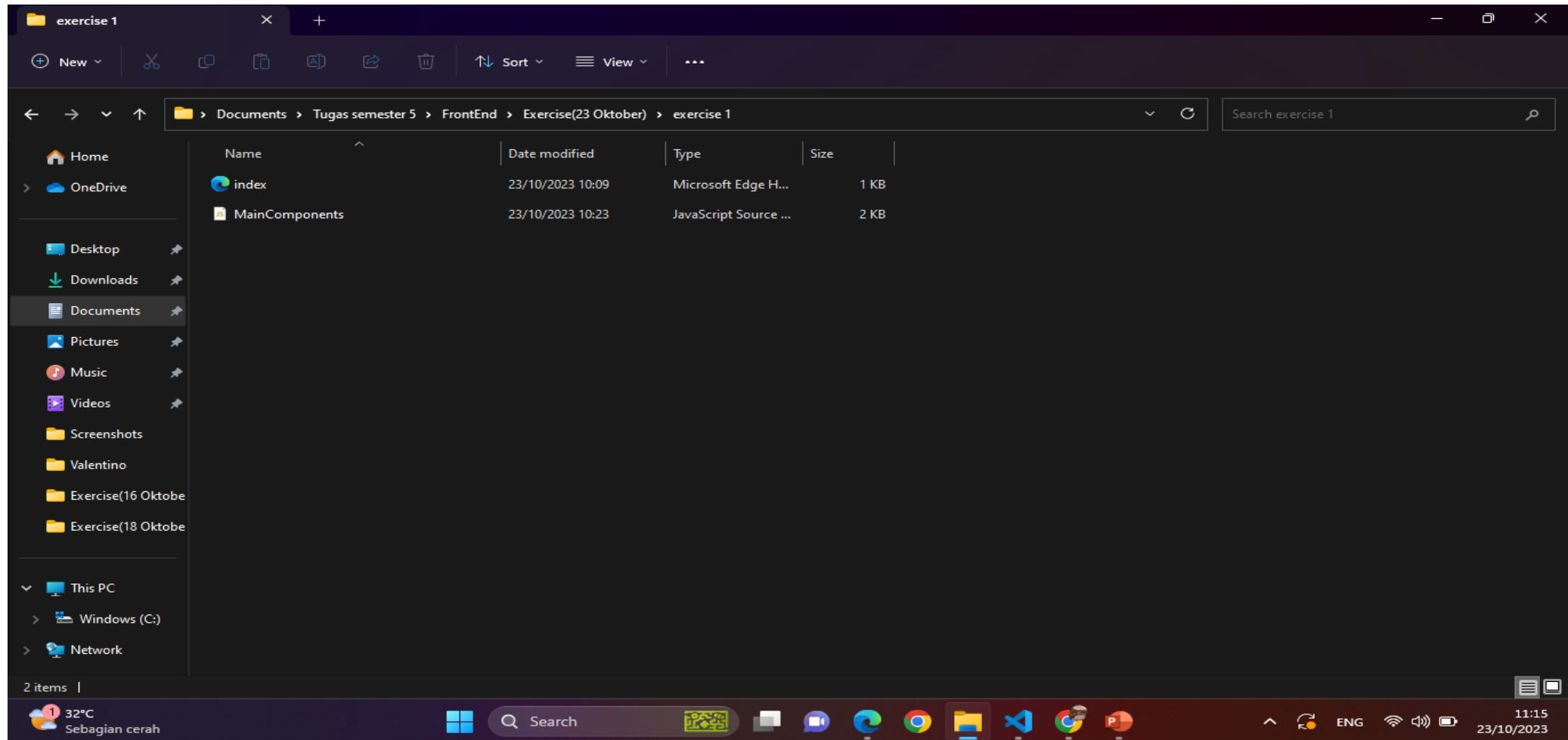


# Front-end Web Development - B

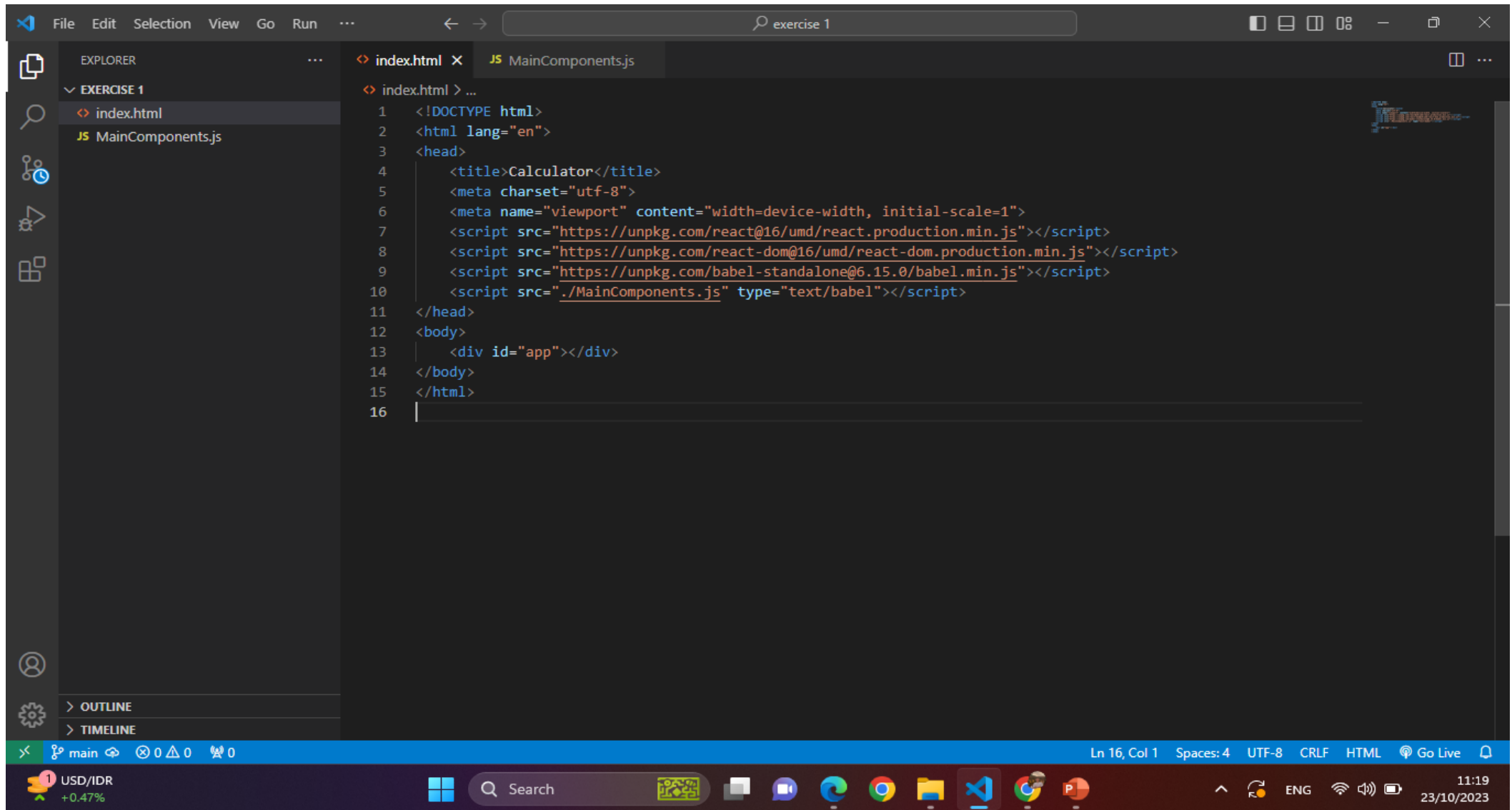
Nama : Londah, Shawn Samayim Albertkarl

## Exercise #1 [Button event handling di dalam Calculator sederhana menggunakan ReactJS

Create Files in Exercise1 Folder



# HTML Code



The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The Explorer sidebar on the left shows a project named "EXERCISE 1" with two files: "index.html" and "JS MainComponents.js". The "index.html" file is selected and its content is displayed in the main editor area. The code is an HTML document for a calculator application, including a title, meta tags for charset and viewport, and script tags for React, ReactDOM, Babel, and the local MainComponents.js file. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>Calculator</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <script src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
8   <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
9   <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
10  <script src="./MainComponents.js" type="text/babel"></script>
11 </head>
12 <body>
13   <div id="app"></div>
14 </body>
15 </html>
16
```

The status bar at the bottom indicates the current position is Line 16, Column 1, with 4 spaces. It also shows the encoding as UTF-8, line endings as CRLF, and the file type as HTML. The Windows taskbar at the very bottom shows the time as 11:19 on 23/10/2023, along with various system icons and the Start menu button.

# Javascript File

The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the file structure for 'EXERCISE 1', containing 'index.html' and 'JS MainComponents.js'. The main editor area shows the 'MainComponents.js' file with the following code:

