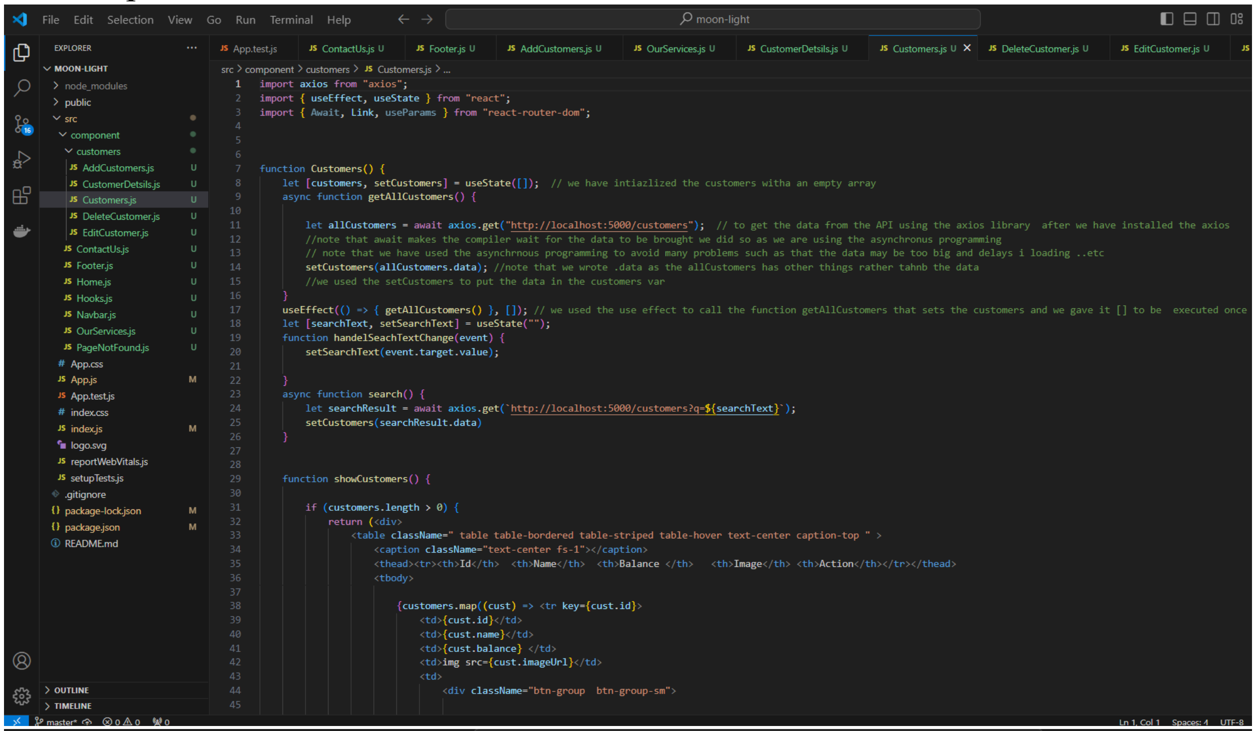
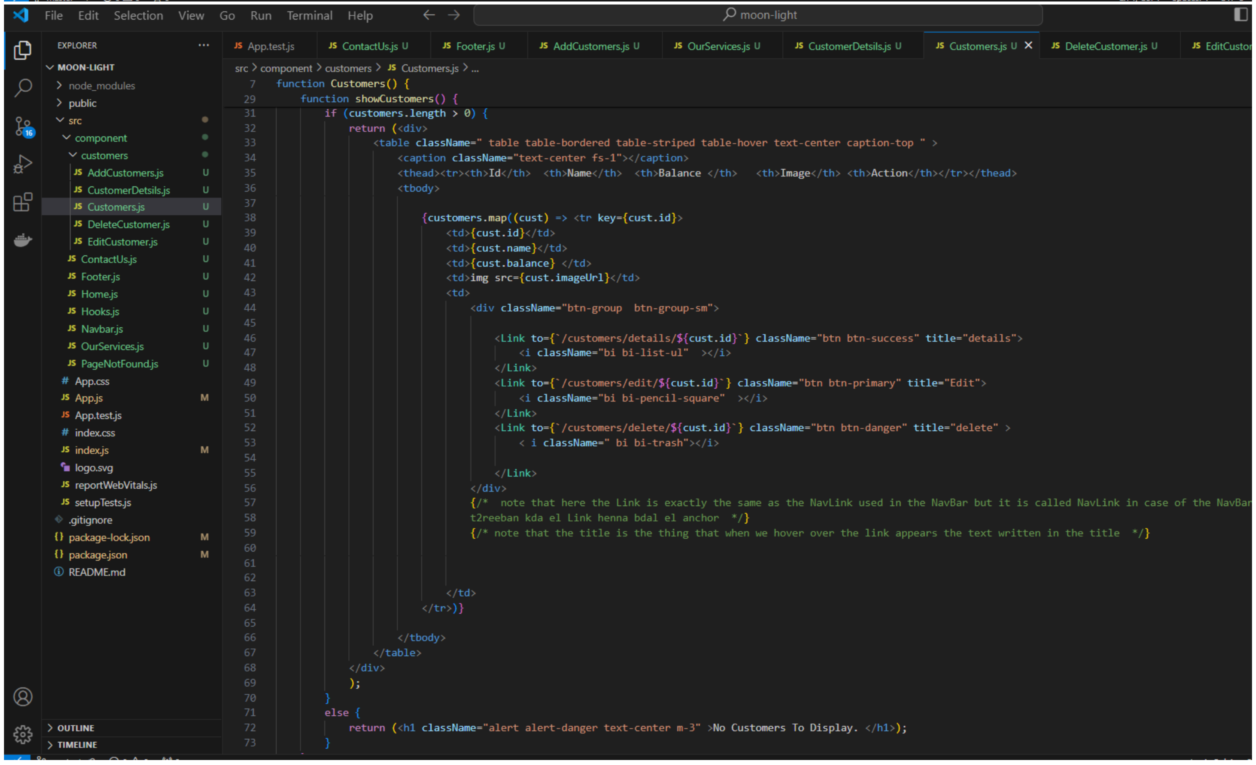
Java Script :

over all picture of the code :

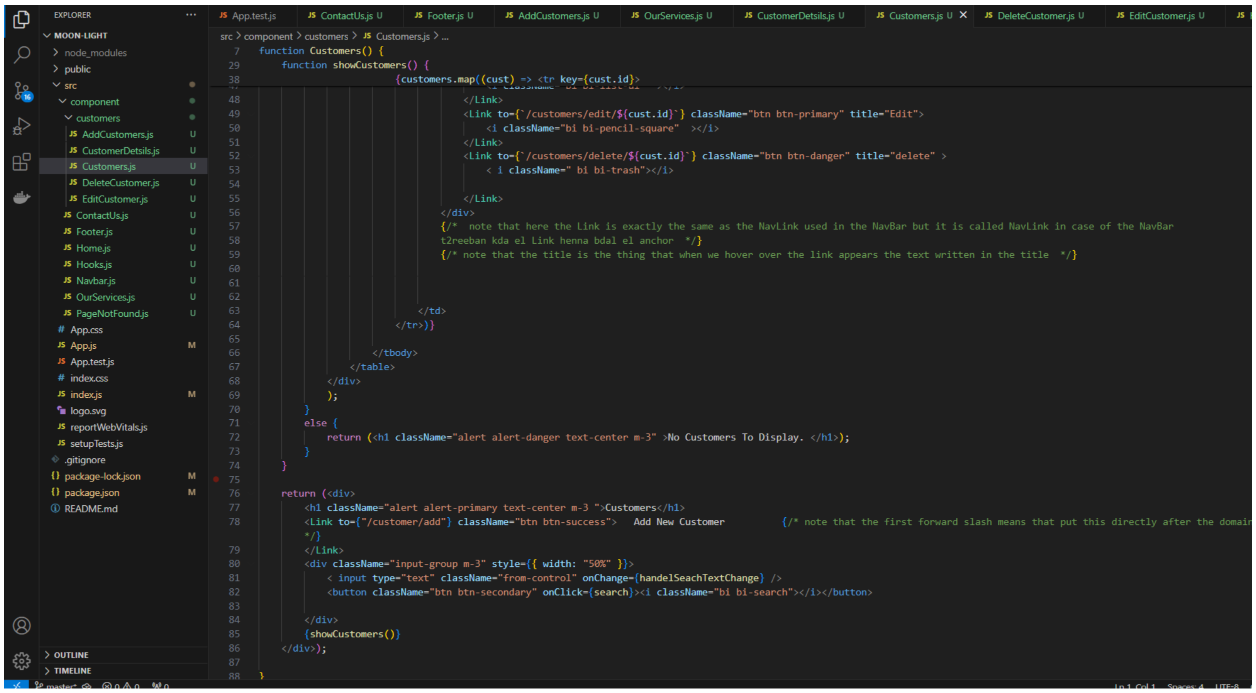


```javascript
import axios from "axios";
import { useEffect, useState } from "react";
import { Await, Link, useParams } from "react-router-dom";



function Customers() {
    let [customers, setCustomers] = useState([]);  // we have intiazlized the customers witha an empty array
    async function getAllCustomers() {

        let allCustomers = await axios.get("http://localhost:5000/customers");  // to get the data from the API using the axios library  after we have installed the axios
        //note that await makes the compiler wait for the data to be brought we did so as we are using the asynchronus programming
        // note that we have used the asynchrnous programming to avoid many problems such as that the data may be too big and delays i loading ..etc
        setCustomers(allCustomers.data); //note that we wrote .data as the allCustomers has other things rather tahnb the data
        //we used the setCustomers to put the data in the customers var
    }
    useEffect(() => { getAllCustomers() }, []); // we used the use effect to call the function getAllCustomers that sets the customers and we gave it [] to be  executed once
    let [searchText, setSearchText] = useState("");
    function handelSeachTextChange(event) {
        setSearchText(event.target.value);

    }
    async function search() {
        let searchResult = await axios.get(`http://localhost:5000/customers?q=${searchText}`);
        setCustomers(searchResult.data)
    }


    function showCustomers() {

        if (customers.length > 0) {
            return (<div>
                <table className=" table table-bordered table-striped table-hover text-center caption-top " >
                    <caption className="text-center fs-1"></caption>
                    <thead><tr><th>Id</th>  <th>Name</th>  <th>Balance </th>   <th>Image</th> <th>Action</th></tr></thead>
                    <tbody>

