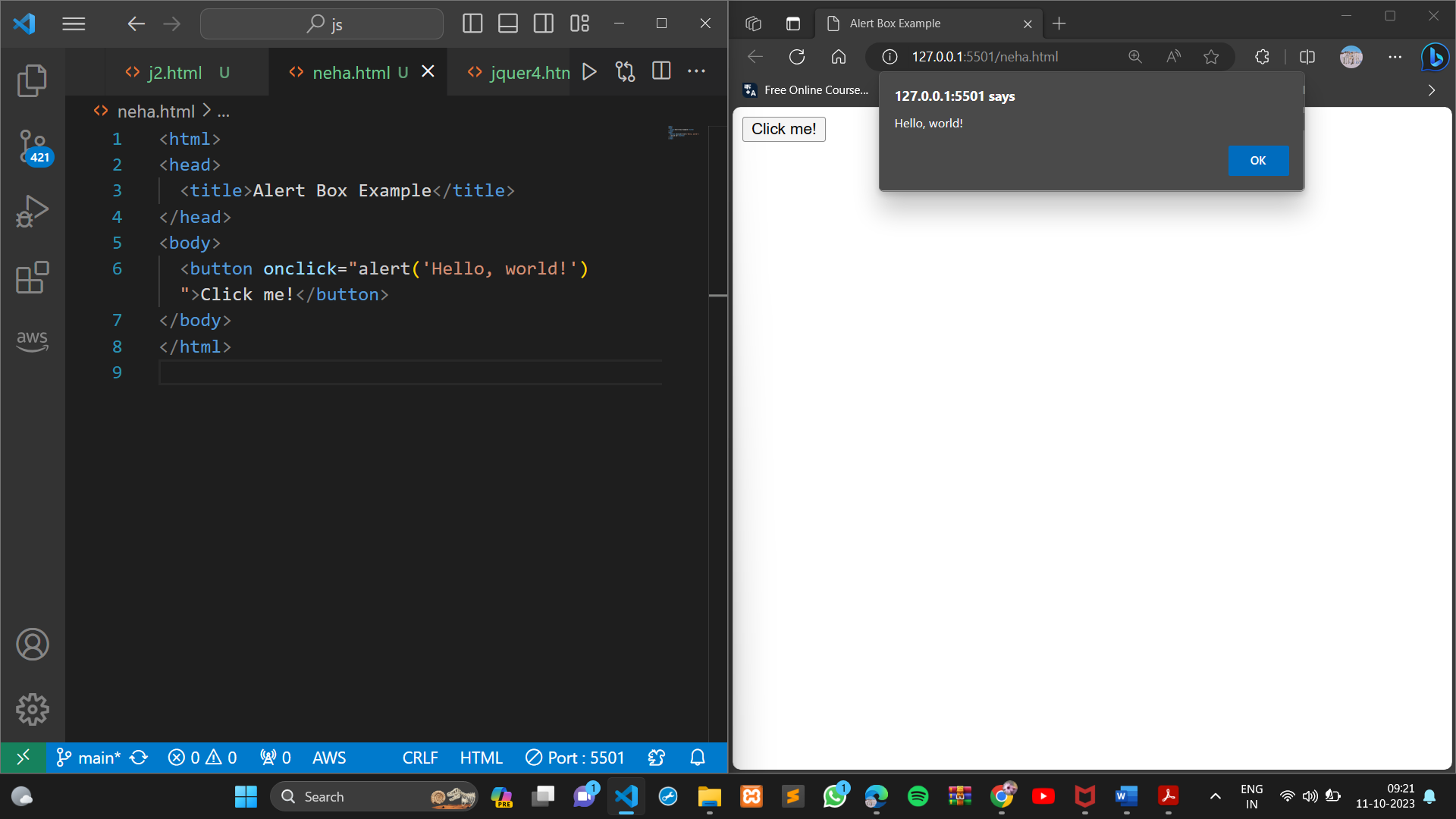
**Advance JavaScript**

**MODULE: 1 (Introduction and Code Quality)**

● Write a program to Show an alert



● What will be the result for these expressions?

5 > 4

"apple" > "pineapple"

"2" > "12"

undefined == null

undefined === null

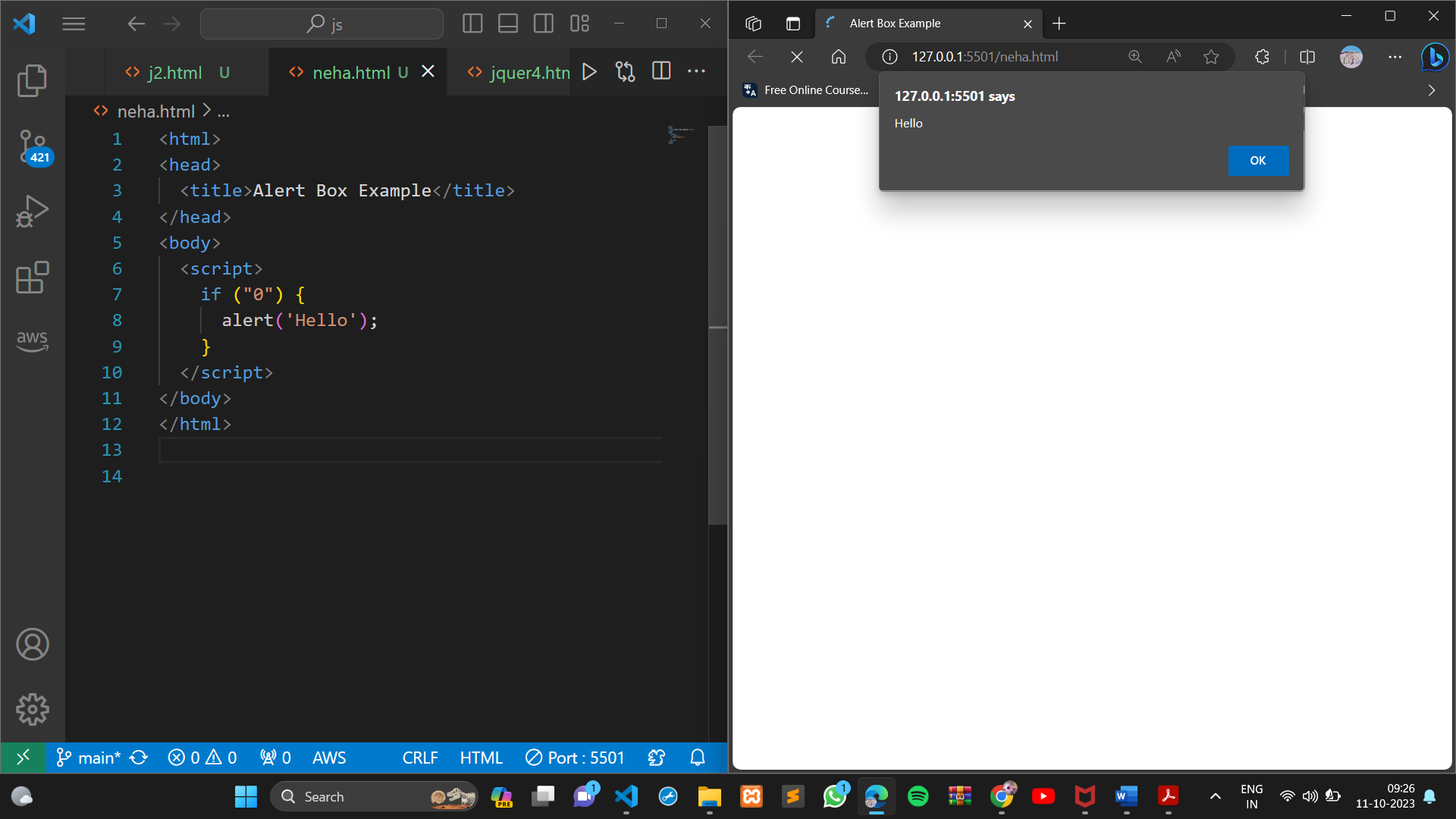
null == "\n0\n"

null === +"\n0\n"

* 5 > 4 is true because 5 is greater than 4.
* "apple" > "pineapple" is false because "apple" comes before "pineapple" in the dictionary.
* "2" > "12" is true because the first character of "2" is greater than the first character of "12".
* undefined == null is true because undefined and null are both falsy values.
* undefined === null: This uses strict equality, which checks both value and data type. Since undefined and null are of different data types, the result is false.
* null == "\n0\n": This is a loose equality comparison. null is not equal to any non-null value, so the result is false.
* null === +"\n0\n": Here, +"\n0\n" converts the string "\n0\n" to a number, which is 0. The strict equality comparison checks both value and data type, and since null is not equal to 0 in terms of value or data type, the result is false.

● Will alert be shown?

if ("0") { alert( 'Hello'); }



Yes, the alert box will be shown. In JavaScript, any non-empty string is considered to be true in a Boolean context. This means that the condition if ("0") will evaluate to true, and the alert box will be shown.

● What is the code below going to output? alert( null || 2 || undefined );

output the value 2. This is because the logical OR (||) operator evaluates its operands from left to right, and returns the first operand that is truthy. In this case, the first operand is null, which is false. The second operand is 2, which is true.

● The following function returns true if the parameter age is greater than

18. Otherwise it asks for a confirmation and returns its result:

function

checkAge(age)

{

if (age> 18) { return true; }

else {

// ...return confirm (‘did parents allow you?');

}

}

Yes, your understanding of the function checkAge() is correct. The function returns true if the parameter age is greater than 18. Otherwise, it asks the user for confirmation by displaying a confirmation dialog box with the message "Did parents allow you?". The function then returns the result of the confirmation dialog box.

● Replace Function Expressions with arrow functions in the code below:

Function

ask(question, yes, no)

{ if (confirm(question))yes();

else

no();