```
1 // create class header
2 class Header extends React.Component {
3   constructor() {
4     super();
5     // membuat objek state
6     this.state = {
7       angka1 : "0",
8       angka2 : "0",
9       result : "0"
10    };
11  }
12  // set values
13  setValue1 = (val1) => { this.setState({ angka1: ""+val1 }); }
14  setValue2 = (val2) => { this.setState({ angka2: ""+val2 }); }
15
16  // Arithmetic operations such as addition, subtraction, division, multiplication
17  funcMultiply = (val1) => {
18    var hasil = parseInt(this.state.angka1) * parseInt(this.state.angka2);
19    this.setState({ result: ""+hasil });
20  }
21
22  funcDivide = (val1) => {
23    var hasil = parseInt(this.state.angka1) / parseInt(this.state.angka2);
24    this.setState({ result: ""+hasil });
25  }
26
27  funcSubtract = (val1) => {
28    var hasil = parseInt(this.state.angka1) - parseInt(this.state.angka2);
29    this.setState({ result: ""+hasil });
30  }
31
32  funcAdd = (val1) => {
```

The status bar at the bottom indicates the current file is 'main\*', the cursor is at line 35, column 6, and the file encoding is UTF-8. The system tray at the very bottom shows the temperature as 32°C and the date as 23/10/2023.

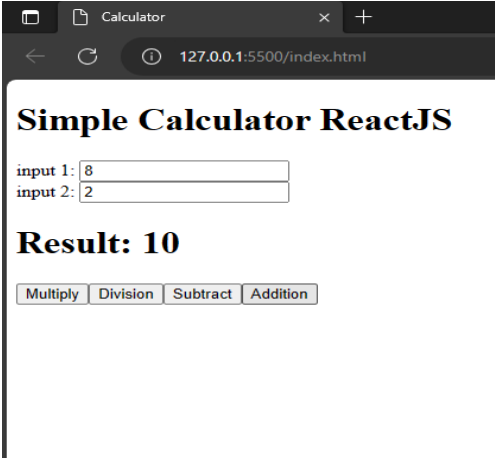
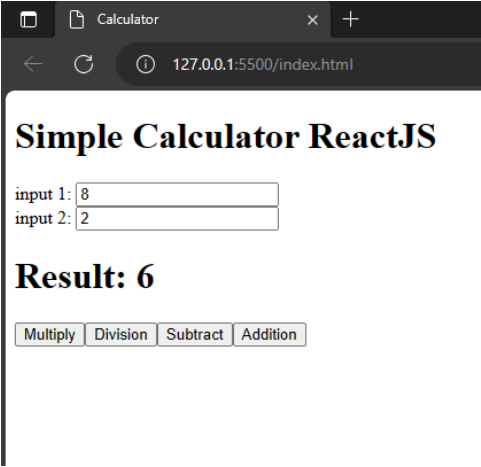
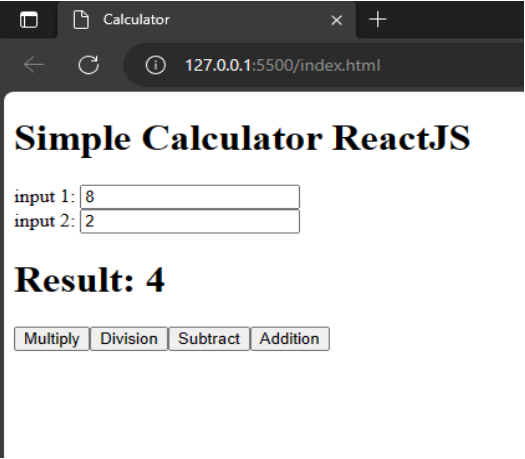
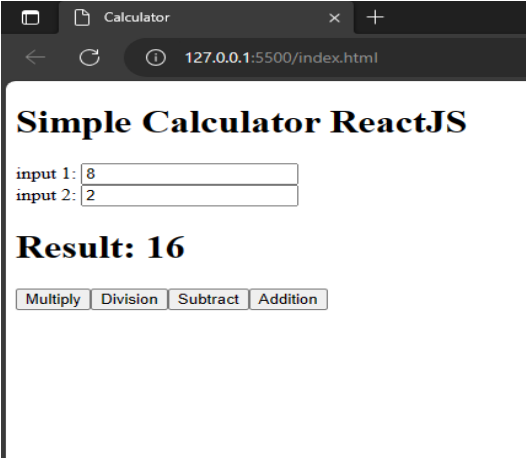
# Javascript File

The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named "EXERCISE 1" containing two files: "index.html" and "JS MainComponents.js". The main editor area displays the content of "MainComponents.js". The code defines a component with state and functions for a simple calculator.

