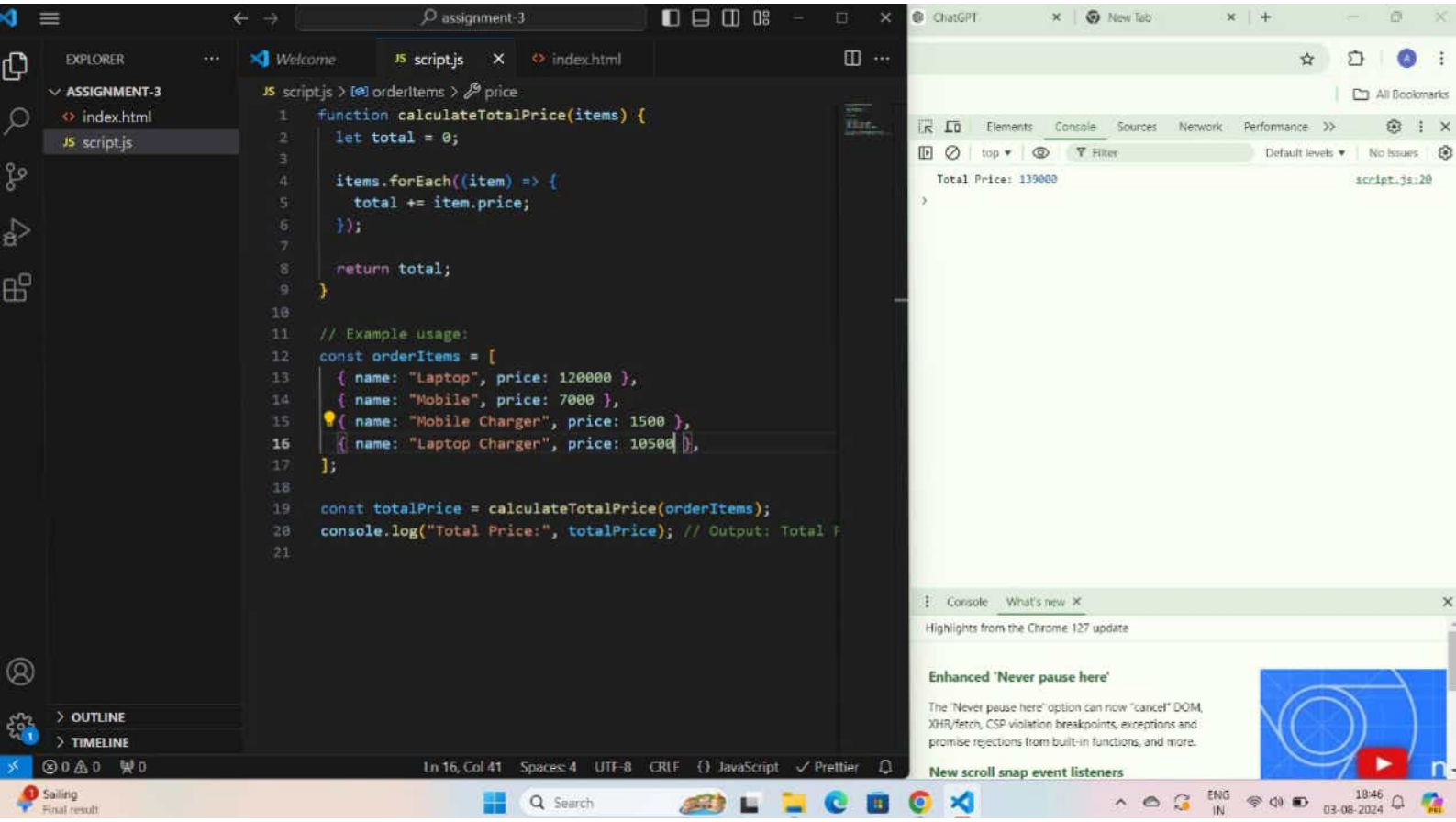