}

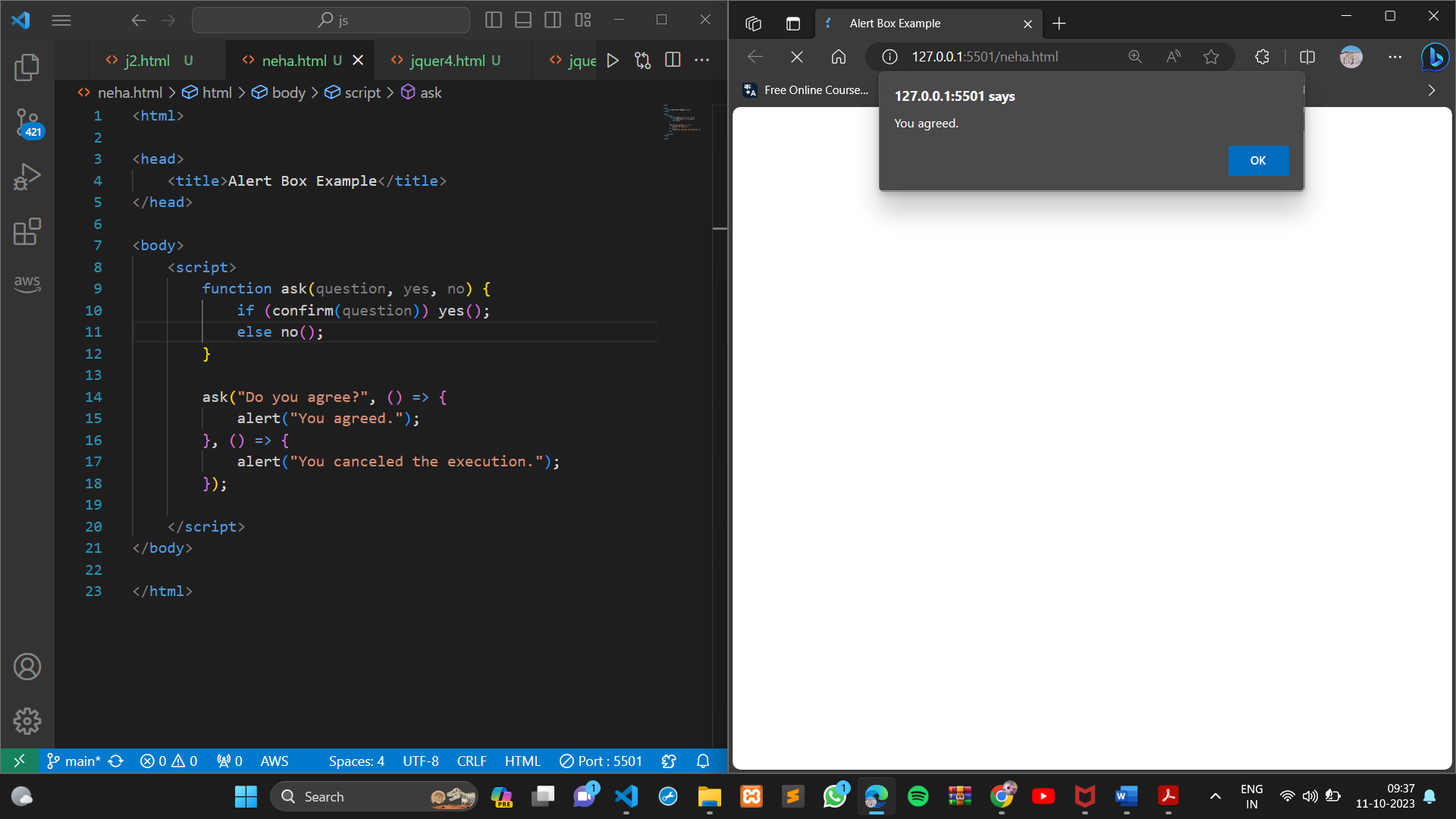
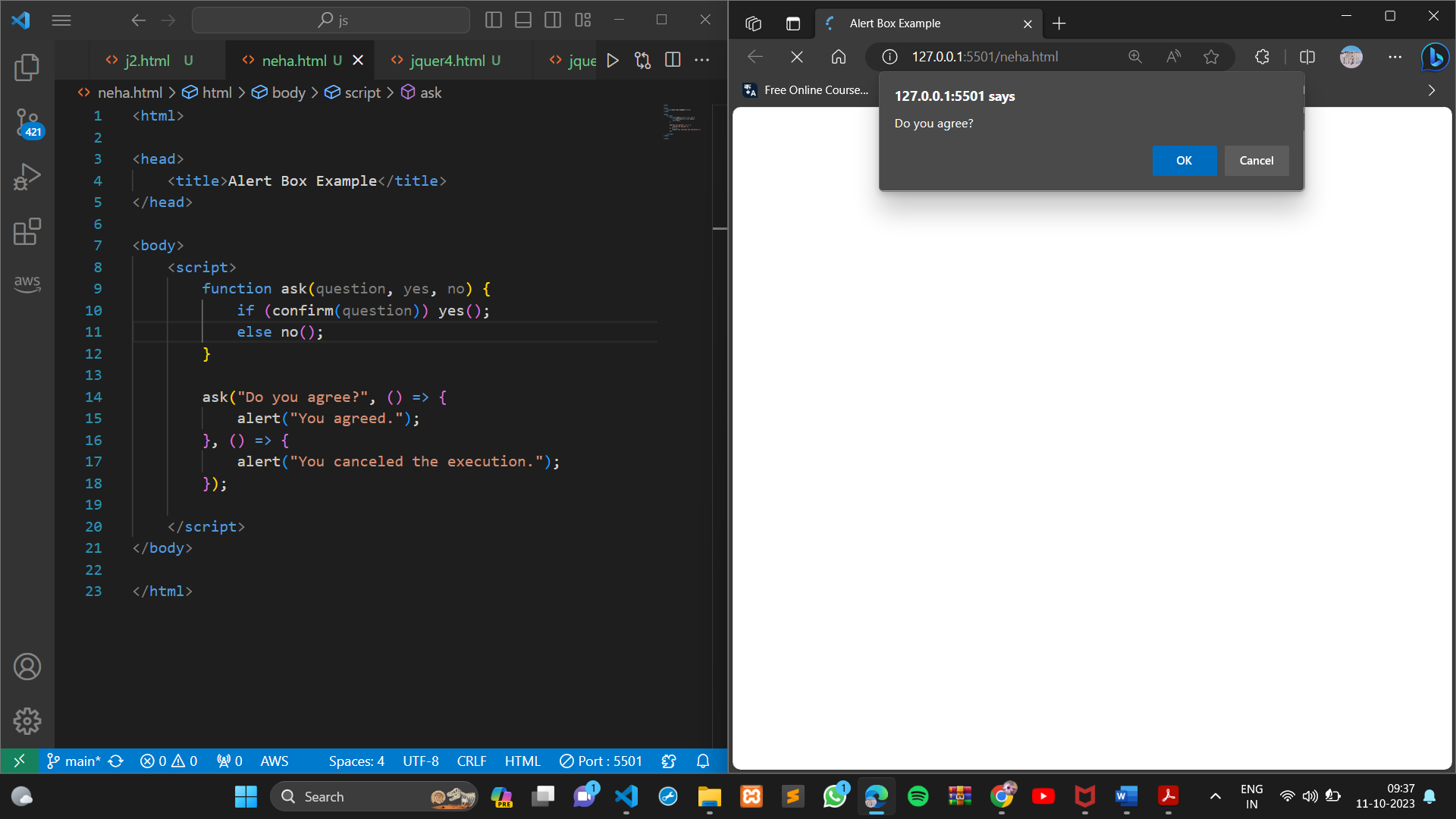
ask("Do you agree?", function()

{ alert("You agreed."); },

function() {

alert("You canceled the execution."); }

}



**MODULE: 2 (Data Types and Objects)**

● Write the code, one line for each action:

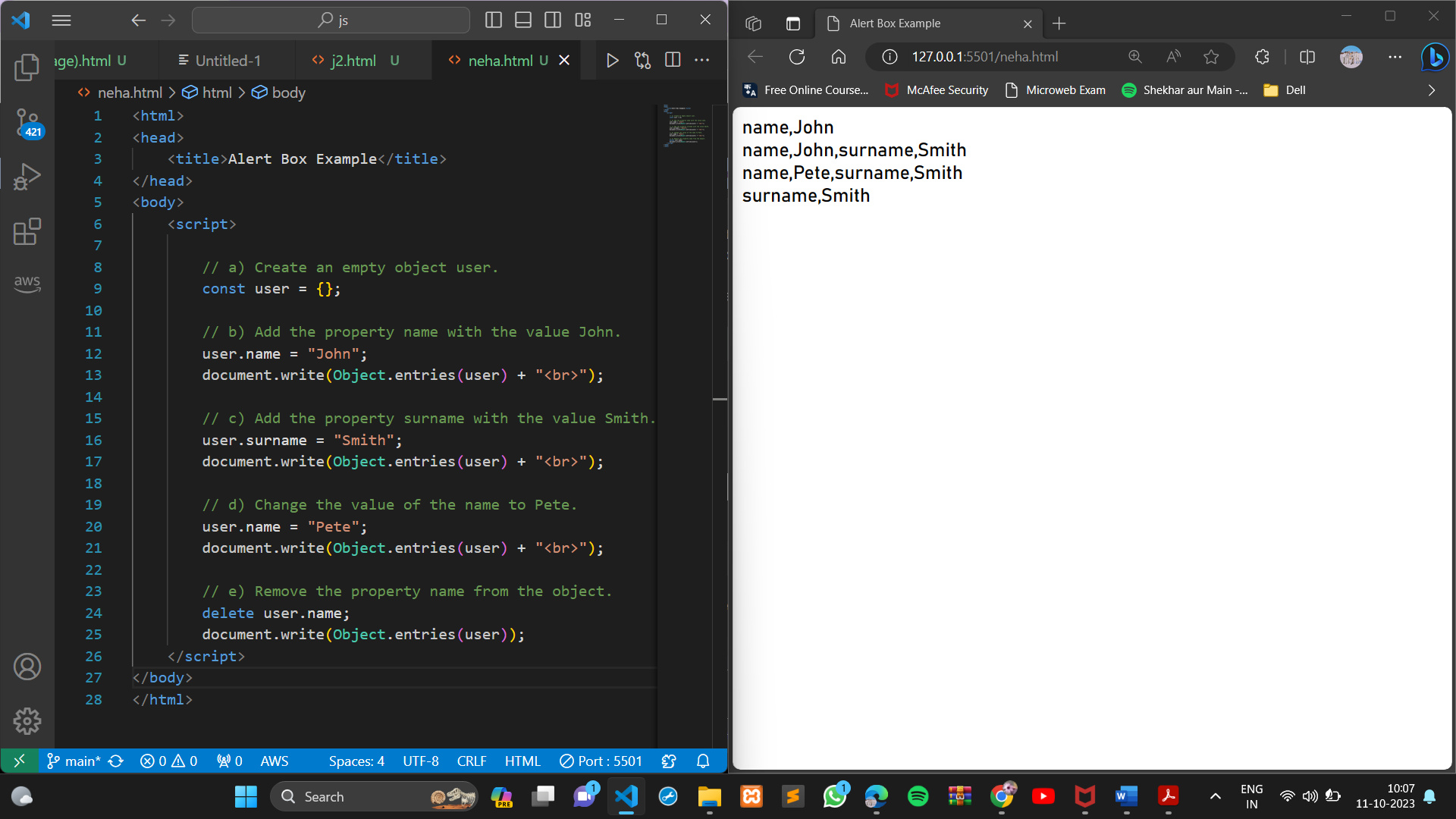
Create an empty object user.

Add the property name with the value John.

Add the property surname with the value Smith.

Change the value of the name to Pete.