                        {customers.map((cust) => <tr key={cust.id}>
                            <td>{cust.id}</td>
                            <td>{cust.name}</td>
                            <td>{cust.balance} </td>
                            <td>img src={cust.imageUrl}</td>
                            <td>
                                <div className="btn-group  btn-group-sm">

```



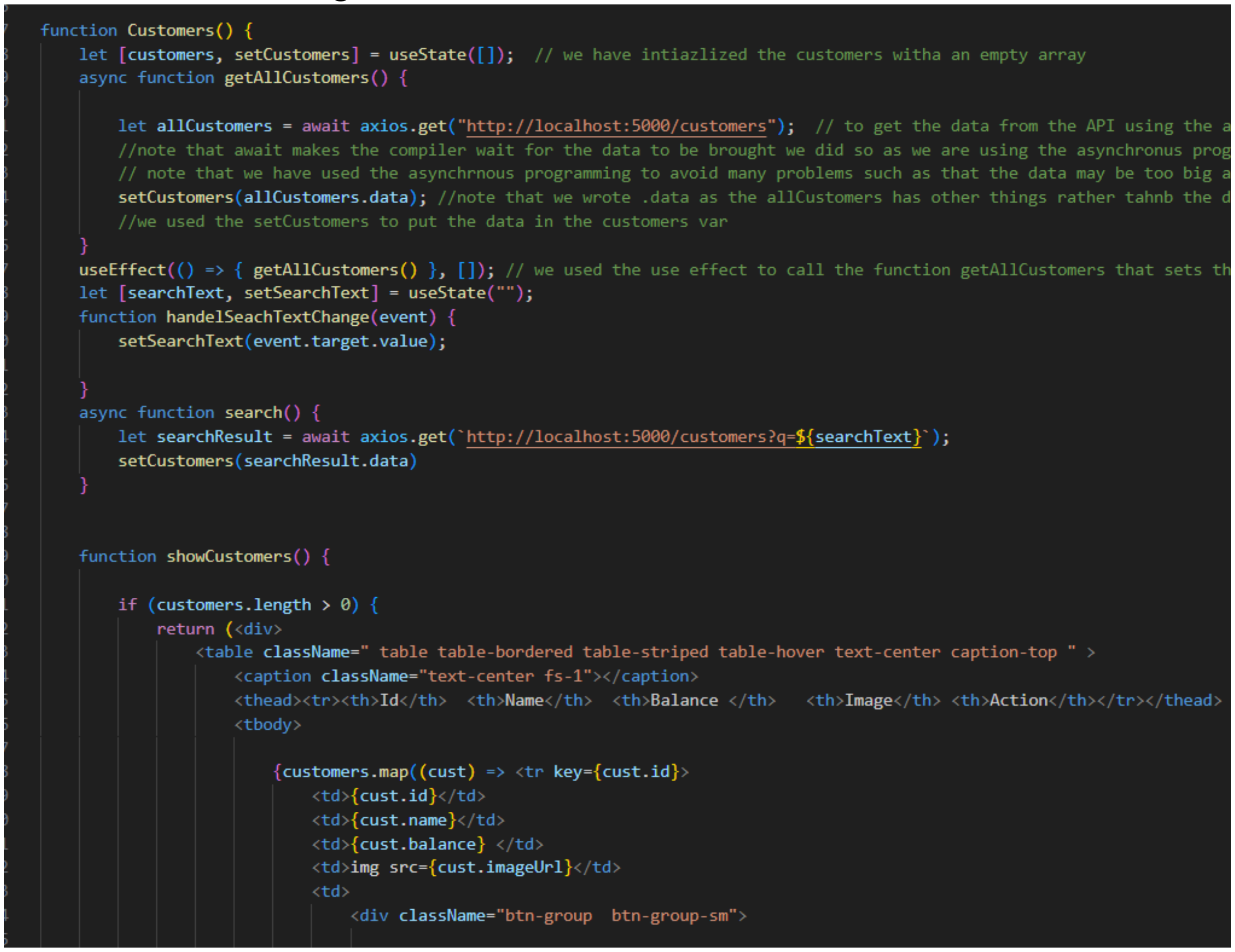
```javascript
function Customers() {
    function showCustomers() {
        if (customers.length > 0) {
            return (<div>
                <table className=" table table-bordered table-striped table-hover text-center caption-top " >
                    <caption className="text-center fs-1"></caption>
                    <thead><tr><th>Id</th>  <th>Name</th>  <th>Balance </th>   <th>Image</th> <th>Action</th></tr></thead>
                    <tbody>

                        {customers.map((cust) => <tr key={cust.id}>
                            <td>{cust.id}</td>
                            <td>{cust.name}</td>
                            <td>{cust.balance} </td>
                            <td>img src={cust.imageUrl}</td>
                            <td>
                                <div className="btn-group  btn-group-sm">

```

```javascript
                                    <Link to={`/customers/details/${cust.id}`} className="btn btn-success" title="details">
                                        <i className="bi bi-list-ul"  ></i>
                                    </Link>
                                    <Link to={`/customers/edit/${cust.id}`} className="btn btn-primary" title="Edit">
                                        <i className="bi bi-pencil-square"  ></i>
                                    </Link>
                                    <Link to={`/customers/delete/${cust.id}`} className="btn btn-danger" title="delete" >
                                        < i className=" bi bi-trash"></i>

                                    </Link>
                                </div>
