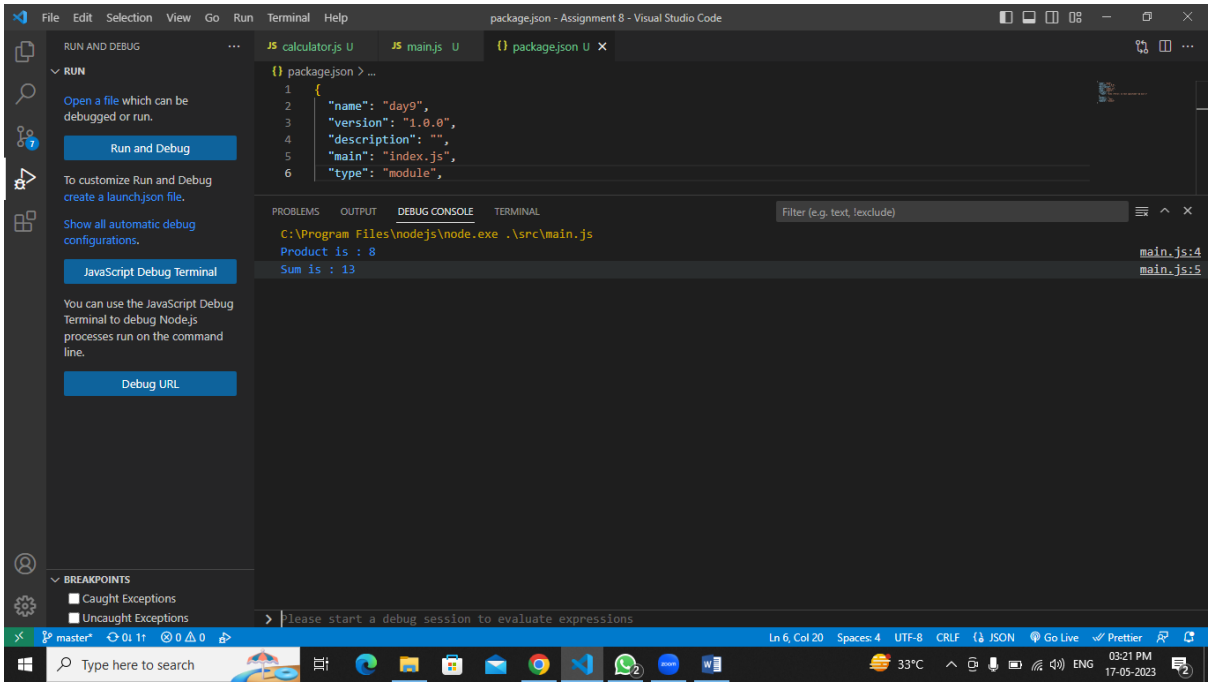
```
function product(n1, n2) {
  return n1 * n2;
}

function sum(n1, n2) {
  return n1 + n2;
}

export { product, sum };
```

main.js

```
import { product, sum } from "./calculator.js";

function main() {
  console.log("Product is : " + product(2, 4));
  console.log("Sum is : " + sum(7, 6));
}

main();
```

package.json

```
{
  "name": "day9",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "module": "esnext"
}
```



2) Create a js function to write a string to a file Create another js function to read and return the contents of a file. Call these 2 functions from another module.

Write.js

```
import { writeFileSync } from "node:fs";
import { readFileSync } from "node:fs";
function write() {
  let data = "hello World ";
  let filePath = "C:/Users/Paresh/Desktop/Web Programing/Assignment
8/abc.txt";
  writeFileSync(filePath, data);
  console.log("File written successfully");
  console.log("File has following content");
  let fileData = readFileSync(filePath, { encoding: "utf-8" });
  console.log(fileData);
}

export { write };
```