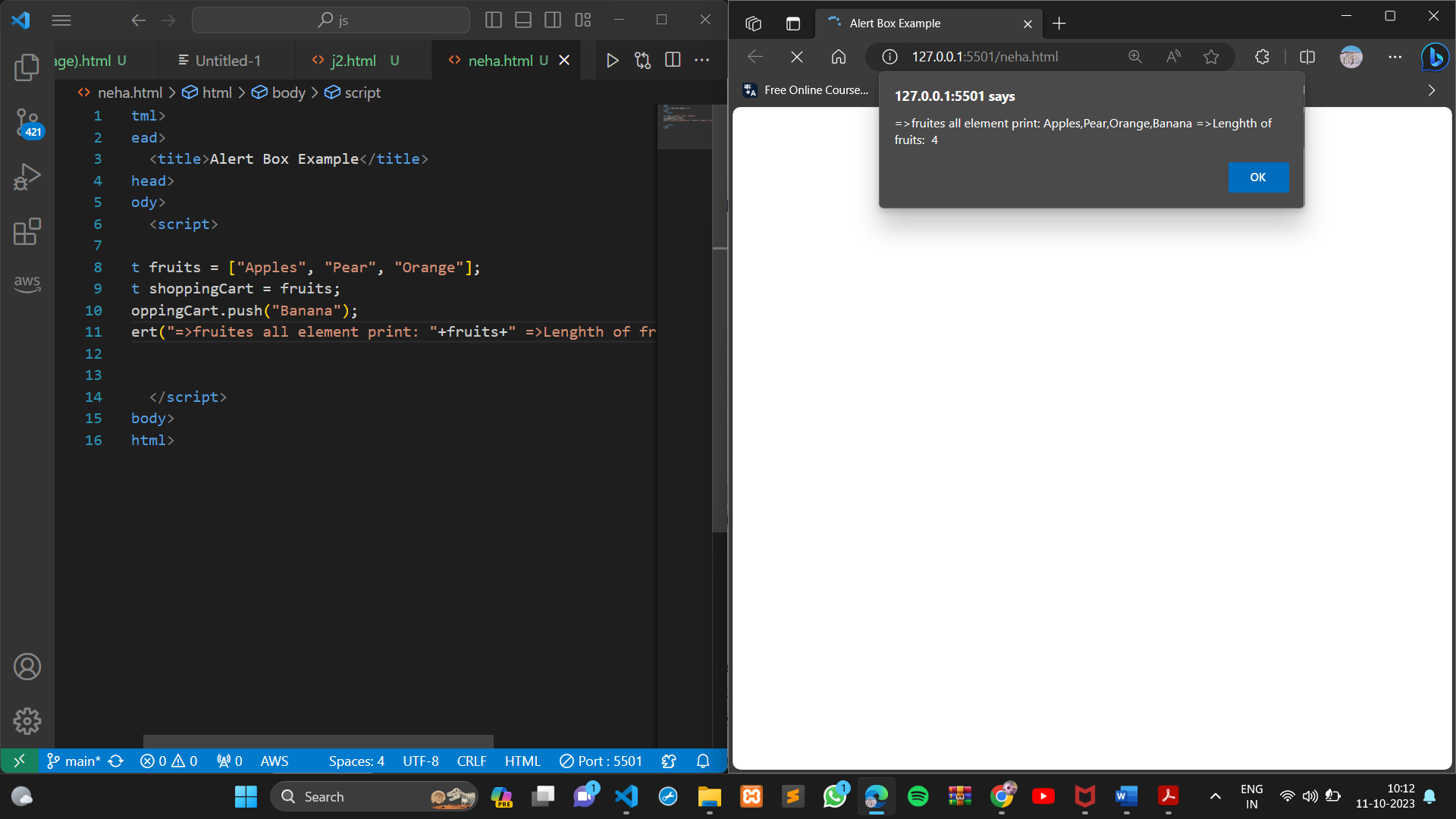
Remove the property name from the object.



● Is array copied?

let fruits = ["Apples", "Pear", "Orange"]; // push a new value into the "copy" let shoppingCart = fruits; shoppingCart.push("Banana"); // what's in fruits?

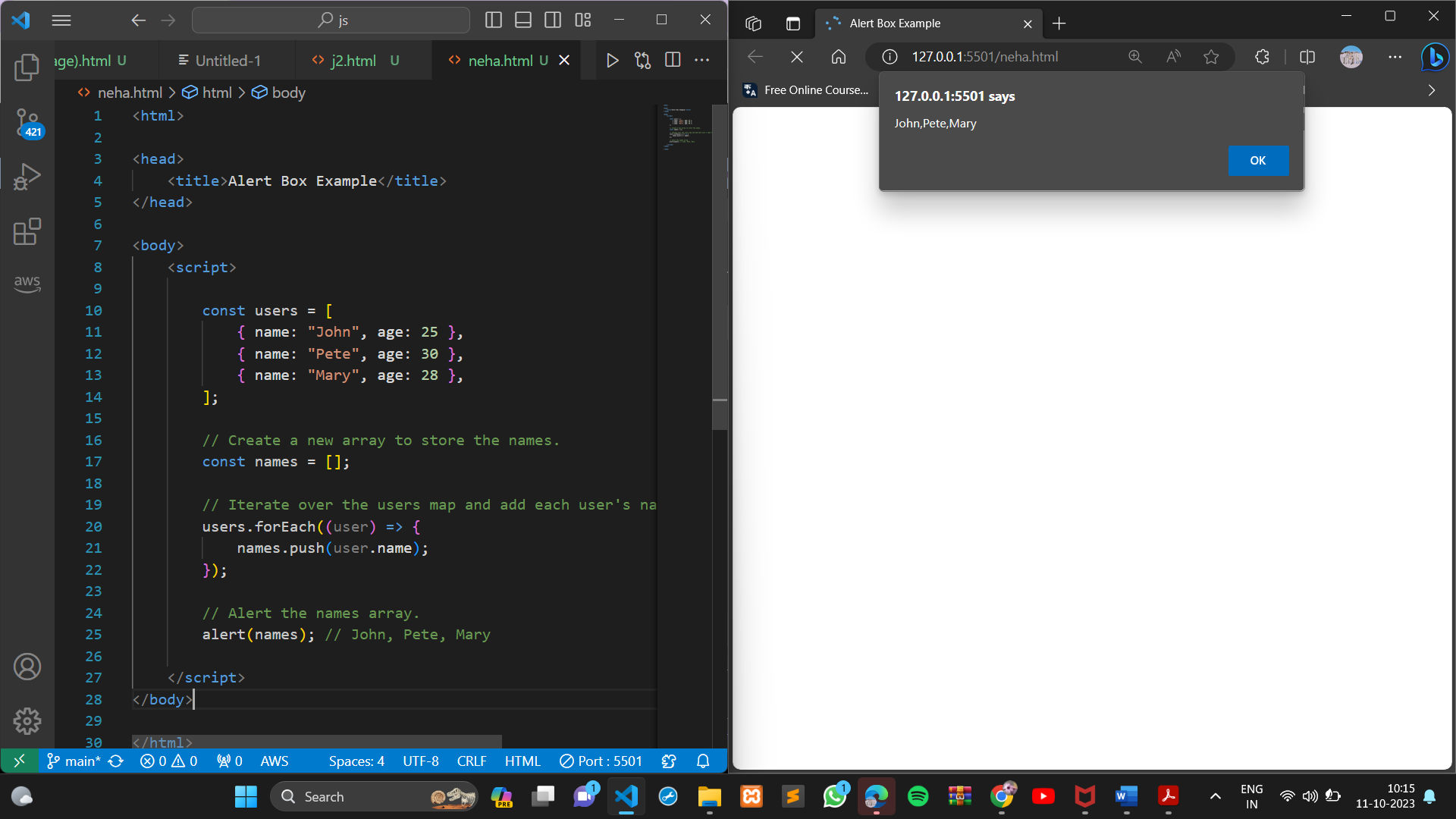
alert( fruits.length ); // ?



he array fruits is not copied when you assign it to the variable shoppingCart. Both variables point to the same array object in memory. This means that when you push a new value into the shoppingCart array, you are also pushing it into the fruits array.

● Map to names

let john = { name: "John", age: 25 }; let pete = { name: "Pete", age: 30 }; let mary = { name: "Mary", age: 28 }; let users = [ john, pete, mary ]; let names = /\* ... your code \*/ alert( names ); // John, Pete, Mary



● Map to objects

let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [ john, pete, mary ]; let usersMapped = /\* ... your code ... \*/

/\*

usersMapped = [

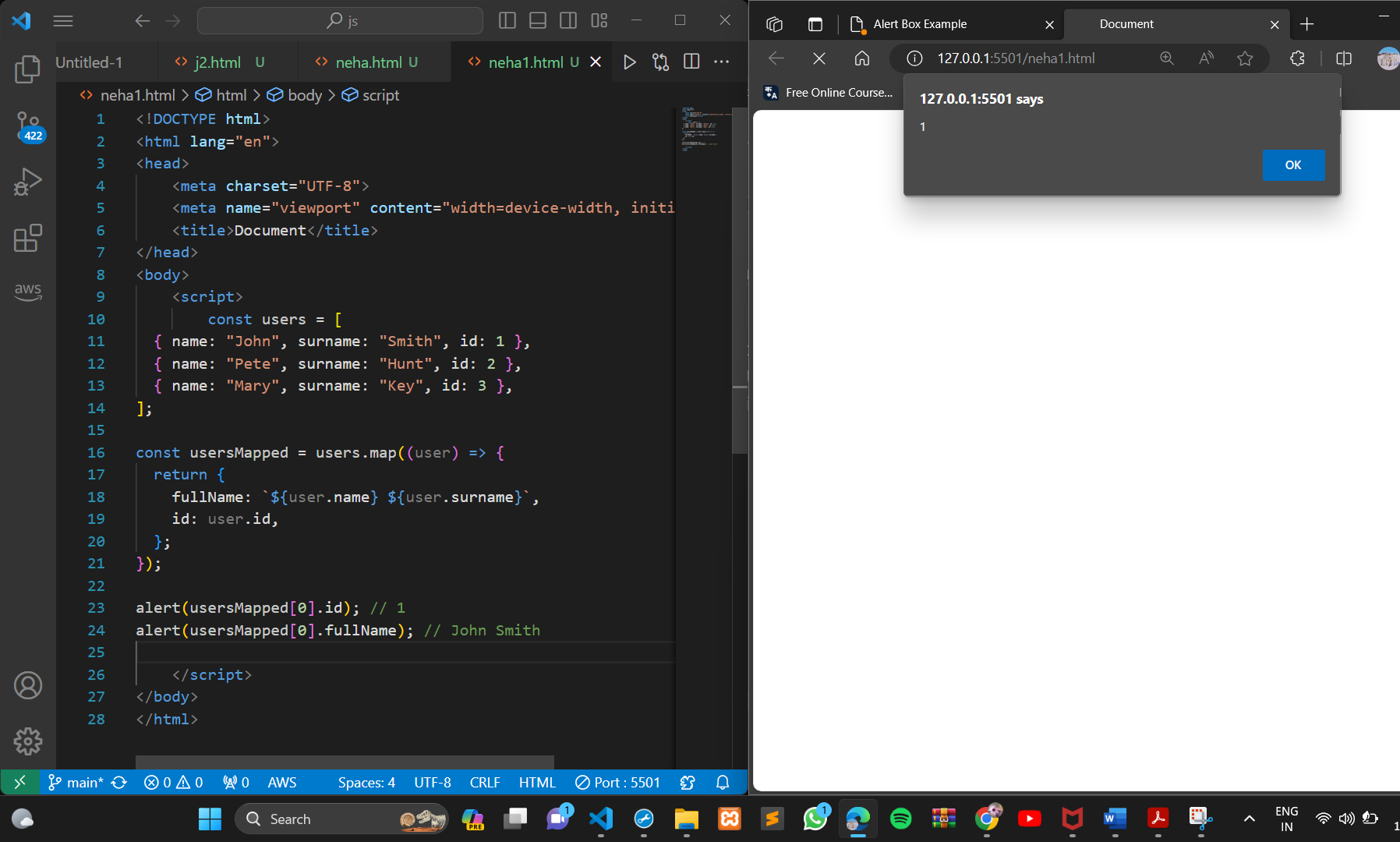
{ fullName: "John Smith", id: 1 },

{ fullName: "Pete Hunt", id: 2 },

{ fullName: "Mary Key", id: 3 }

]