                                {/*  note that here the Link is exactly the same as the NavLink used in the NavBar but it is called NavLink in case of the NavBar
                                t2reeban kda el Link henna bdal el anchor  */}
                                {/* note that the title is the thing that when we hover over the link appears the text written in the title  */}



                            </td>
                        </tr>)}

                    </tbody>
                </table>
            </div>
            );
        }
        else {
            return (<h1 className="alert alert-danger text-center m-3" >No Customers To Display. </h1>);
        }
```

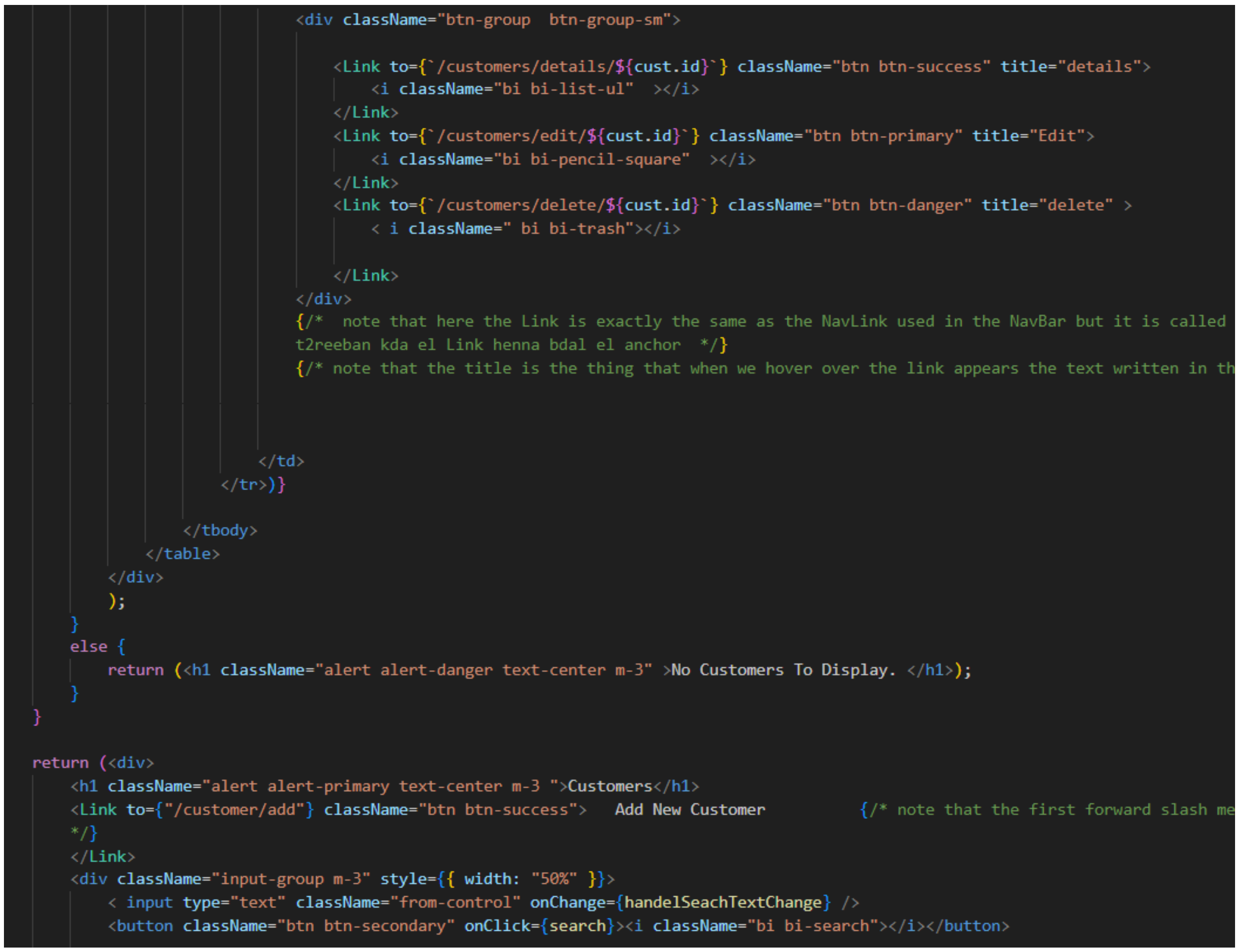


```
function Customers() {
    function showCustomers() {
                        {customers.map((cust) => <tr key={cust.id}>
                                </Link>
                                <Link to={`/customers/edit/${cust.id}`} className="btn btn-primary" title="Edit">
                                    <i className="bi bi-pencil-square"  ></i>
                                </Link>
                                <Link to={`/customers/delete/${cust.id}`} className="btn btn-danger" title="delete" >
                                    < i className=" bi bi-trash"></i>

                                </Link>
                            </div>
                            {/*  note that here the Link is exactly the same as the NavLink used in the NavBar but it is called NavLink in case of the NavBar
                            t2reeban kda el Link henna bdal el anchor  */}
                            {/* note that the title is the thing that when we hover over the link appears the text written in the title  */}



                        </td>
                    </tr>)}

                </tbody>
            </table>
        </div>
        );
    }
    else {
        return (<h1 className="alert alert-danger text-center m-3" >No Customers To Display. </h1>);
    }
  }

  return (<div>
      <h1 className="alert alert-primary text-center m-3 ">Customers</h1>