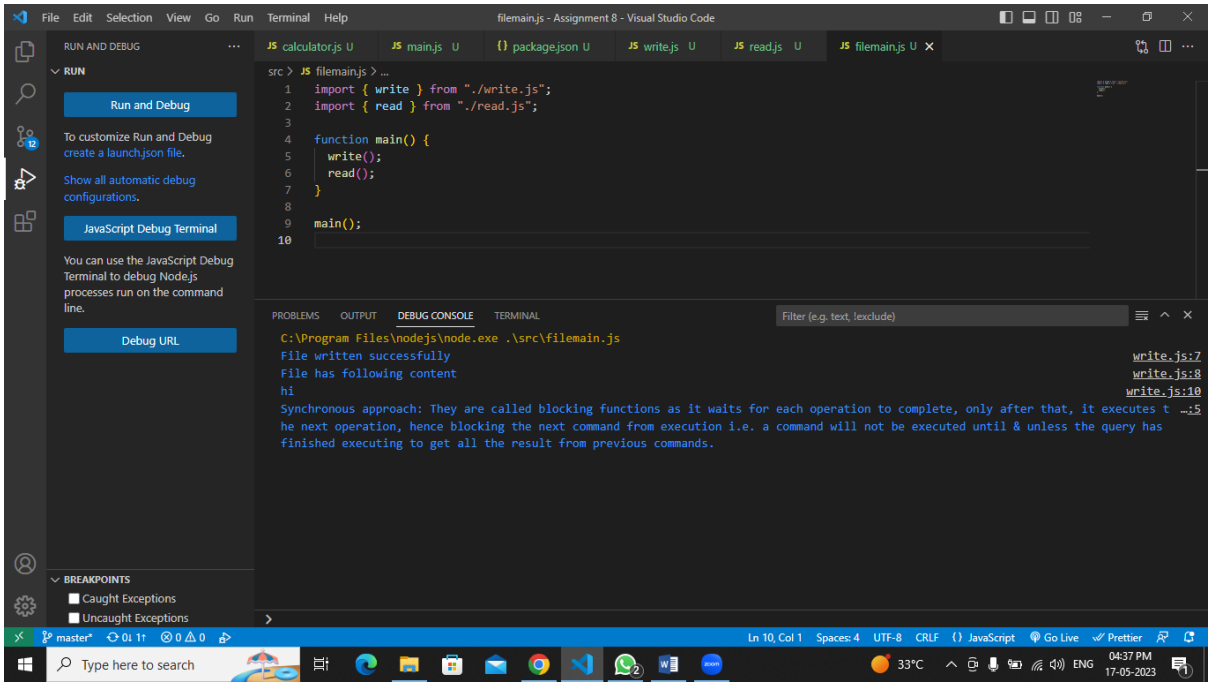
read.js

```
import { readFileSync } from "node:fs";
function read() {
  let filePath = "C:/Users/Paresh/Desktop/Web Programing/Assignment
8/demo.txt";
  let fileData = readFileSync(filePath, { encoding: "utf-8" });
  console.log(fileData);
```