\*/ alert( usersMapped[0].id ) // 1 alert( usersMapped[0].fullName ) // John Smith





● Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0.

let salaries = {

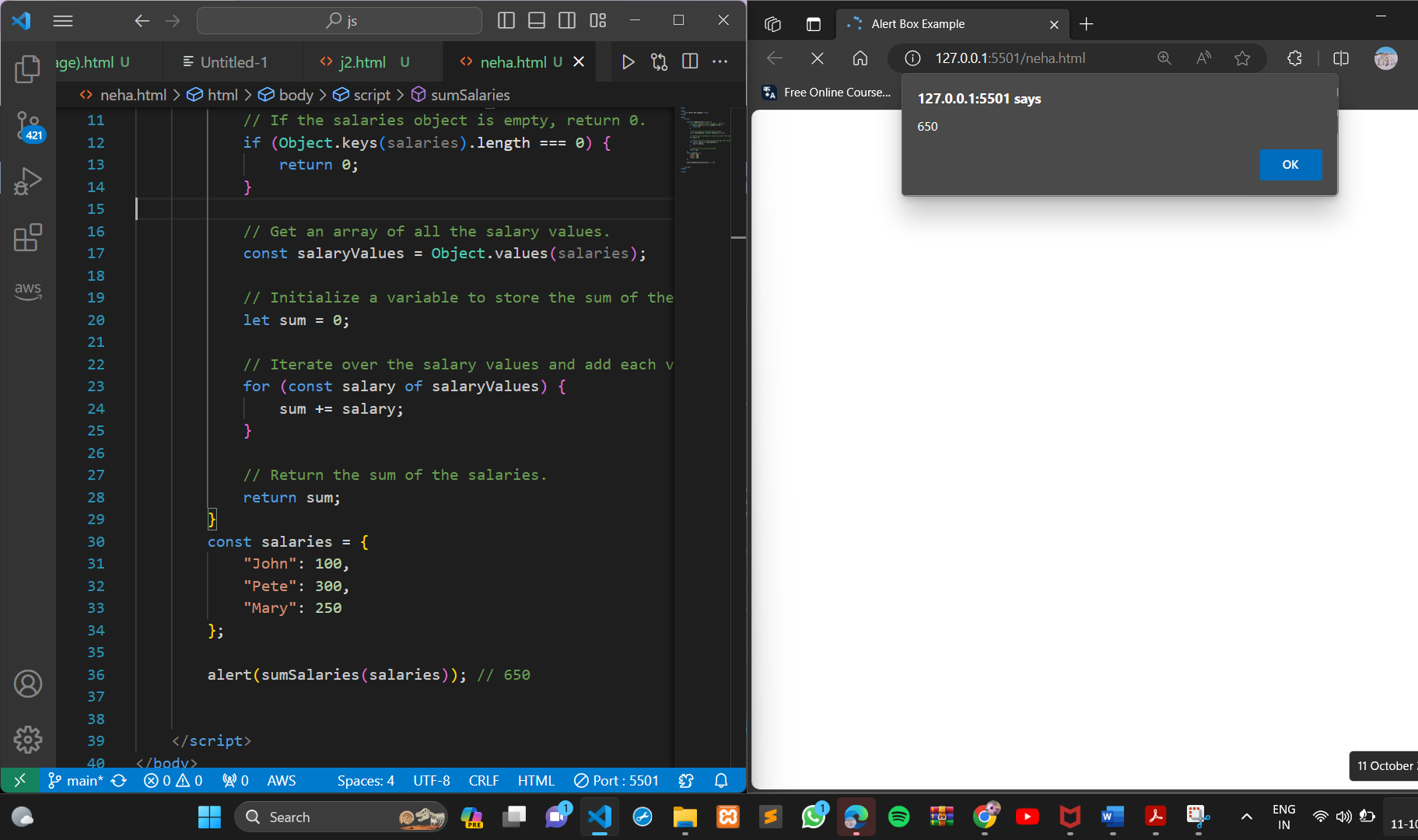
"John": 100,

"Pete": 300,

"Mary": 250

};

alert( sumSalaries(salaries) ); // 650



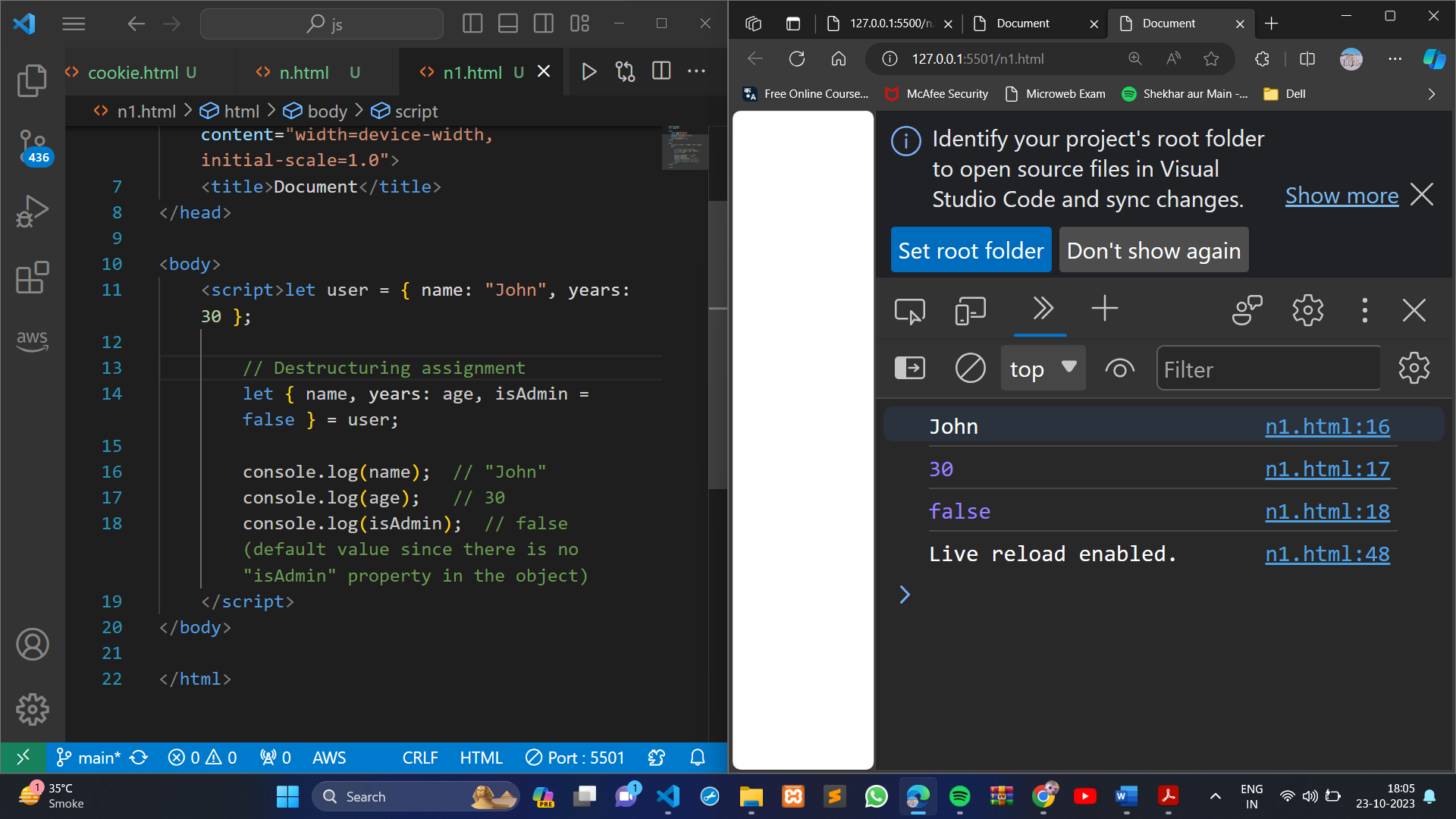
● Destructuring assignment We have an object: Write the Destructuring assignment that reads:

Name property into the variable name.

Year’s property into the variable age.

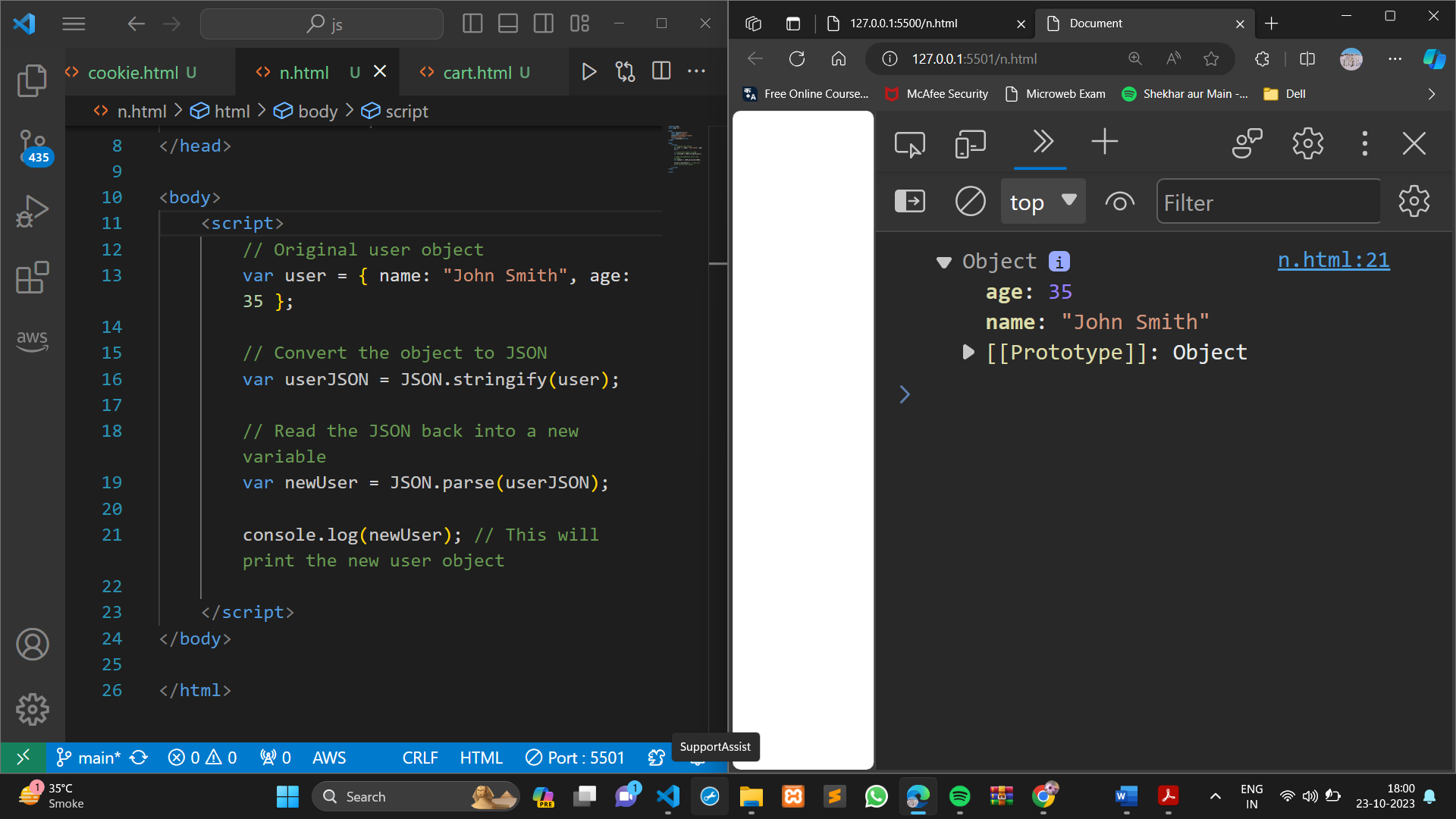
isAdmin property into the variable isAdmin (false, if no such property)

let user = { name: "John", years: 30};