      <Link to={"/customer/add"} className="btn btn-success">   Add New Customer          {/* note that the first forward slash means that put this directly after the domain
      */}
      </Link>
      <div className="input-group m-3" style={{ width: "50%" }}>
          < input type="text" className="from-control" onChange={handelSeachTextChange} />
          <button className="btn btn-secondary" onClick={search}><i className="bi bi-search"></i></button>

      </div>
      {showCustomers()}
  </div>);

}
```

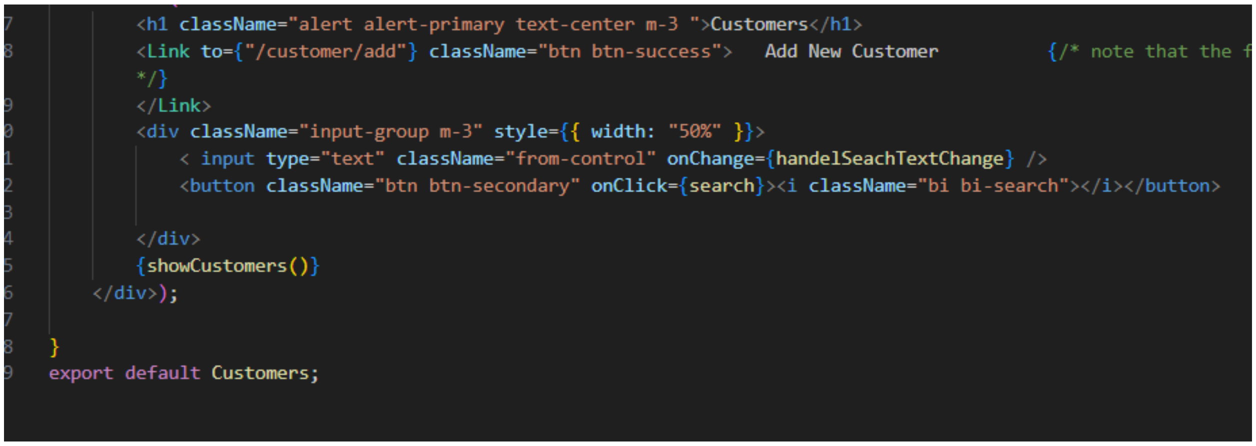
detailed view of the logic :



```
function Customers() {
    let [customers, setCustomers] = useState([]);  // we have intiazlized the customers witha an empty array
    async function getAllCustomers() {

        let allCustomers = await axios.get("http://localhost:5000/customers");  // to get the data from the API using the a
        //note that await makes the compiler wait for the data to be brought we did so as we are using the asynchronus prog
        // note that we have used the asynchrnous programming to avoid many problems such as that the data may be too big a
        setCustomers(allCustomers.data); //note that we wrote .data as the allCustomers has other things rather tahnb the d
        //we used the setCustomers to put the data in the customers var
    }
    useEffect(() => { getAllCustomers() }, []); // we used the use effect to call the function getAllCustomers that sets th
    let [searchText, setSearchText] = useState("");
    function handelSeachTextChange(event) {
        setSearchText(event.target.value);

    }
    async function search() {
        let searchResult = await axios.get(`http://localhost:5000/customers?q=${searchText}`);
        setCustomers(searchResult.data)
    }


    function showCustomers() {

        if (customers.length > 0) {
            return (<div>
                <table className=" table table-bordered table-striped table-hover text-center caption-top " >
                    <caption className="text-center fs-1"></caption>