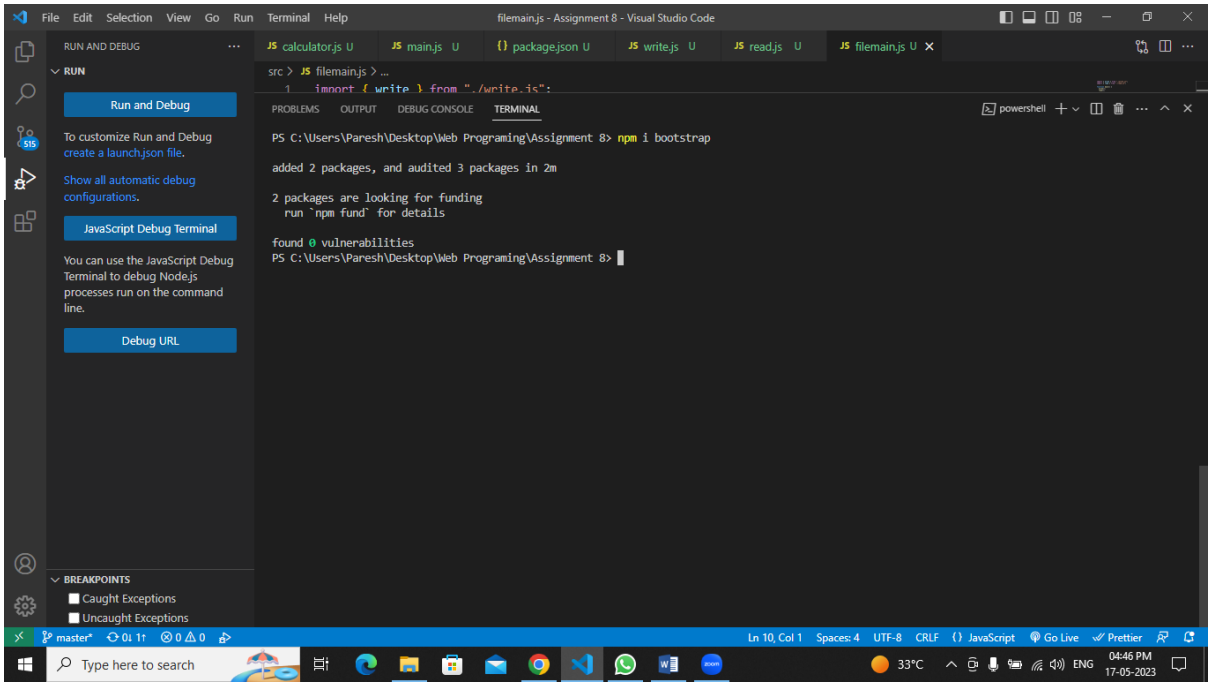
```
}

export { read };
```

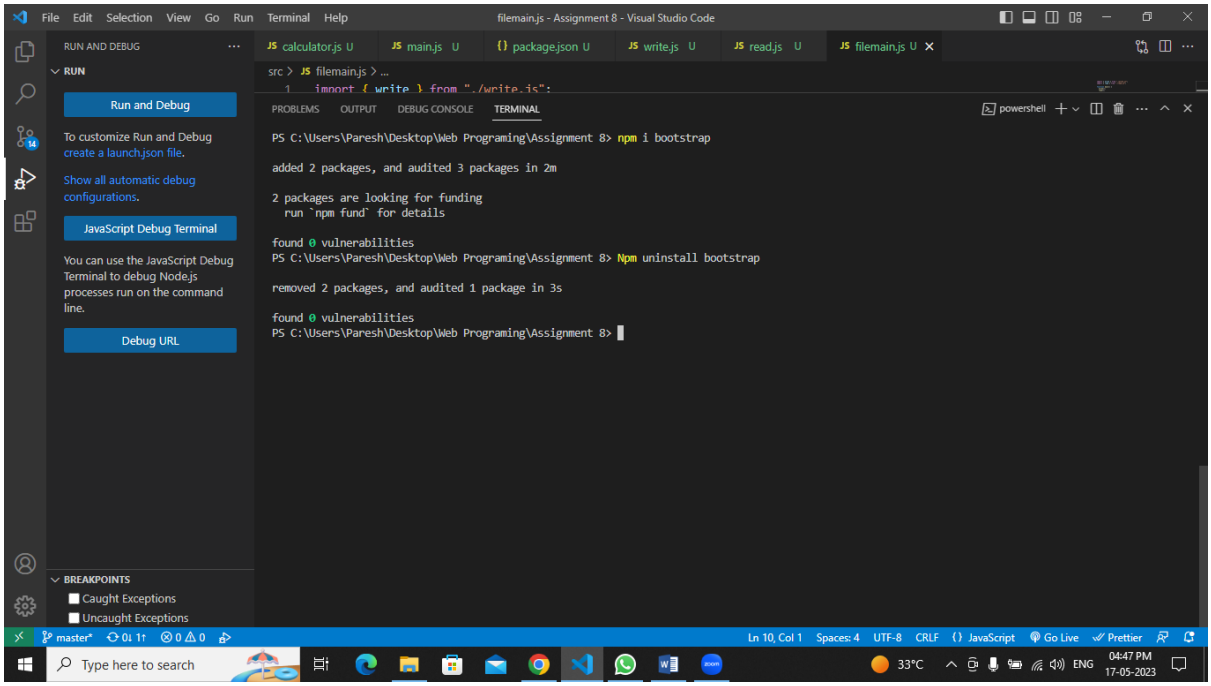
FileMain.js

```
import { write } from "./write.js";
import { read } from "./read.js";

function main() {
  write();
  read();
}

main();
```



3) practice installing & un installing an external module eg:- bootstrap to your Node.js project.