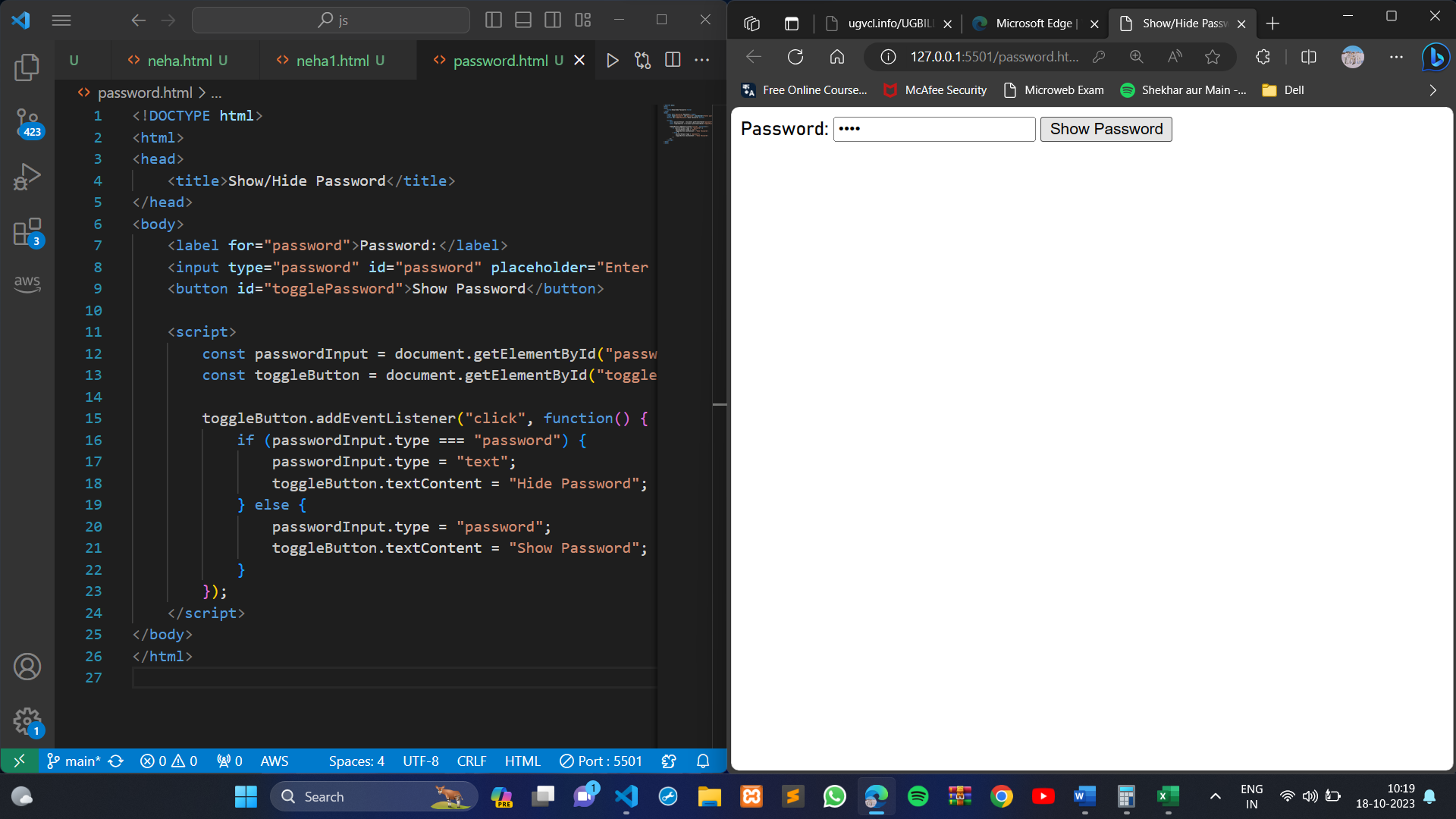
● Turn the object into JSON and back Turn the user into JSON and then read it back into another variable.

user = { name: "John Smith", age: 35};

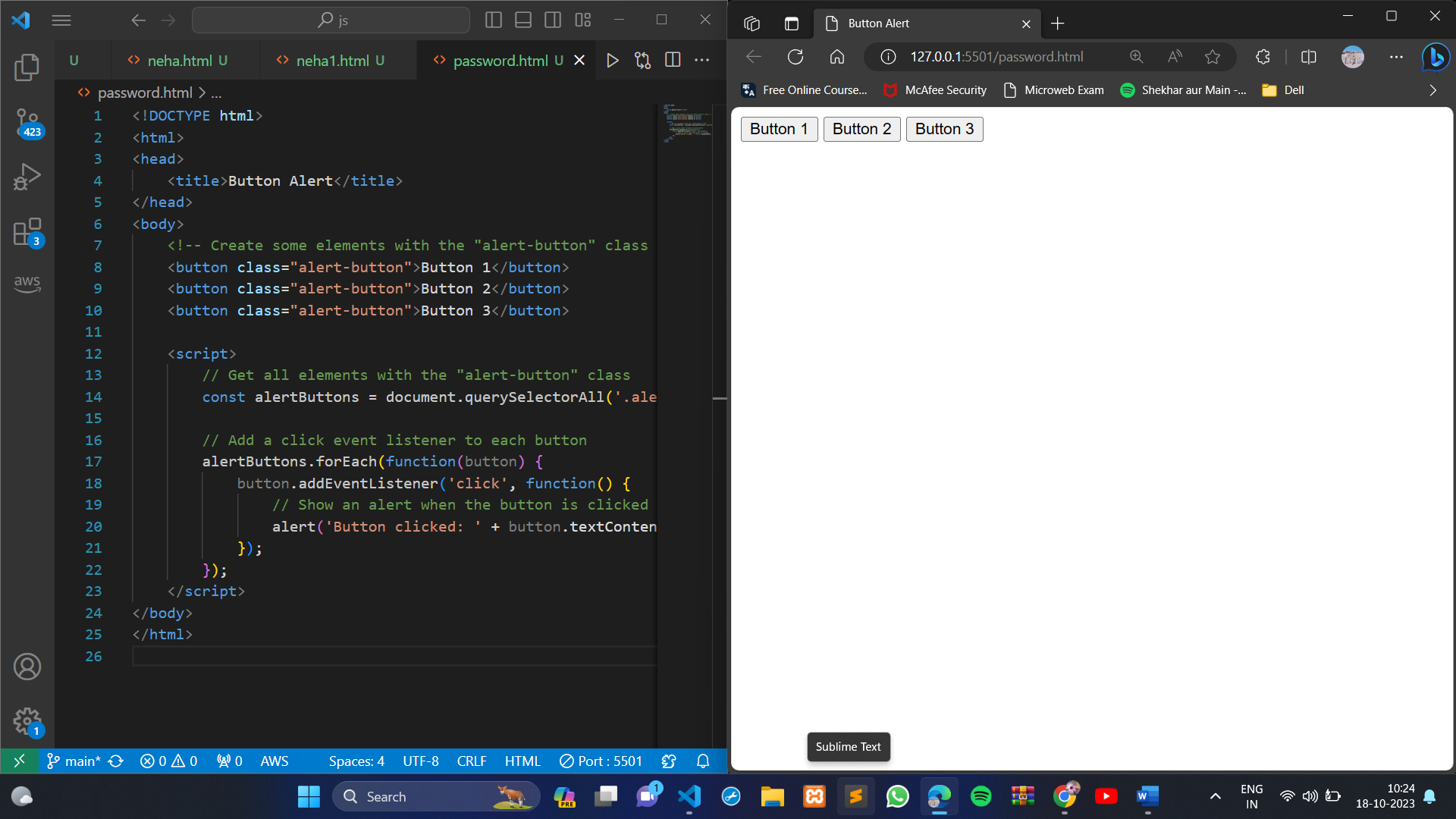


**MODULE: 3 (Document, Event and Controls)**

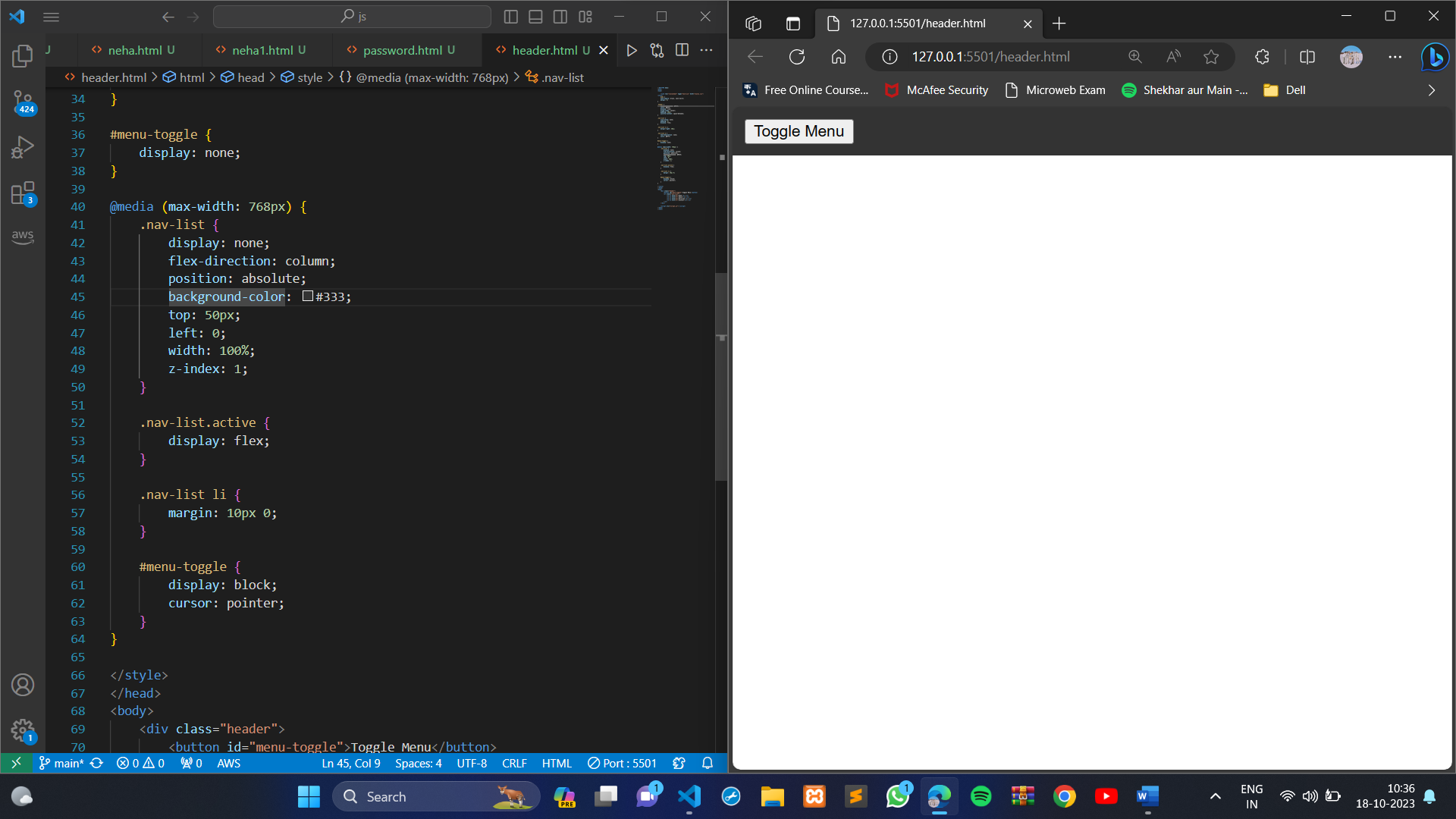
● Create a program to hide/show the password