                    <thead><tr><th>Id</th>  <th>Name</th>  <th>Balance </th>   <th>Image</th> <th>Action</th></tr></thead>
                    <tbody>

                        {customers.map((cust) => <tr key={cust.id}>
                            <td>{cust.id}</td>
                            <td>{cust.name}</td>
                            <td>{cust.balance} </td>
                            <td>img src={cust.imageUrl}</td>
                            <td>
                                <div className="btn-group  btn-group-sm">
```



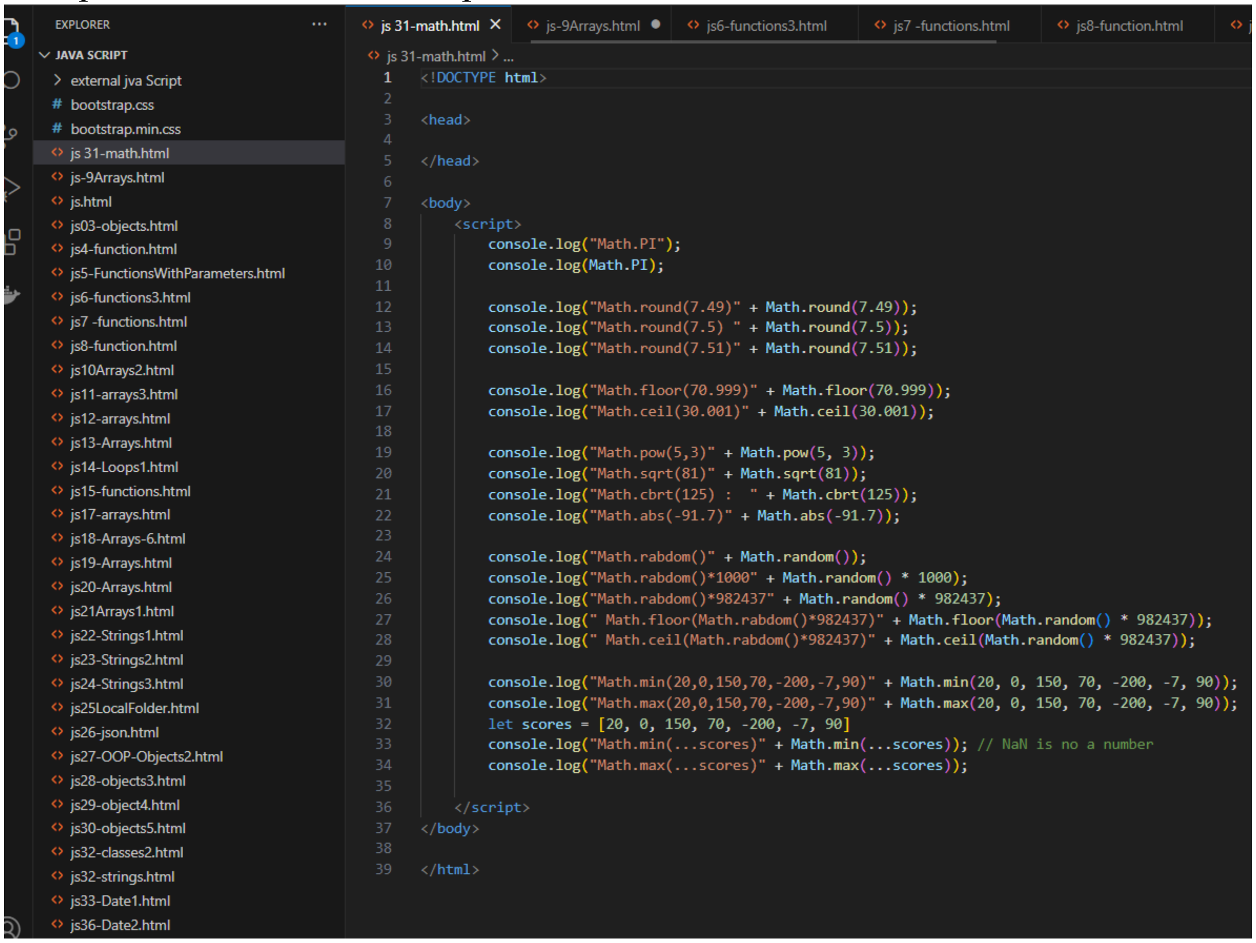
```
                        <div className="btn-group  btn-group-sm">

                            <Link to={`/customers/details/${cust.id}`} className="btn btn-success" title="details">
                                <i className="bi bi-list-ul"  ></i>
                            </Link>
                            <Link to={`/customers/edit/${cust.id}`} className="btn btn-primary" title="Edit">
                                <i className="bi bi-pencil-square"  ></i>
                            </Link>
                            <Link to={`/customers/delete/${cust.id}`} className="btn btn-danger" title="delete" >
                                < i className=" bi bi-trash"></i>

                            </Link>
                        </div>
                        {/*  note that here the Link is exactly the same as the NavLink used in the NavBar but it is called
                        t2reeban kda el Link henna bdal el anchor  */}
                        {/* note that the title is the thing that when we hover over the link appears the text written in th



                    </td>
                </tr>)}

            </tbody>
        </table>
    </div>
    );
  }
  else {
    return (<h1 className="alert alert-danger text-center m-3" >No Customers To Display. </h1>);
  }
}

return (<div>
    <h1 className="alert alert-primary text-center m-3 ">Customers</h1>
    <Link to={"/customer/add"} className="btn btn-success">   Add New Customer          {/* note that the first forward slash me