```
JS MainComponents.js > Header > funcAdd
28     var hasil = parseInt(this.state.angka1) - parseInt(this.state.angka2);
29     this.setState({ result: ""+hasil });
30 }
31
32 funcAdd = (val1) => {
33     var hasil = parseInt(this.state.angka1) + parseInt(this.state.angka2);
34     this.setState({ result: ""+hasil });
35 }
36
37 render () {
38     return (
39         <div>
40             <h1>Simple Calculator ReactJS</h1>
41             input 1: <input value={this.state.angka1} onChange={ (e) => {this.setValue1(e.target.value)}} />
42             <br/>
43             input 2: <input value={this.state.angka2} onChange={ (e) => {this.setValue2(e.target.value)}} />
44             <h1>Result: {this.state.result}</h1>
45             <button onClick={this.funcMultiply}>Multiply</button>
46             <button onClick={this.funcDivide}>Division</button>
47             <button onClick={this.funcSubtract}>Subtract</button>
48             <button onClick={this.funcAdd}>Addition</button>
49         </div>
50     );
51 }
52
53
54 //render komponen ke RealDOM
55 ReactDOM.render(<Header />, document.getElementById("app"));
```

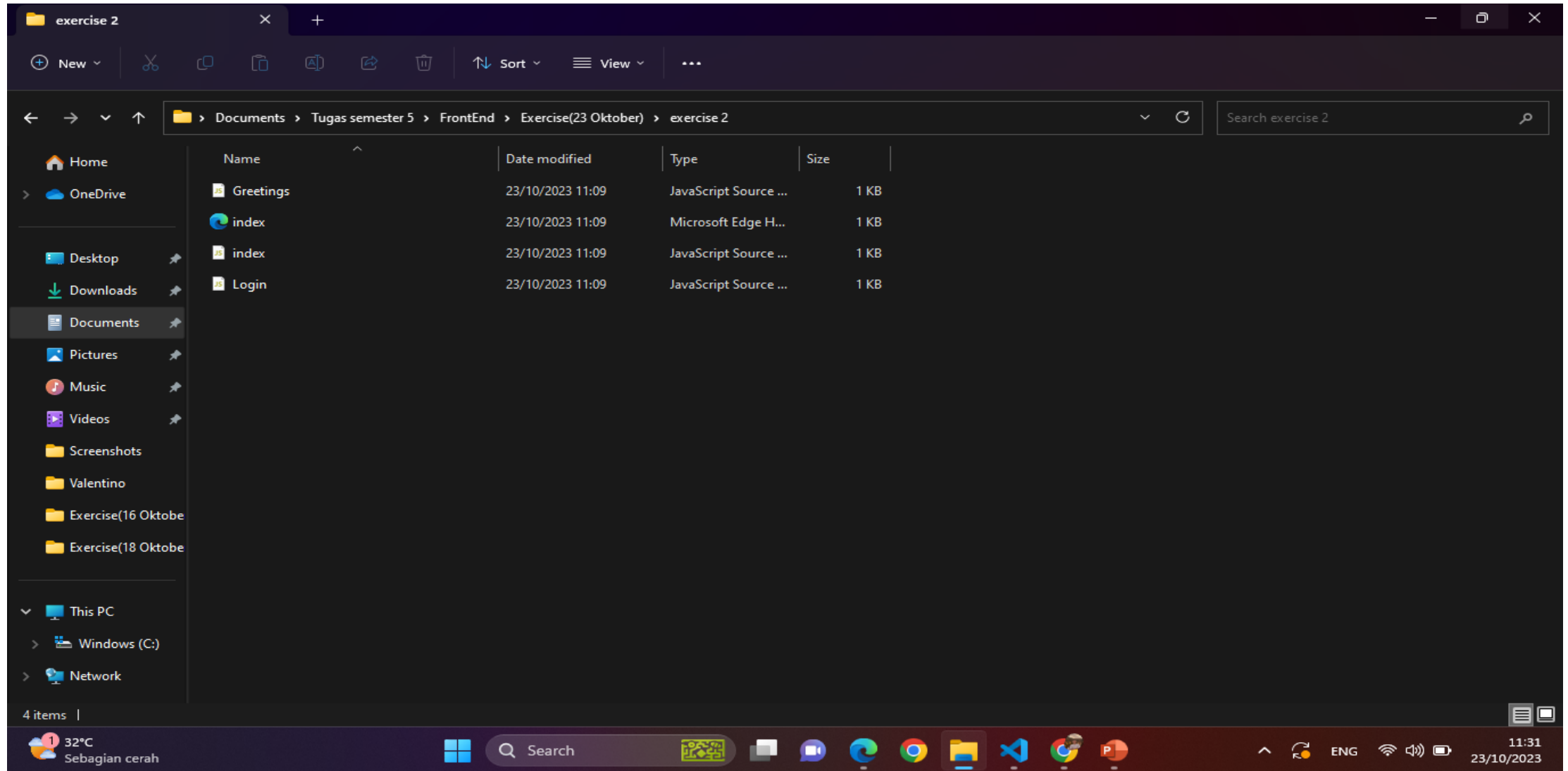
