Contents

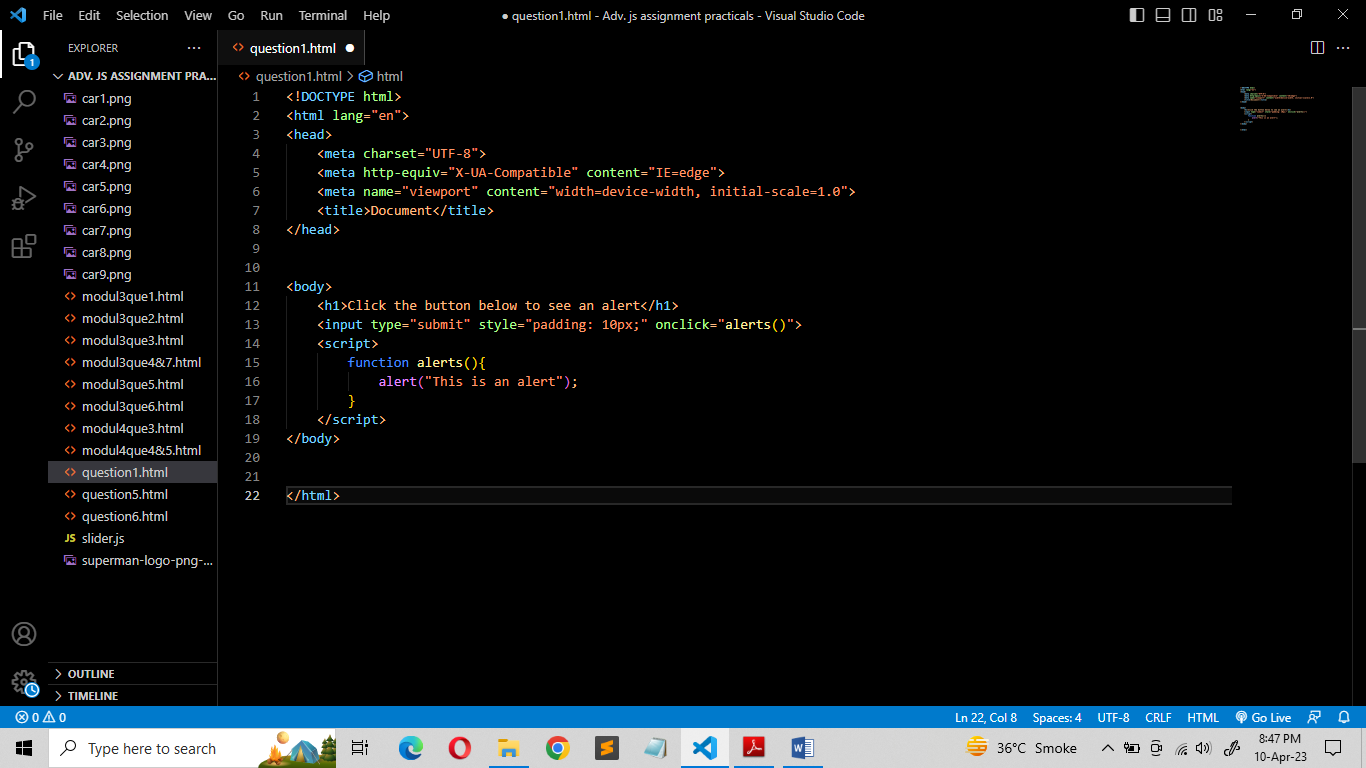
[**MODULE: 1 (Introduction and Code Quality)** 2](#_Toc132054356)

[**MODULE: 2 (Data Types and Objects)** 4](#_Toc132054357)

[**MODULE: 3 (Document, Event and Controls)** 6](#_Toc132054358)

[**MODULE: 4 (New Request)** 7](#_Toc132054359)

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

1. **Write a program to Show an alert**

**→**

1. **What will be the result for these expressions?**

**→**

1. 5 > 4

→ Result will be “true” because it is obvious that number 5 is greater than 4.

1. "apple" > "pineapple"

→ Result will be “false” because “a” is smaller than “p”.

1. "2" > "12"

→ Result will be “true” because first character of string “2” is greater than “1”.

1. undefined == null

→ Result will be “true” because values null and undefined are equal to each other.