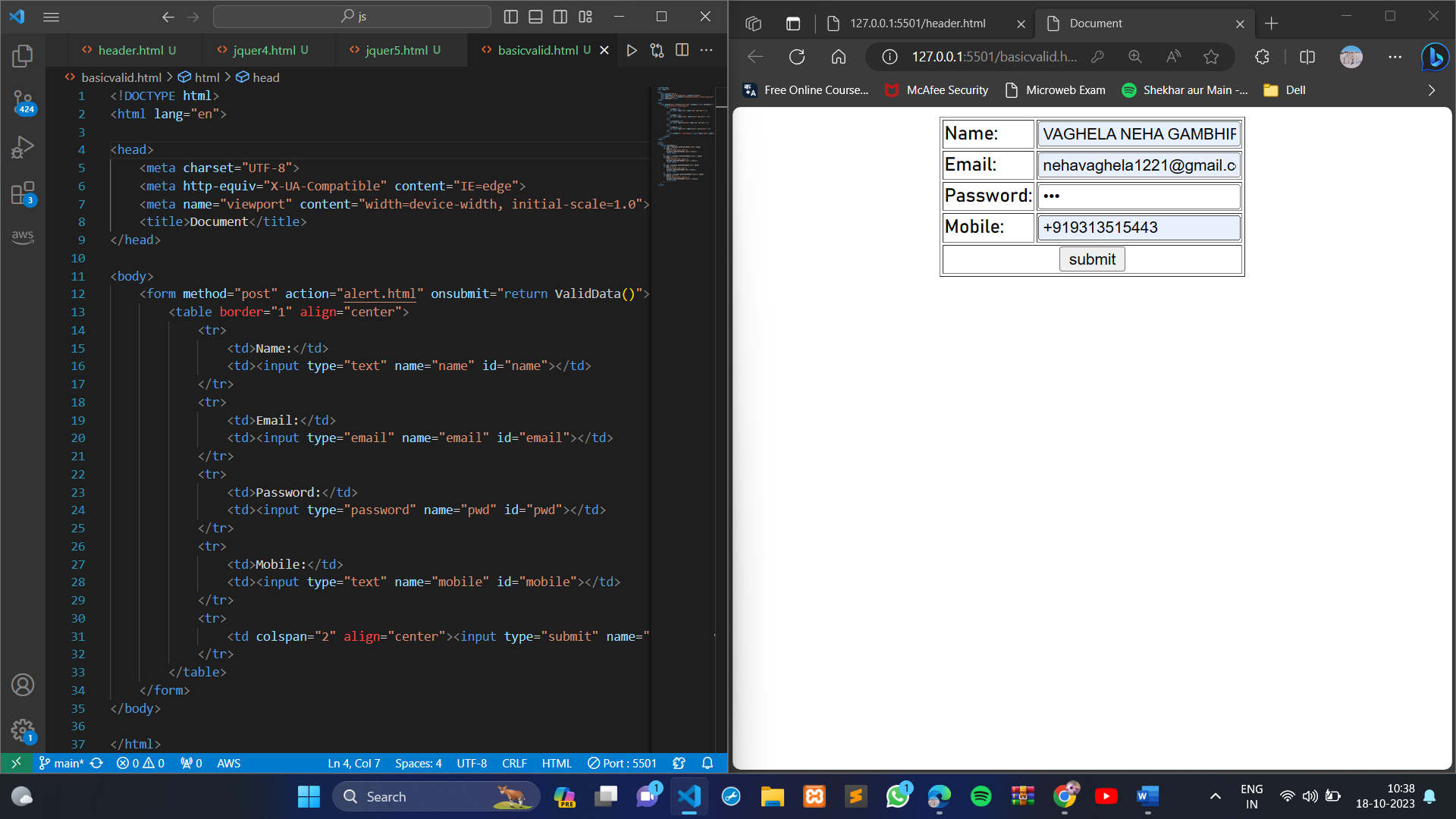
● Create a program that will select all the classes and loop over and whenever i click the button the alert should show



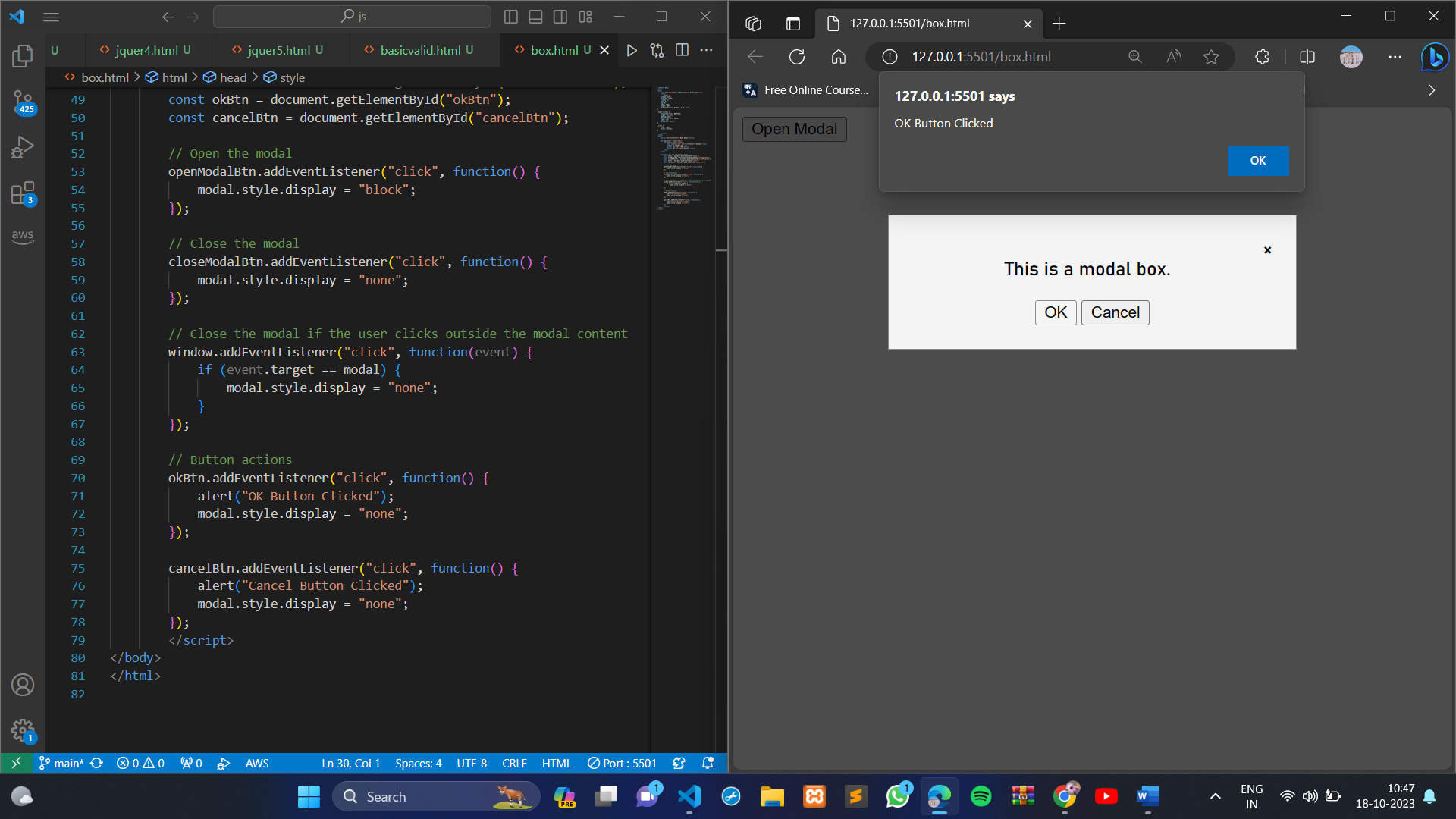
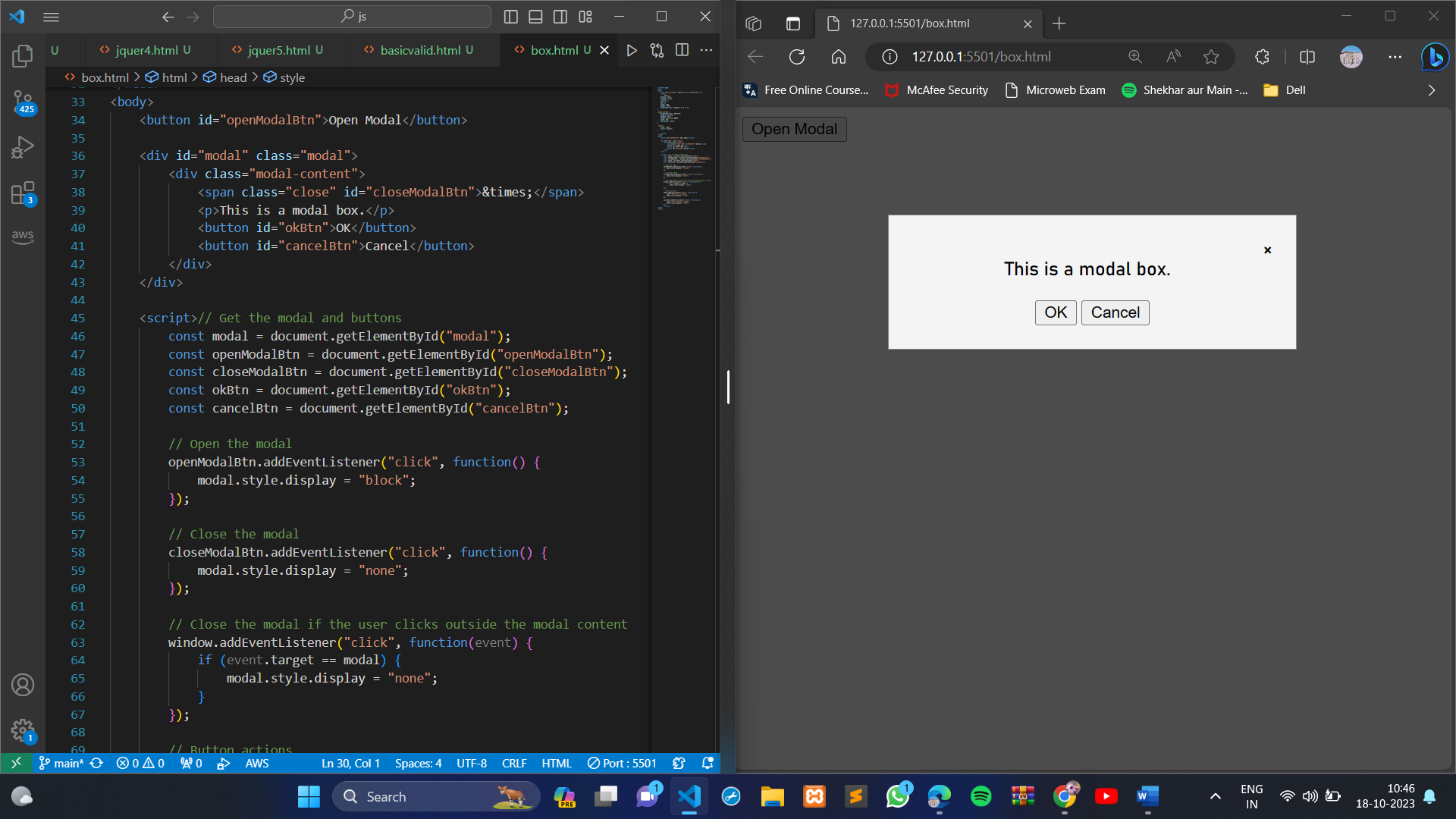
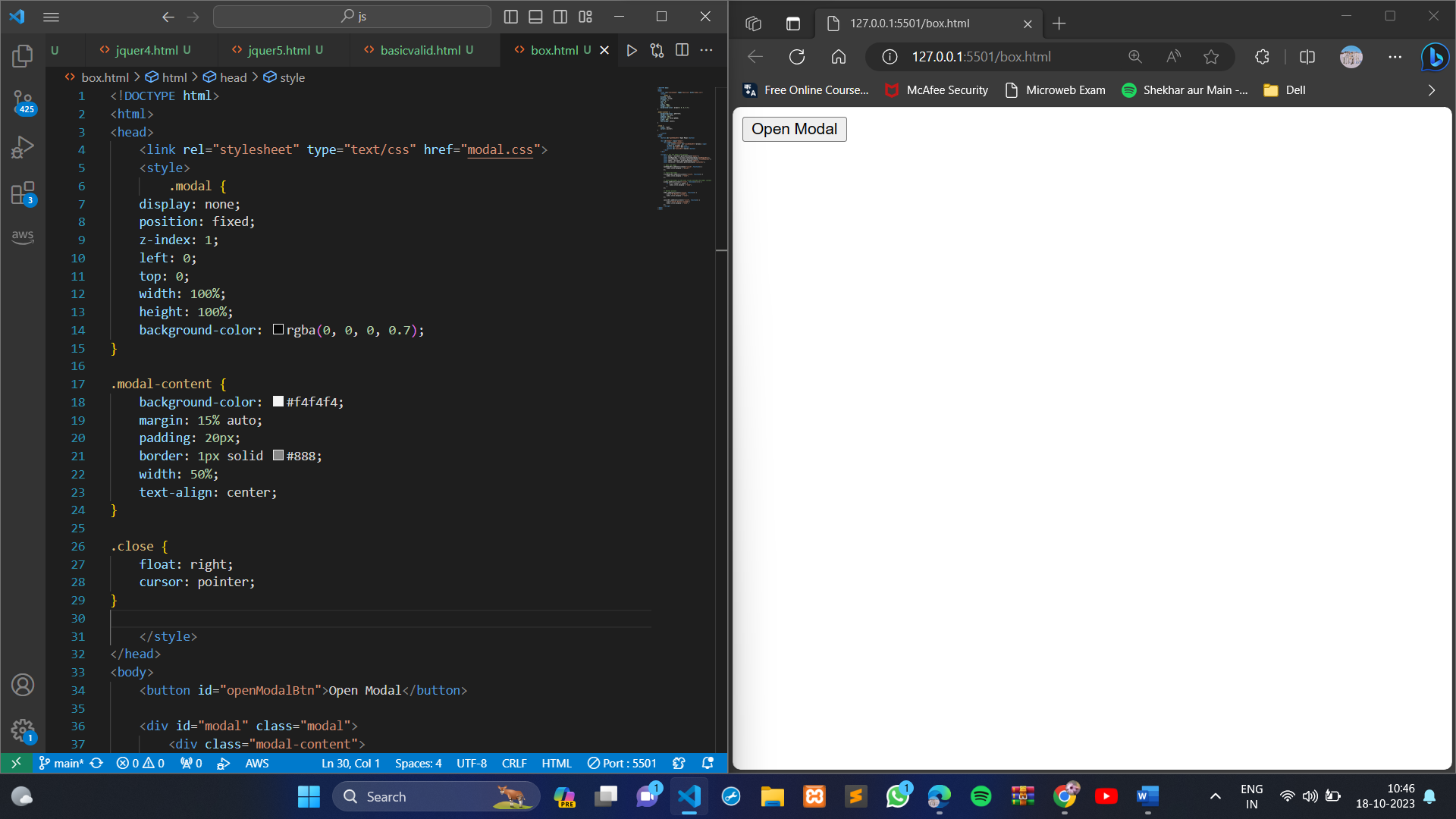
● Create a responsive header using proper JavaScript