The status bar at the bottom shows the current file is "main\*", the encoding is "UTF-8", the line and column are "Ln 35, Col 6", and the language is "JavaScript". The system tray at the very bottom shows the temperature is 32°C, the time is 11:20, and the date is 23/10/2023.

Expected Output



## Exercise #2 [Login/Logout page sederhana menggunakan ReachJS Conditional Rendering

Create Files in Exercise1 Folder



# HTML Code

File Edit Selection View Go Run ... exercise 2

EXPLORER

- EXERCISE 2
  - OUTLINE
  - TIMELINE

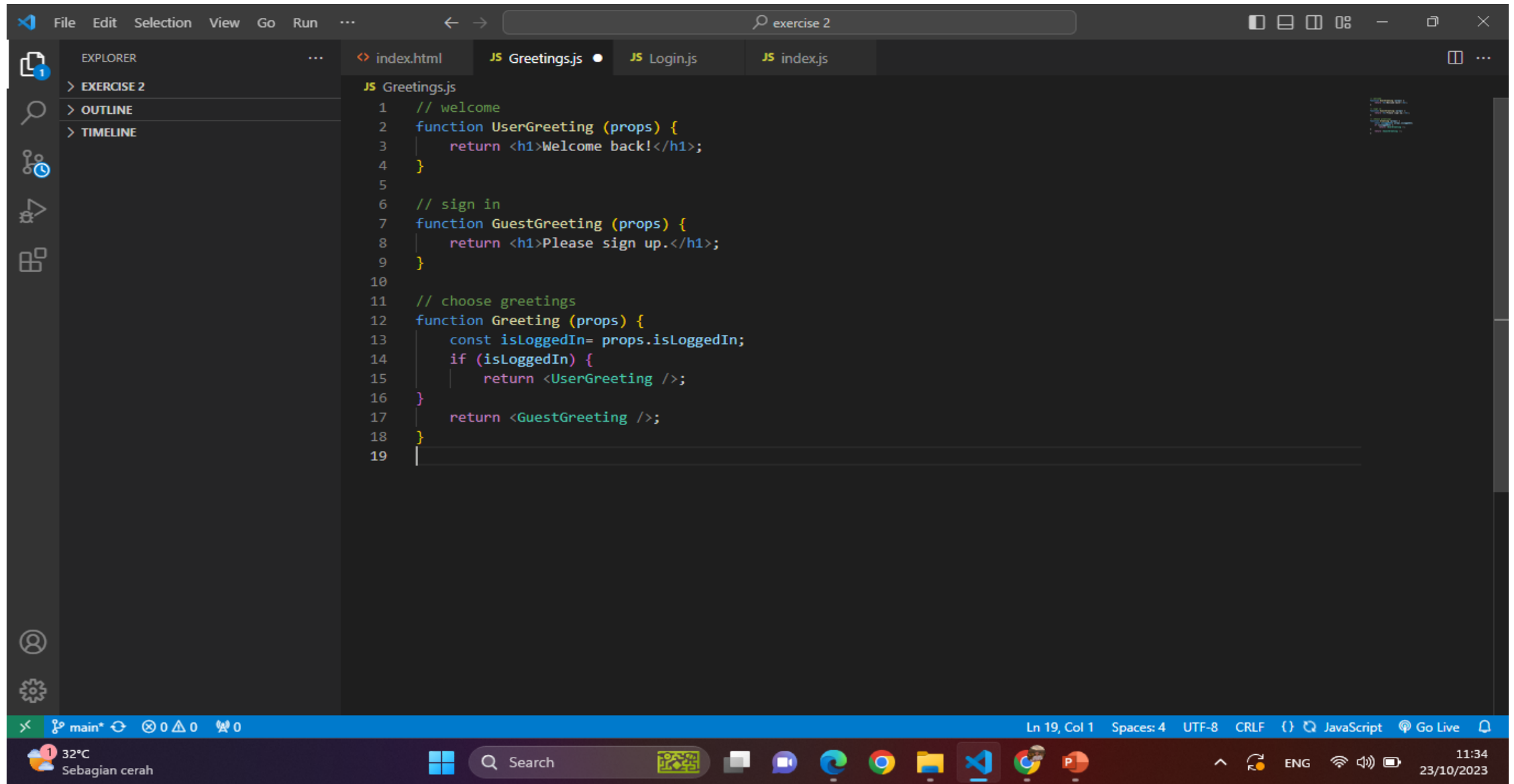
index.html > html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <title>CSS test</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <script src="https://unpkg.com/react@16/umd/react.production.min.js"></script>
8   <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>
9   <script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
10  <script src="./Greetings.js" type="text/babel"></script>
11  <script src="./Login.js" type="text/babel"></script>
12  <script src="./index.js" type="text/babel"></script>
13 </head>
14 <body>
15   <div id="app"></div>
16 </body>
17 </html>
```