1. undefined === null

→ Result will be “false” because different types of both sides lead to false.

1. null == "\n0\n"

→ Result will be “false” because null is only equals to undefined.

1. null === +"\n0\n"

→ Result will be “false” because strict equality of different types.

1. **Will alert be shown?**

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

**→** The code under this condition would never execute because number 0, an empty string “”, undefined and NaN all become false. Because of that they are called “falsy” values.

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

**→** The answer is 2. That is the first truthy value. Because if an operand is not a Boolean, it’s converted to a Boolean for the evaluation. That’s why number 2 is treated as true.

1. **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?'); }**

**}**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul1que5.html**

1. **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."); }**

**}**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul1que6.html**

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

**1) Write the code, one line for each action:**

1. Create an empty object user.

→ let user = {};

1. Add the property name with the value John.

→ user.name = “John”;

1. Add the property surname with the value Smith.

→ user.surname = “Smith”;

1. Change the value of the name to Pete.

→ user.name = “Pete”;

1. Remove the property name from the object.

→ delete user.name;

**2) 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 ); // ?**

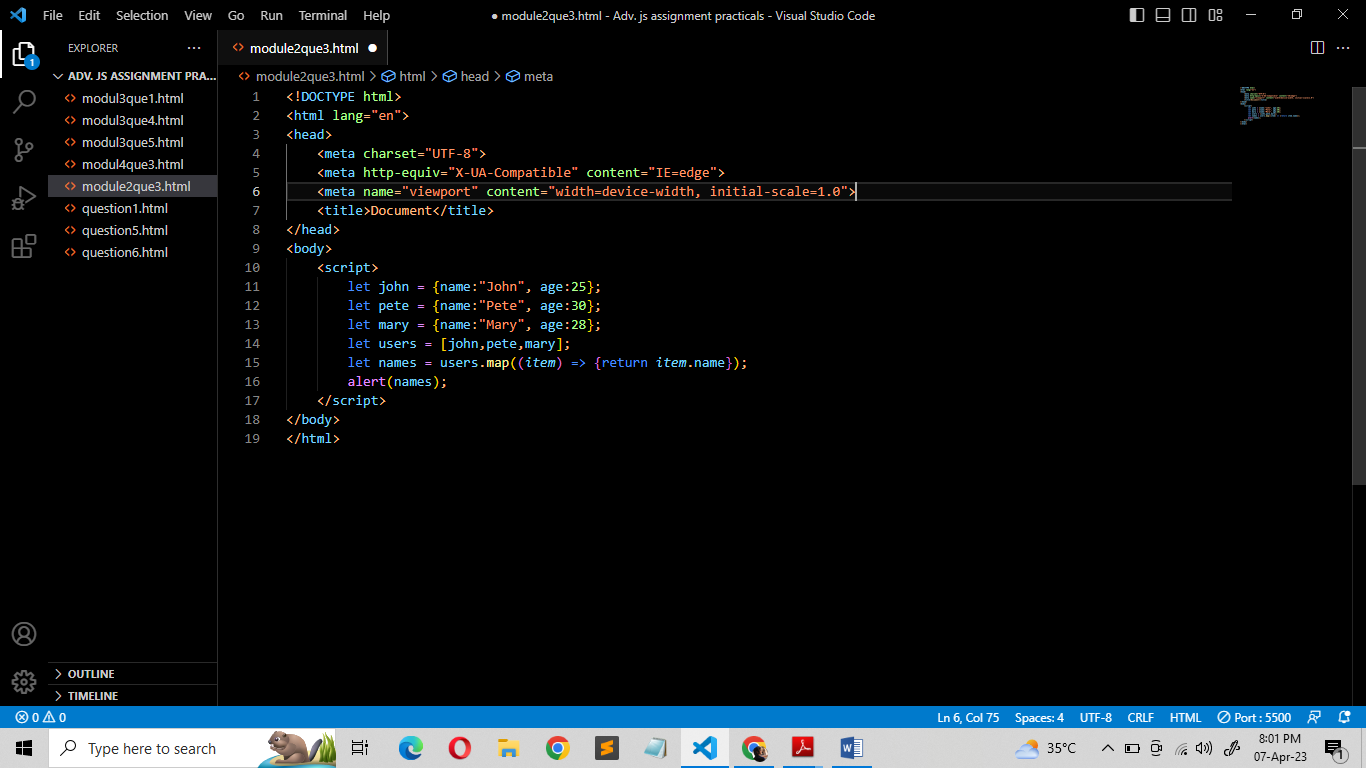
**→** Yes array is copied.