    */}
    </Link>
    <div className="input-group m-3" style={{ width: "50%" }}>
        < input type="text" className="from-control" onChange={handelSeachTextChange} />
        <button className="btn btn-secondary" onClick={search}><i className="bi bi-search"></i></button>
```



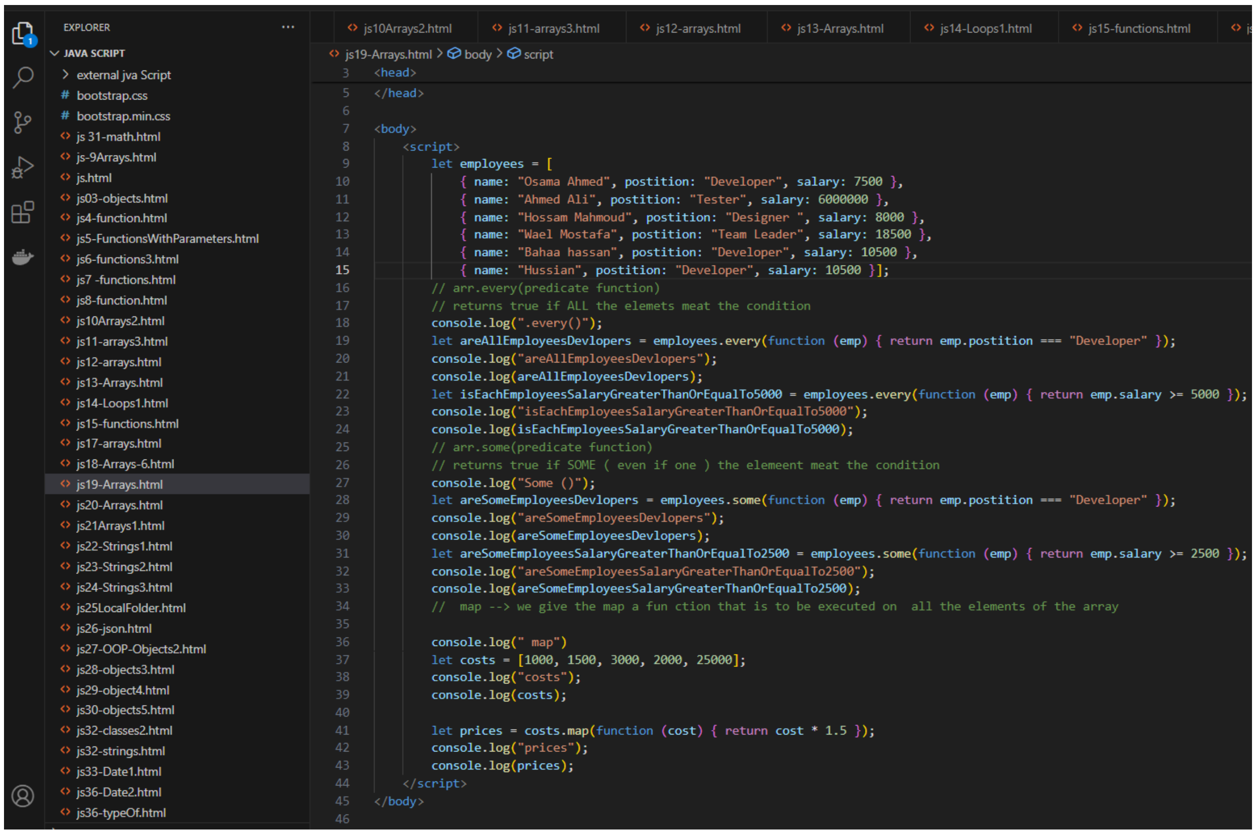
```
        <h1 className="alert alert-primary text-center m-3 ">Customers</h1>
        <Link to={"/customer/add"} className="btn btn-success">   Add New Customer          {/* note that the f
        */}
        </Link>
        <div className="input-group m-3" style={{ width: "50%" }}>
            < input type="text" className="from-control" onChange={handelSeachTextChange} />
            <button className="btn btn-secondary" onClick={search}><i className="bi bi-search"></i></button>
```

```

        </div>
        {showCustomers()}
    </div>);

}
export default Customers;
```

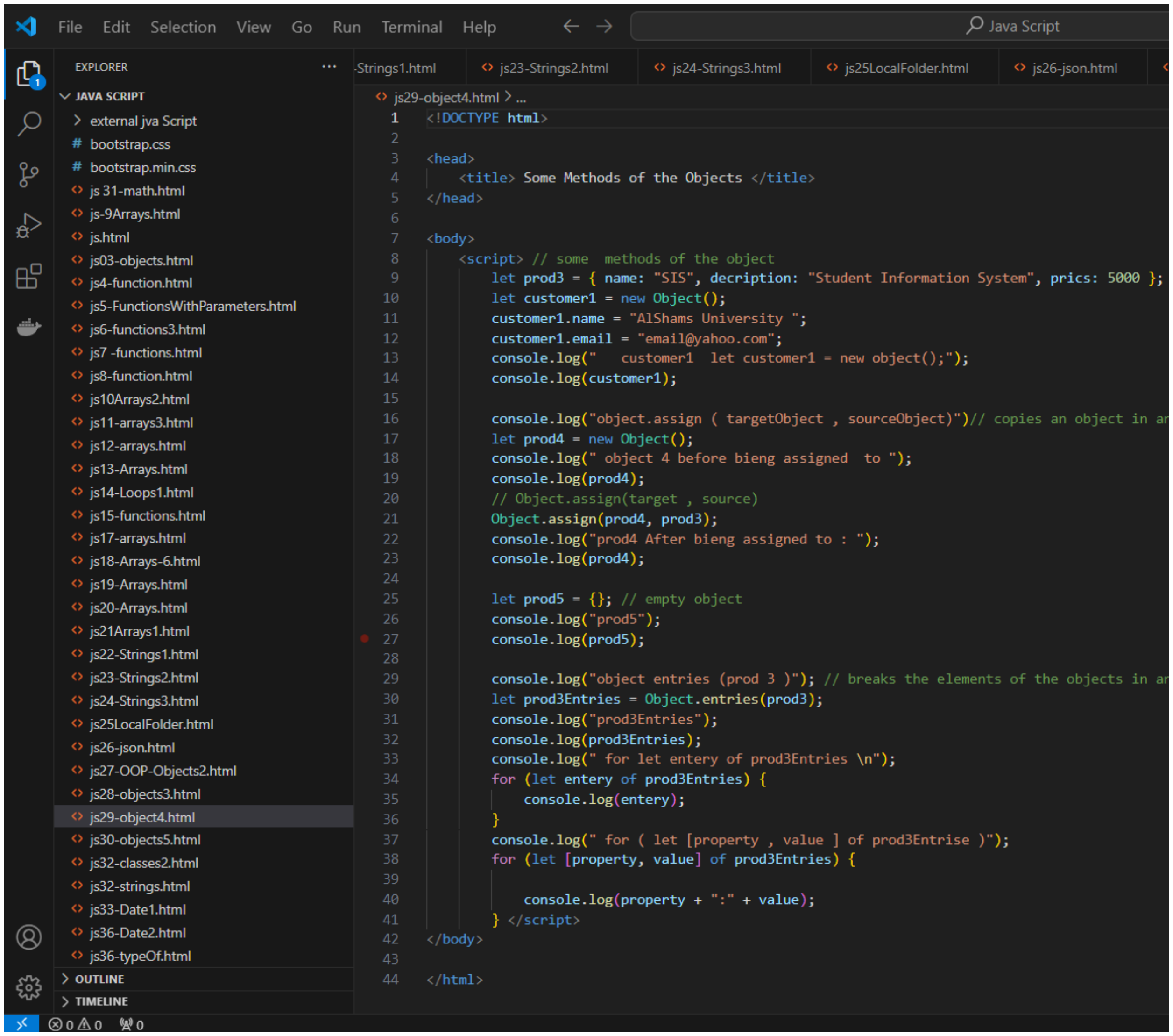
note that the Java Script code is written between the script tags
<script > // JS code ….. </ script>



```html
<!DOCTYPE html>

<head>

</head>

<body>
    <script>
        console.log("Math.PI");
        console.log(Math.PI);

        console.log("Math.round(7.49)" + Math.round(7.49));
        console.log("Math.round(7.5) " + Math.round(7.5));
        console.log("Math.round(7.51)" + Math.round(7.51));

        console.log("Math.floor(70.999)" + Math.floor(70.999));
        console.log("Math.ceil(30.001)" + Math.ceil(30.001));

        console.log("Math.pow(5,3)" + Math.pow(5, 3));
        console.log("Math.sqrt(81)" + Math.sqrt(81));
        console.log("Math.cbrt(125) :  " + Math.cbrt(125));
        console.log("Math.abs(-91.7)" + Math.abs(-91.7));

        console.log("Math.rabdom()" + Math.random());
        console.log("Math.rabdom()*1000" + Math.random() * 1000);
        console.log("Math.rabdom()*982437" + Math.random() * 982437);
        console.log(" Math.floor(Math.rabdom()*982437)" + Math.floor(Math.random() * 982437));
        console.log(" Math.ceil(Math.rabdom()*982437)" + Math.ceil(Math.random() * 982437));