● Create a form and validate using JavaScript

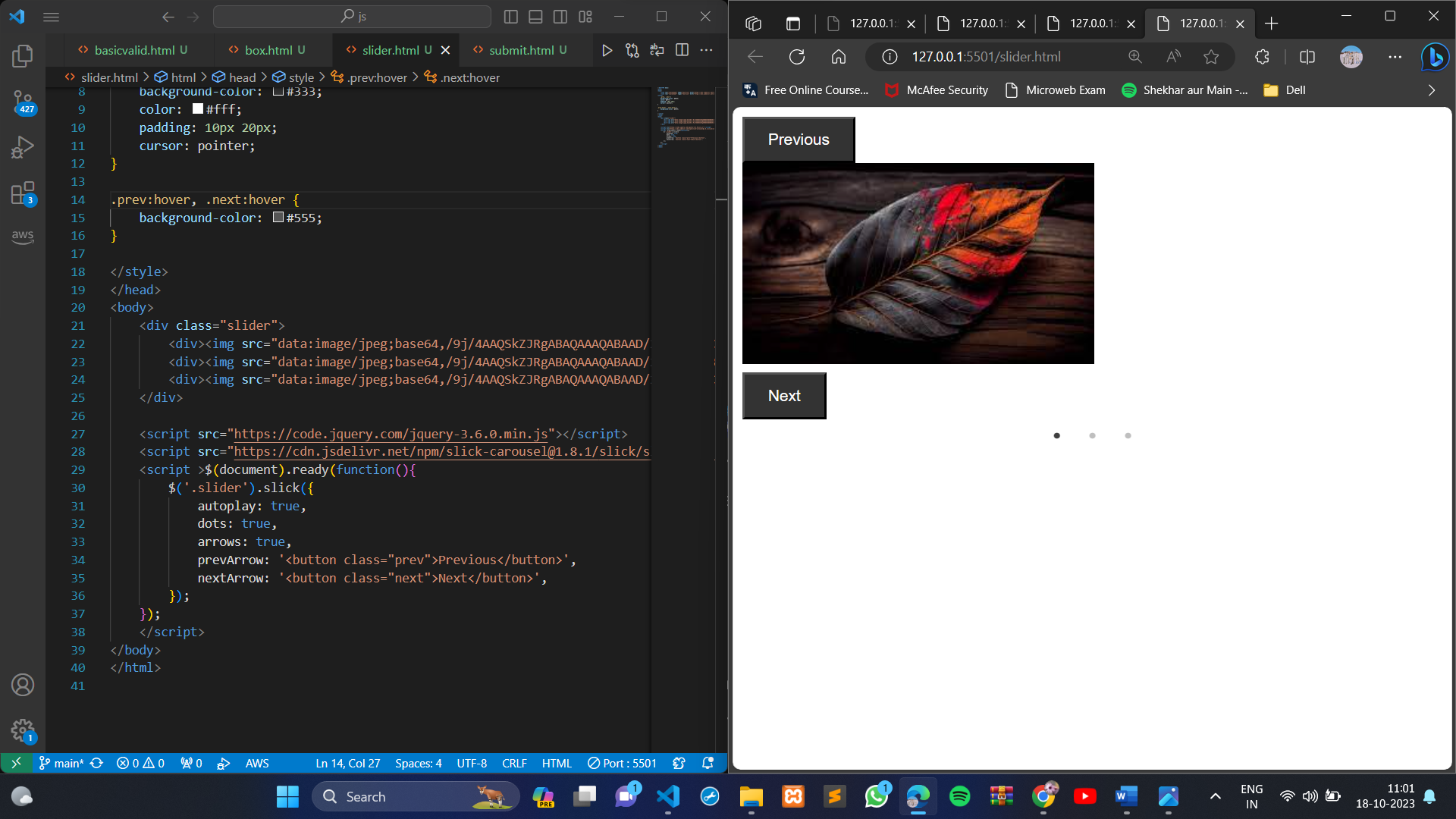


● Create a modal box using css and Js with three buttons

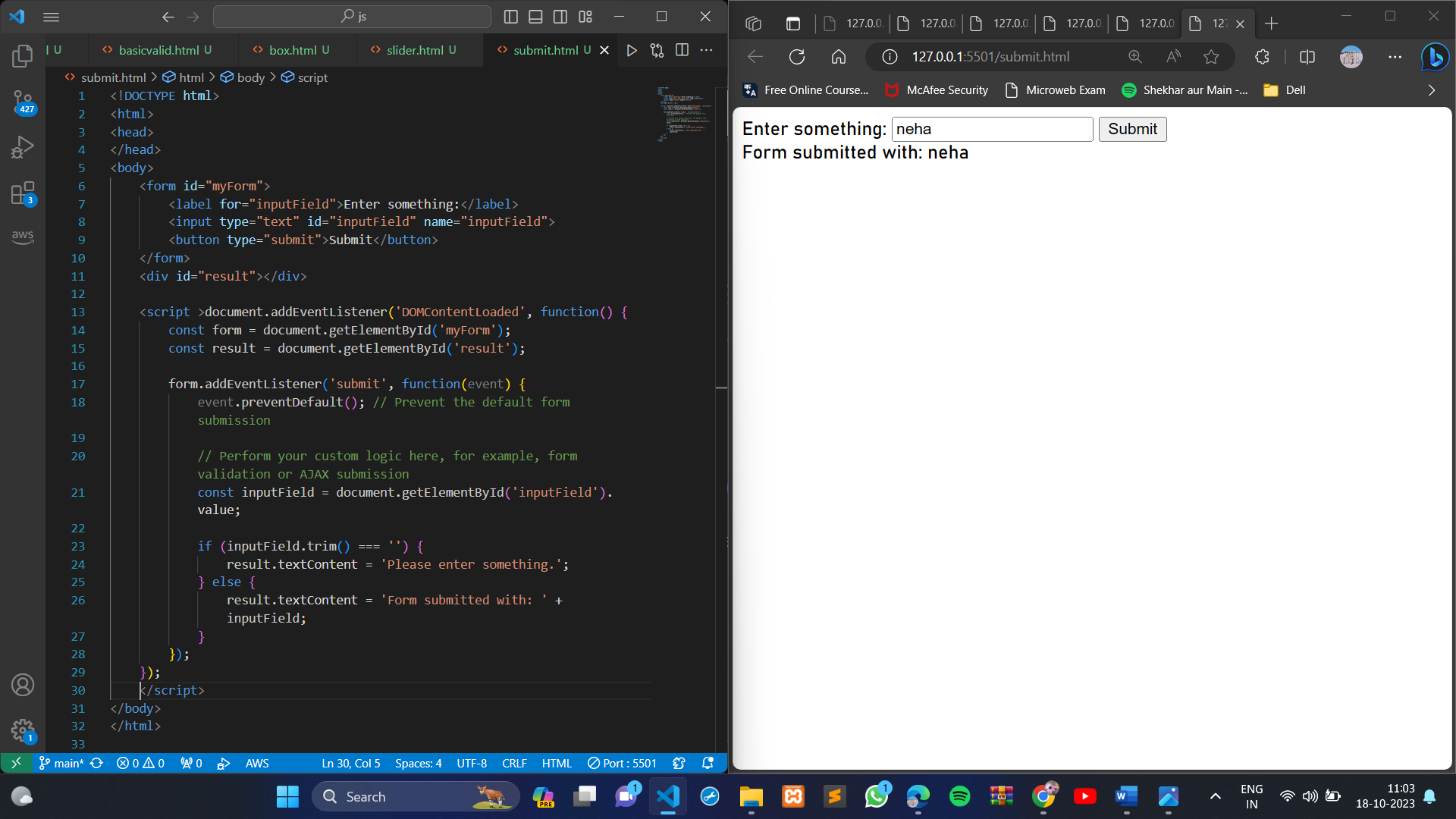


This code creates a modal box with "Open Modal," "OK," and "Cancel" buttons. When you click the "Open Modal" button, the modal will appear, and you can click either the "OK" or "Cancel" button to trigger respective actions and close the modal. Clicking the close button (X) or outside the modal will also close it.

● Use external js library to show slider



● Prevent the browser when i click the form submit button



In this example, we're using JavaScript to capture the form's submit event and prevent the default behavior using **event.preventDefault()**. We then perform custom logic, such as form validation or handling the form submission. If the input field is empty, we display a message; otherwise, we indicate that the form has been submitted with the input value.

**MODULE: 4 (New Request)**

● What is JSON

JSON stands for JavaScript Object Notation. It is a lightweight data interchange format that is easy to read and write for both humans and machines. JSON is commonly used in web applications to exchange data between the client and server.

JSON is a subset of JavaScript syntax, but it is also language-independent. This means that JSON data can be parsed and used by any programming language, not just JavaScript.