Fruits = ["Apples","Pear","Orange","Banana"];

//That’s because arrays are objects. So both shoppingCart and fruits are the references to the same array.

**3) 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**

**→**

**4) Map to objects**

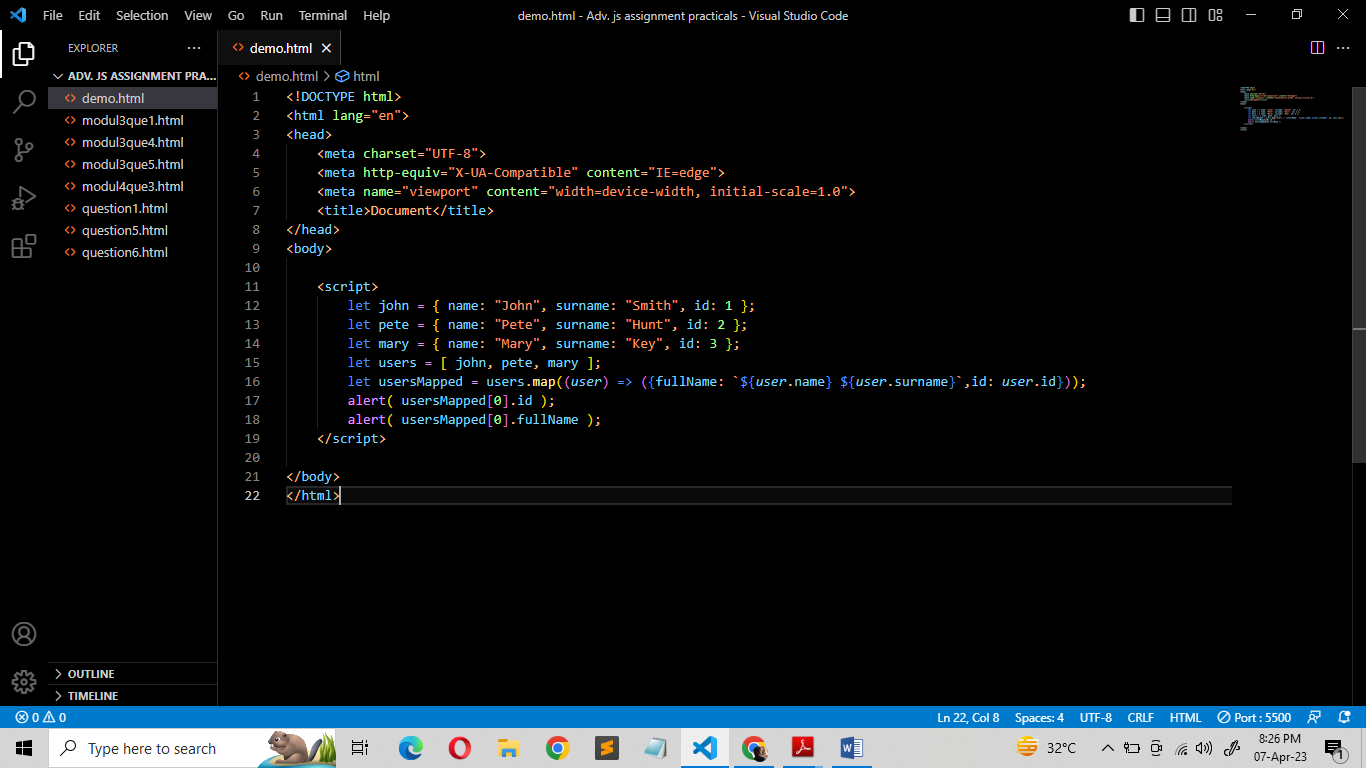
**let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [ john, pete, mary ]; let usersMapped = /\* ... your code ... \*/**