4) using AJAX make a call to the url

https://jsonplaceholder.typicode.com/todos/

&

https://jsonplaceholder.typicode.com/todos/<id>

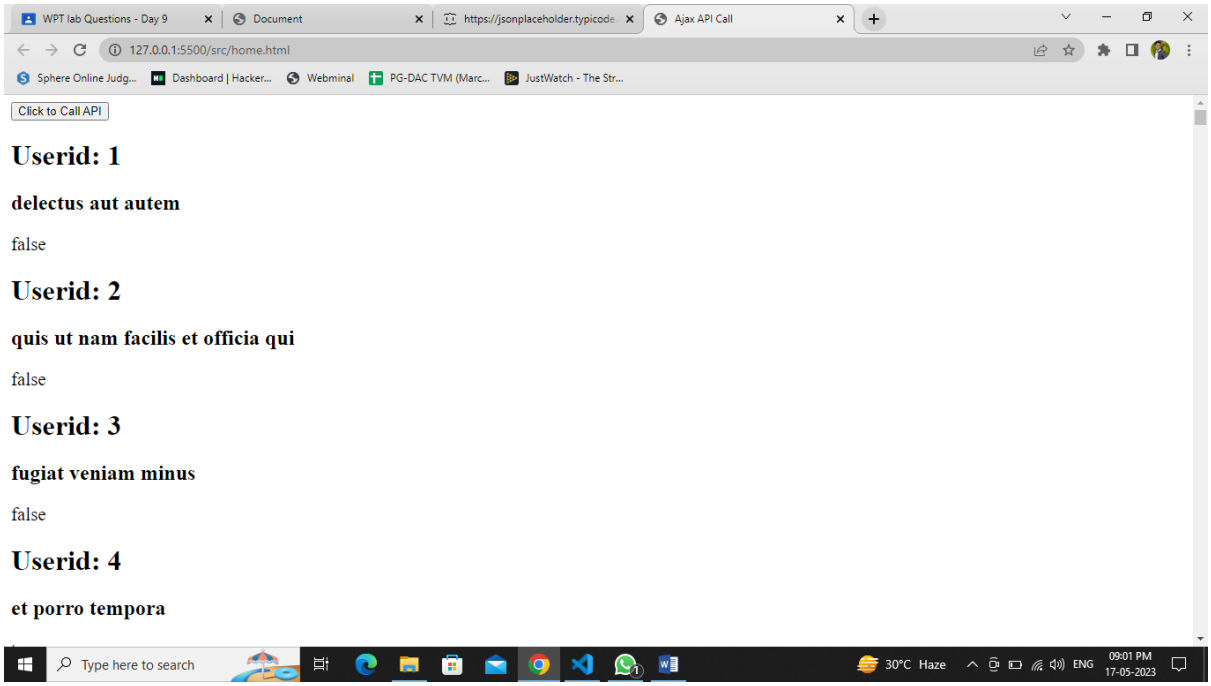
and display the results.

AjaxCall.js

```
async function apicall() {
  let path = "https://jsonplaceholder.typicode.com/todos";

  let result = await fetch(path);
  let data = await result.json();

  console.log(data);
  for (let item of data) {
    let displaytag = document.querySelector("#display");
    let existingvalue = displaytag.innerHTML;
    let newdata = `<h1>Userid: ${item.id}</h1> <h2>${item.title}</h2><p
style="font-size:21px">${item.completed}</p>`;
    displaytag.innerHTML = existingvalue + newdata;
  }
}
```

Home.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Ajax API Call</title>
    <script src="./ajaxCall.js"></script>
  </head>
  <body>
    <input type="button" value="Click to Call API" onclick="apicall()" />
    <div id="display"></div>
  </body>
</html>
```



AjaxCall using id.js

```
async function apicallUsingID() {
  let idtag = document.querySelector("#id");
  let loc = idtag.value;
  let path = `https://jsonplaceholder.typicode.com/todos/${loc}`;

  let result = await fetch(path);
  let data = await result.json();

  console.log(data);

  let displaytag = document.querySelector("#display");
  let newdata = `<h1>Userid: ${data.id}</h1> <h2>${data.title}</h2><p
style="font-size:21px">${data.completed}</p>`;
  displaytag.innerHTML = newdata;
}
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <script src="./apicallUsingID.js"></script>
    <title>Document</title>
  </head>
```

```
  <body>
    <input type="text" placeholder="Enter ID" id="id" />
    <input type="button" value="Press to call API" onclick="apicallUsingID()"
/>
    <div id="display"></div>
  </body>
</html>
```