Ln 17, Col 8 Spaces: 4 UTF-8 CRLF HTML Go Live

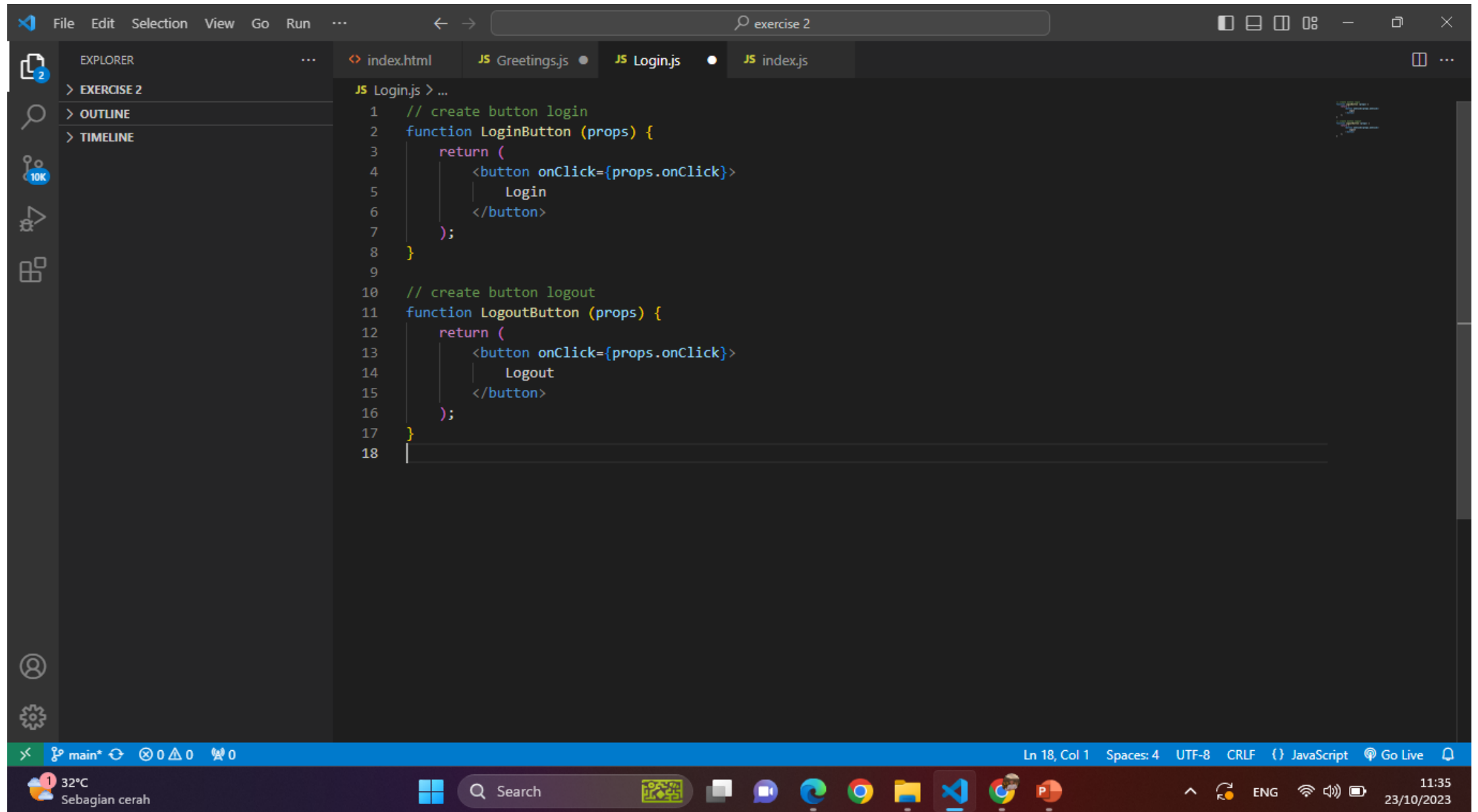
32°C Sebagian cerah 11:32 23/10/2023

# Component Greeting File





# Component Buttons File

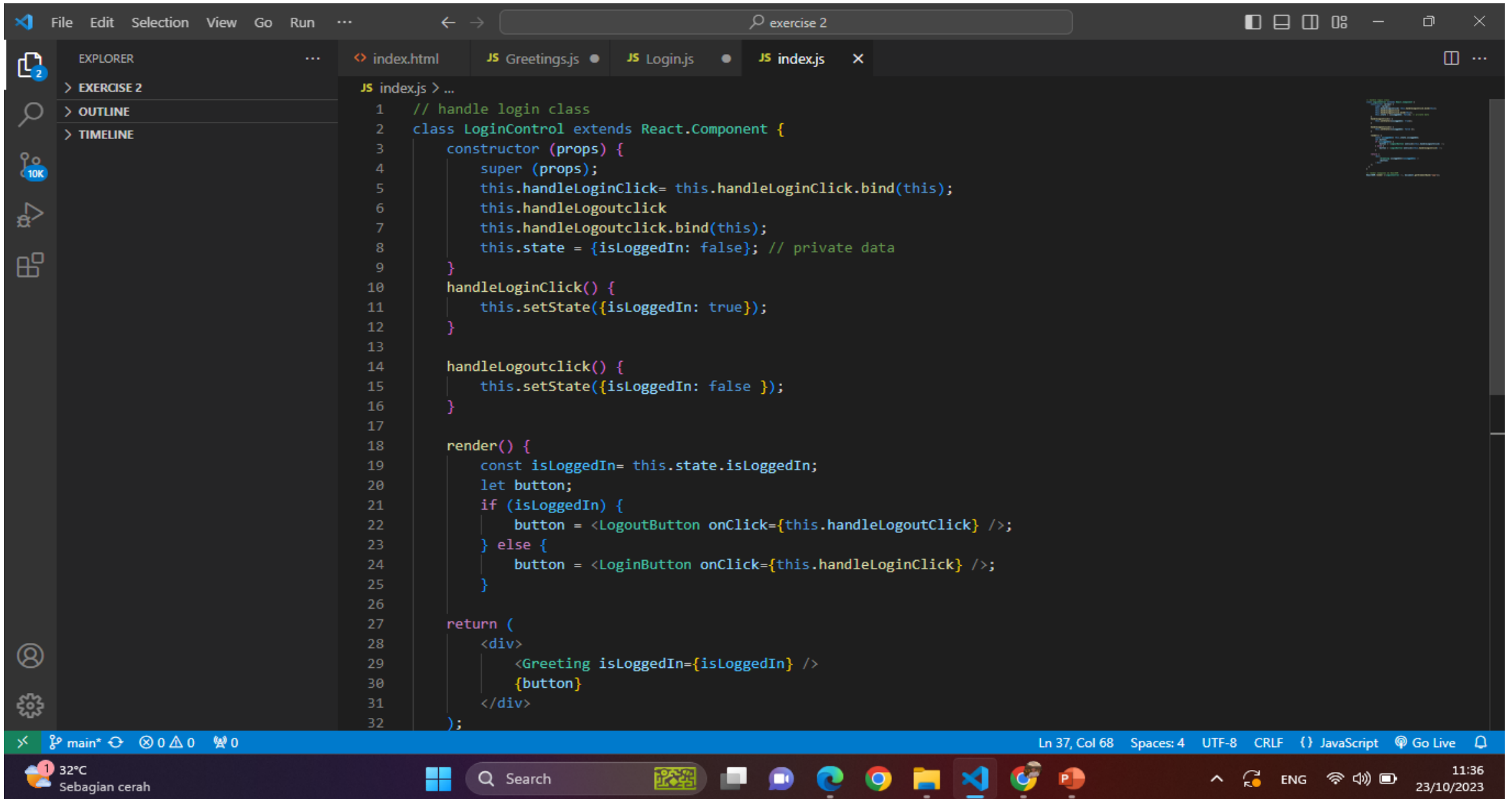


The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The title bar at the top reads "exercise 2". The Explorer sidebar on the left shows a project structure with "EXERCISE 2" expanded, containing "OUTLINE" and "TIMELINE". The file explorer shows "index.html", "JS Greetings.js", "JS Login.js", and "JS index.js". The main editor area displays the content of "JS Login.js".

```
JS Login.js > ...
1  // create button login
2  function LoginButton (props) {
3      return (
4          <button onClick={props.onClick}>
5              Login
6          </button>
7      );
8  }
9
10 // create button logout
11 function LogoutButton (props) {
12     return (
13         <button onClick={props.onClick}>
14             Logout
15         </button>
16     );
17 }
18
```

The status bar at the bottom indicates the current file is "main\*", the encoding is "UTF-8", the line and column are "Ln 18, Col 1", and the language is "JavaScript". The system tray at the very bottom shows the date and time as "23/10/2023 11:35".