**/\* usersMapped = [ { fullName: "John Smith", id: 1 },**

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

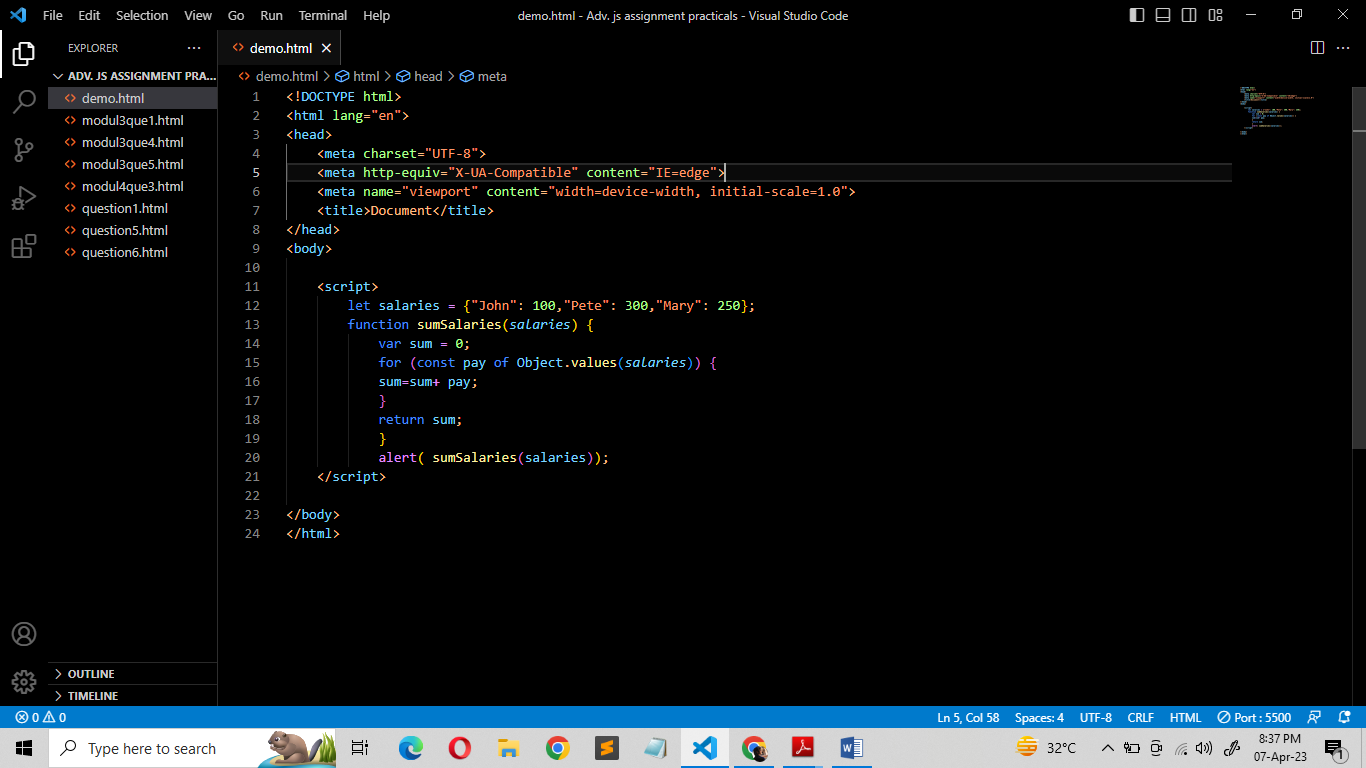
**{ fullName: "Mary Key", id: 3 } ] \*/**