        console.log("Math.min(20,0,150,70,-200,-7,90)" + Math.min(20, 0, 150, 70, -200, -7, 90));
        console.log("Math.max(20,0,150,70,-200,-7,90)" + Math.max(20, 0, 150, 70, -200, -7, 90));
        let scores = [20, 0, 150, 70, -200, -7, 90]
        console.log("Math.min(...scores)" + Math.min(...scores)); // NaN is no a number
        console.log("Math.max(...scores)" + Math.max(...scores));

    </script>
</body>

</html>
```



```
<head>
</head>

<body>
    <script>
        let employees = [
            { name: "Osama Ahmed", postition: "Developer", salary: 7500 },
            { name: "Ahmed Ali", postition: "Tester", salary: 6000000 },
            { name: "Hossam Mahmoud", postition: "Designer ", salary: 8000 },
            { name: "Wael Mostafa", postition: "Team Leader", salary: 18500 },
            { name: "Bahaa hassan", postition: "Developer", salary: 10500 },
            { name: "Hussian", postition: "Developer", salary: 10500 }];
        // arr.every(predicate function)
        // returns true if ALL the elemets meat the condition
        console.log(".every()");
        let areAllEmployeesDevlopers = employees.every(function (emp) { return emp.postition === "Developer" });
        console.log("areAllEmployeesDevlopers");
        console.log(areAllEmployeesDevlopers);
        let isEachEmployeesSalaryGreaterThanOrEqualTo5000 = employees.every(function (emp) { return emp.salary >= 5000 });
        console.log("isEachEmployeesSalaryGreaterThanOrEqualTo5000");
        console.log(isEachEmployeesSalaryGreaterThanOrEqualTo5000);
        // arr.some(predicate function)
        // returns true if SOME ( even if one ) the elemeent meat the condition
        console.log("Some ()");
        let areSomeEmployeesDevlopers = employees.some(function (emp) { return emp.postition === "Developer" });
        console.log("areSomeEmployeesDevlopers");
        console.log(areSomeEmployeesDevlopers);
        let areSomeEmployeesSalaryGreaterThanOrEqualTo2500 = employees.some(function (emp) { return emp.salary >= 2500 });