# Component Handle Login

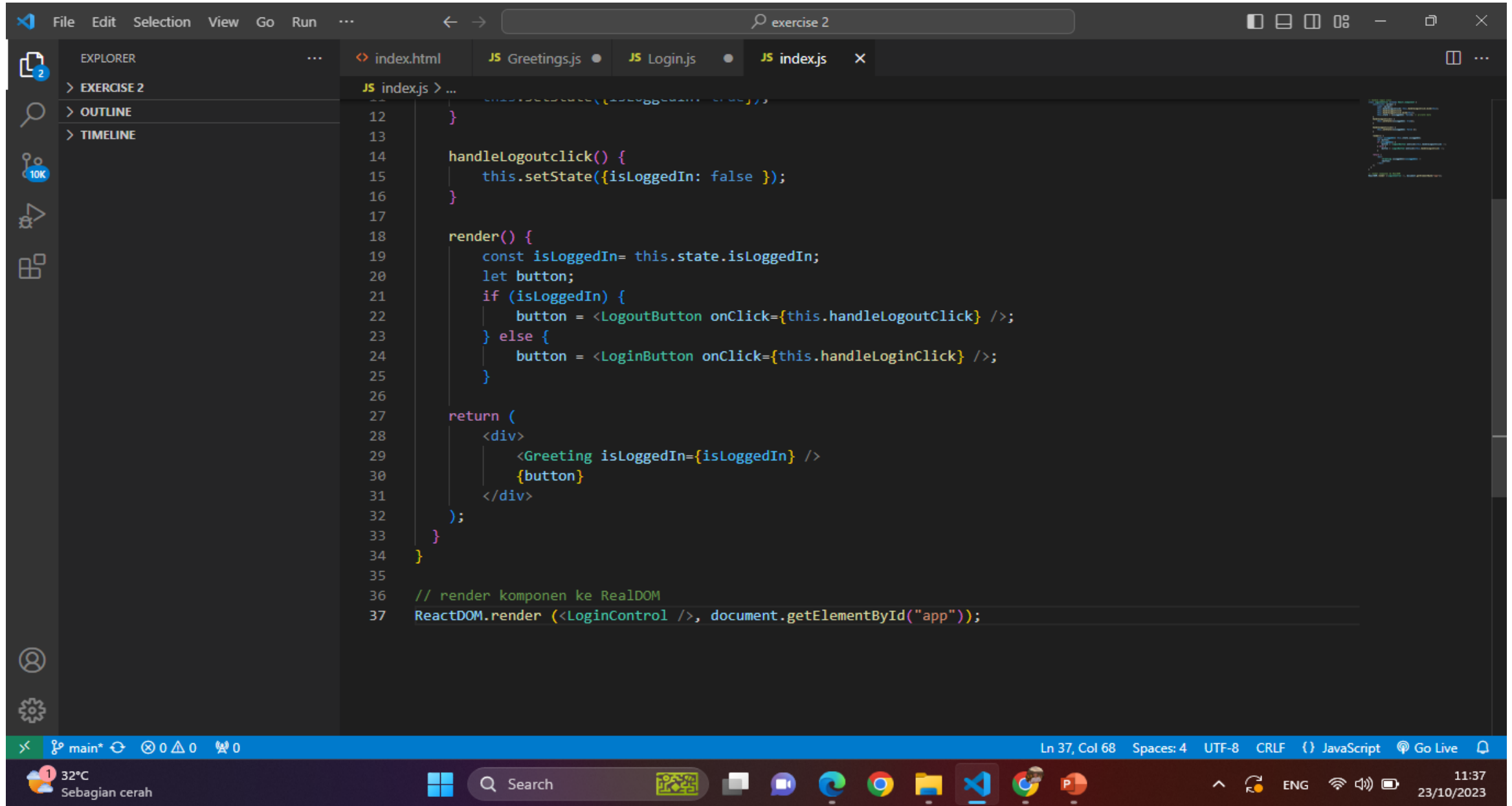


The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project structure with 'EXERCISE 2' containing 'OUTLINE' and 'TIMELINE'. The main editor area displays the file 'JS index.js' with the following code:

```
1 // handle login class
2 class LoginControl extends React.Component {
3   constructor (props) {
4     super (props);
5     this.handleClick= this.handleClick.bind(this);
6     this.handleClick
7     this.handleClick.bind(this);
8     this.state = {isLoggedIn: false}; // private data
9   }
10  handleClick() {
11    this.setState({isLoggedIn: true});
12  }
13
14  handleClick() {
15    this.setState({isLoggedIn: false });
16  }
17
18  render() {
19    const isLoggedIn= this.state.isLoggedIn;
20    let button;
21    if (isLoggedIn) {
22      button = <LogoutButton onClick={this.handleClick} />;
23    } else {
24      button = <LoginButton onClick={this.handleClick} />;
25    }
26
27    return (
28      <div>
29        <Greeting isLoggedIn={isLoggedIn} />
30        {button}
31      </div>
32    );
```

The status bar at the bottom indicates the current file is 'main\*', the cursor is at line 37, column 68, with 4 spaces. The encoding is UTF-8 and line endings are CRLF. The language is JavaScript. The system tray at the bottom shows the temperature as 32°C, the time as 11:36, and the date as 23/10/2023.

# Component Handle Login



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project structure with 'EXERCISE 2' containing 'index.html', 'Greetings.js', 'Login.js', and 'index.js'. The main editor area displays the 'index.js' file, which contains the following code:

```
12     }
13
14     handleLogoutclick() {
15         this.setState({isLoggedIn: false });
16     }
17
18     render() {
19         const isLoggedIn= this.state.isLoggedIn;
20         let button;
21         if (isLoggedIn) {
22             button = <LogoutButton onClick={this.handleLogoutClick} />;
23         } else {
24             button = <LoginButton onClick={this.handleLoginClick} />;
25         }
26
27         return (
28             <div>
29                 <Greeting isLoggedIn={isLoggedIn} />
30                 {button}
31             </div>
32         );
33     }
34 }
35
36 // render komponen ke ReactDOM
37 ReactDOM.render (<LoginControl />, document.getElementById("app"));
```

The status bar at the bottom indicates the file is 'main\*', the cursor is at line 37, column 68, and the file encoding is UTF-8. The Windows taskbar at the very bottom shows the system clock as 11:37 on 23/10/2023, along with weather information (32°C, Sebagian cerah) and various application icons.

## Expected Output