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

**→**

**5) 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**

**→**

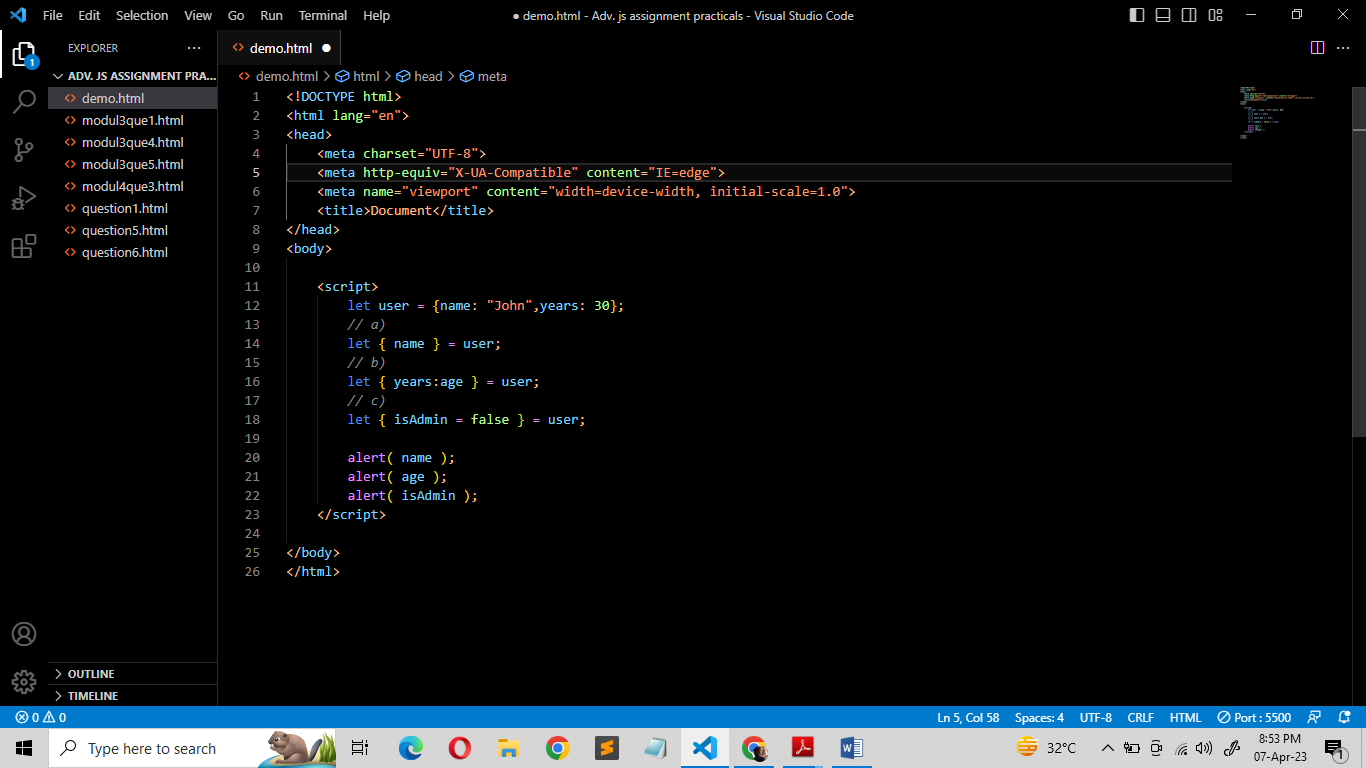
**6) Destructuring assignment We have an object: Write the Destructuring assignment that reads:**