        console.log("areSomeEmployeesSalaryGreaterThanOrEqualTo2500");
        console.log(areSomeEmployeesSalaryGreaterThanOrEqualTo2500);
        //  map --> we give the map a fun ction that is to be executed on  all the elements of the array

        console.log(" map")
        let costs = [1000, 1500, 3000, 2000, 25000];
        console.log("costs");
        console.log(costs);

        let prices = costs.map(function (cost) { return cost * 1.5 });
        console.log("prices");
        console.log(prices);
    </script>
</body>

```



```
<!DOCTYPE html>

<head>
    <title> Some Methods of the Objects </title>
</head>

<body>
    <script> // some  methods of the object
        let prod3 = { name: "SIS", decription: "Student Information System", prics: 5000 };
        let customer1 = new Object();
        customer1.name = "AlShams University ";
        customer1.email = "email@yahoo.com";
        console.log("   customer1  let customer1 = new object();");
        console.log(customer1);

        console.log("object.assign ( targetObject , sourceObject)")// copies an object in ar
        let prod4 = new Object();
        console.log(" object 4 before bieng assigned  to ");
        console.log(prod4);
        // Object.assign(target , source)
        Object.assign(prod4, prod3);
        console.log("prod4 After bieng assigned to : ");
        console.log(prod4);

        let prod5 = {}; // empty object
        console.log("prod5");
        console.log(prod5);

        console.log("object entries (prod 3 )"); // breaks the elements of the objects in ar
        let prod3Entries = Object.entries(prod3);
        console.log("prod3Entries");
        console.log(prod3Entries);
        console.log(" for let entery of prod3Entries \n");
        for (let entery of prod3Entries) {
            console.log(entery);
        }
        console.log(" for ( let [property , value ] of prod3Entrise )");
        for (let [property, value] of prod3Entries) {

            console.log(property + ":" + value);
        } </script>
</body>

</html>
```