JSON data is represented as a string of text. This makes it easy to send and receive JSON data over the network or store it in a file.

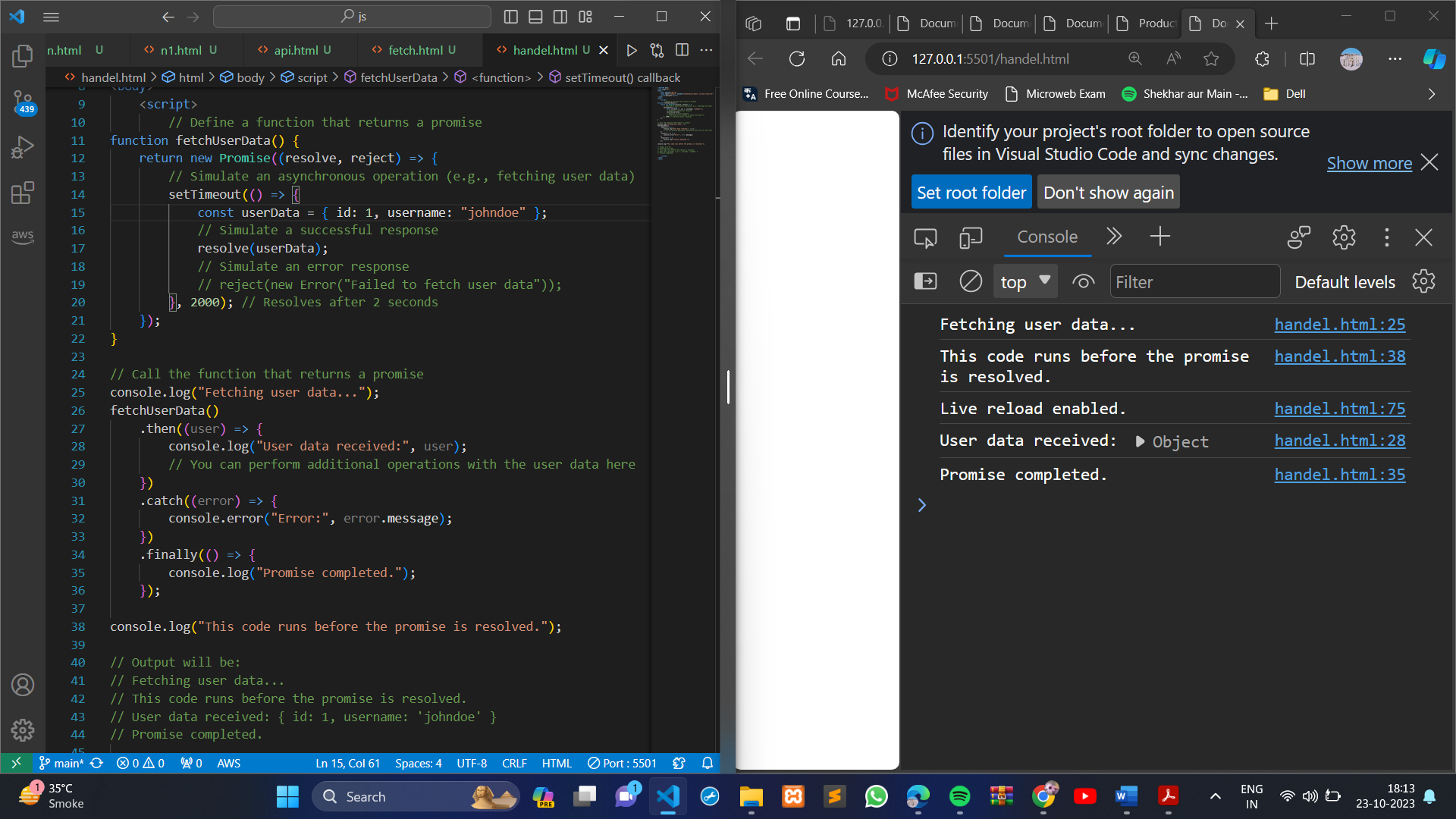
● What is promises

A promise is an object that represents the eventual completion or failure of an asynchronous operation. Promises are used to handle asynchronous operations in a clean and concise way.

Promises have three states:

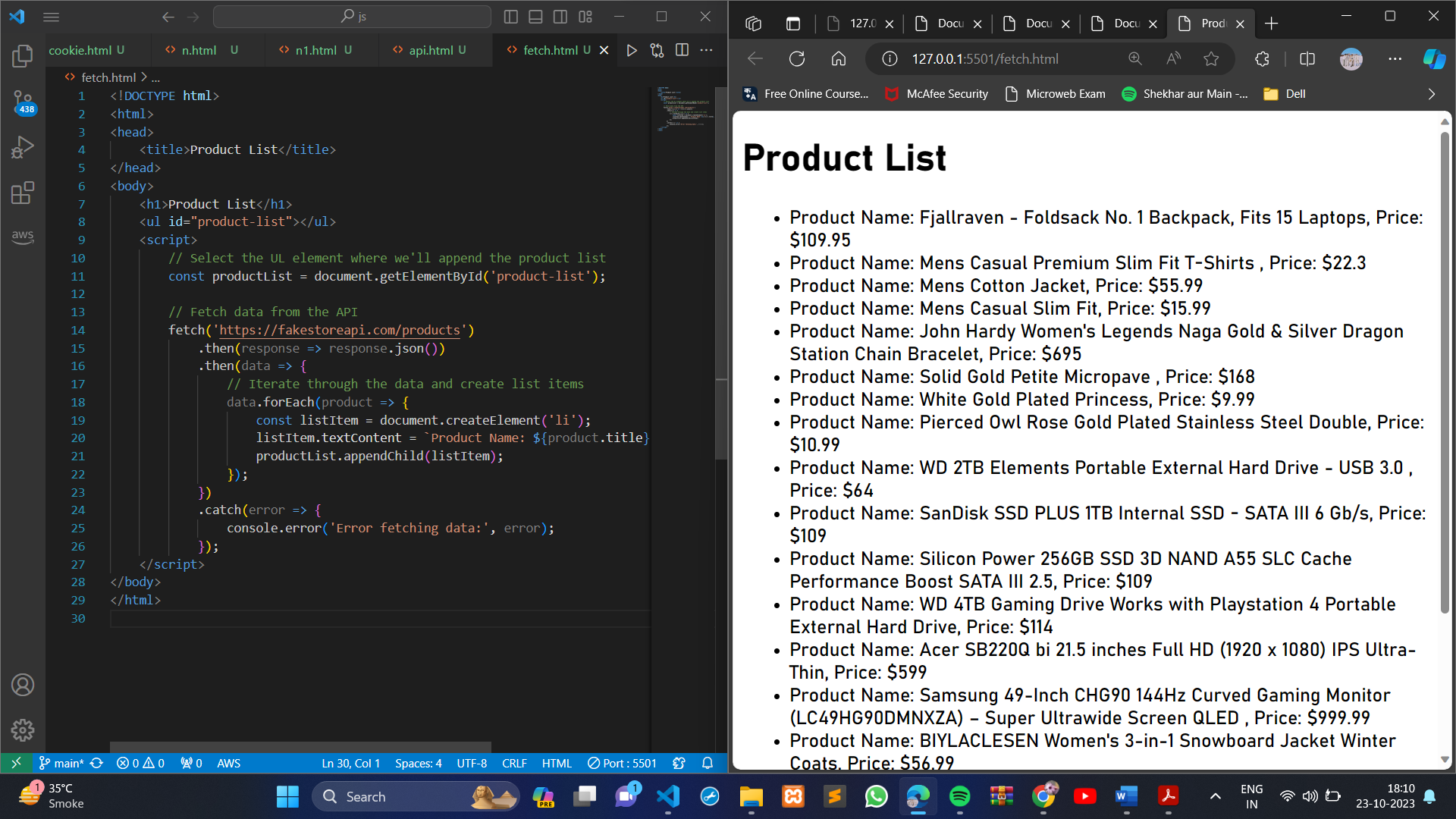
* Pending: This is the initial state of a promise. It means that the asynchronous operation has not yet completed.
* Fulfilled: This means that the asynchronous operation completed successfully.
* Rejected: This means that the asynchronous operation failed.

● Write a program of promises and handle that promises also



1. We define a function **fetchUserData** that returns a promise. Inside the promise, we simulate an asynchronous operation that resolves with user data or rejects with an error.
2. We call **fetchUserData()** and use **.then()** to handle the successful resolution of the promise, **.catch()** to handle errors, and **.finally()** to perform actions that should happen regardless of the promise's outcome.
3. While the promise is being resolved asynchronously, the code continues to execute, as shown by "This code runs before the promise is resolved."

● Use fetch method for calling an api <https://fakestoreapi.com/products>



● Display all the product from the api in your HTML page 