**a) Name property into the variable name.**

**b) Year’s property into the variable age.**

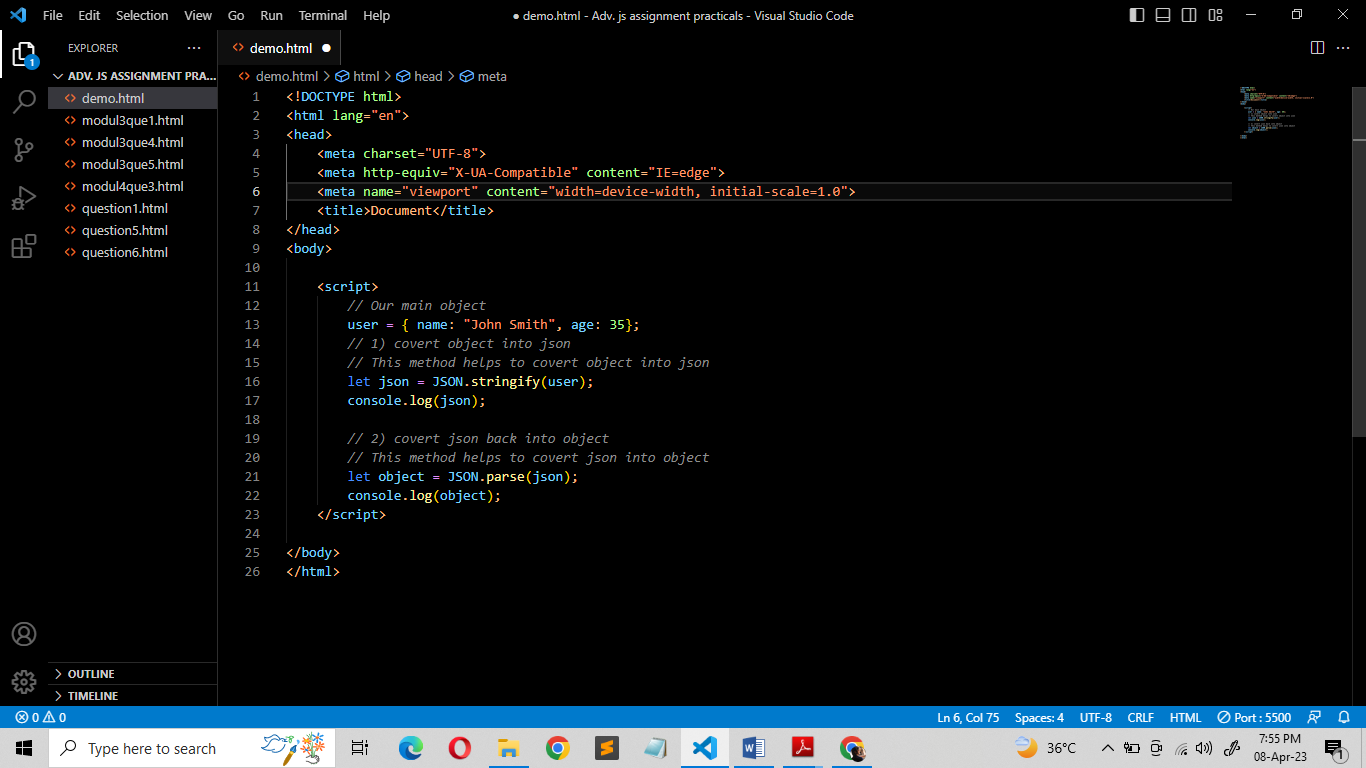
**c) isAdmin property into the variable isAdmin (false, if no such property)**

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

**→**

**7) 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)**

1. **Create a program to hide/show the password**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que1.html**

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

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que2.html**

1. **Create a responsive header using proper JavaScript**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que3.html**

1. **Create a form and validate using JavaScript**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que4%267.html**

1. **Create a modal box using css and Js with three buttons**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que5.html**

1. **Use external js library to show slider**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que6.html**

1. **Prevent the browser when i click the form submit button**

**→ https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul3que4%267.html**

**MODULE: 4 (New Request)**

**1) What is JSON?**

**→** JSON stands for JavaScript Object Notation. It is a lightweight format for storing and transporting data. It is often used when data is sent from a server to a web page. JSON is "self-describing" and easy to understand.

The JSON format is syntactically identical to the code for creating JavaScript objects. Because of this similarity, a JavaScript program can easily convert JSON data into native JavaScript objects.

JSON Syntax Rules:-

* Data is in name/value pairs
* Data is separated by commas
* Curly braces hold objects
* Square brackets hold arrays

**2) What is promises?**

**→** A Promise is a JavaScript object that links producing code and consuming code.

A JavaScript Promise object can be:

* Pending
* Fulfilled
* Rejected

The Promise object supports two properties: state and result.

While a Promise object is "**pending**" (working), the result is **undefined**.

When a Promise object is "**fulfilled**", the result is a **value**.

When a Promise object is "**rejected**", the result is an **error** object.

Syntax:-

myPromise.then(

function(value) { /\* code If successful \*/ },

function(error) { /\* code if some error \*/ }

);

**3) Write a program of promises and handle that promises also**

**→** [**https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que3.html**](https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que3.html)

**4) Use fetch method for calling an api** [**https://fakestoreapi.com/products**](https://fakestoreapi.com/products)

**→** [**https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que4%265.html**](https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que4%265.html)

**5) Display all the product from the api in your HTML page**

**→** [**https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que4%265.html**](https://github.com/PurvangS106/java-script/blob/main/Adv.%20js%20assignment%20practicals/modul4que4%265.html)