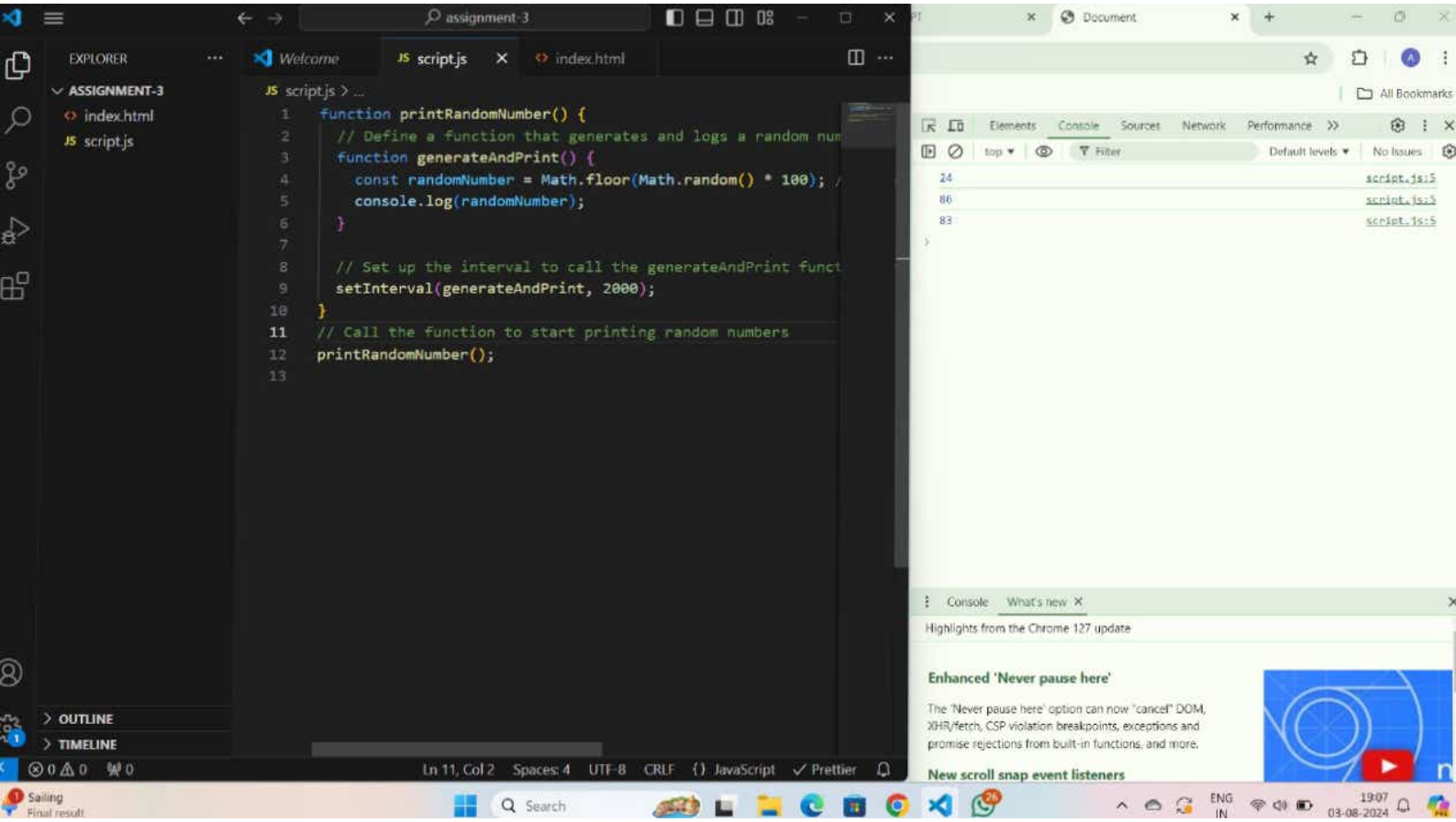
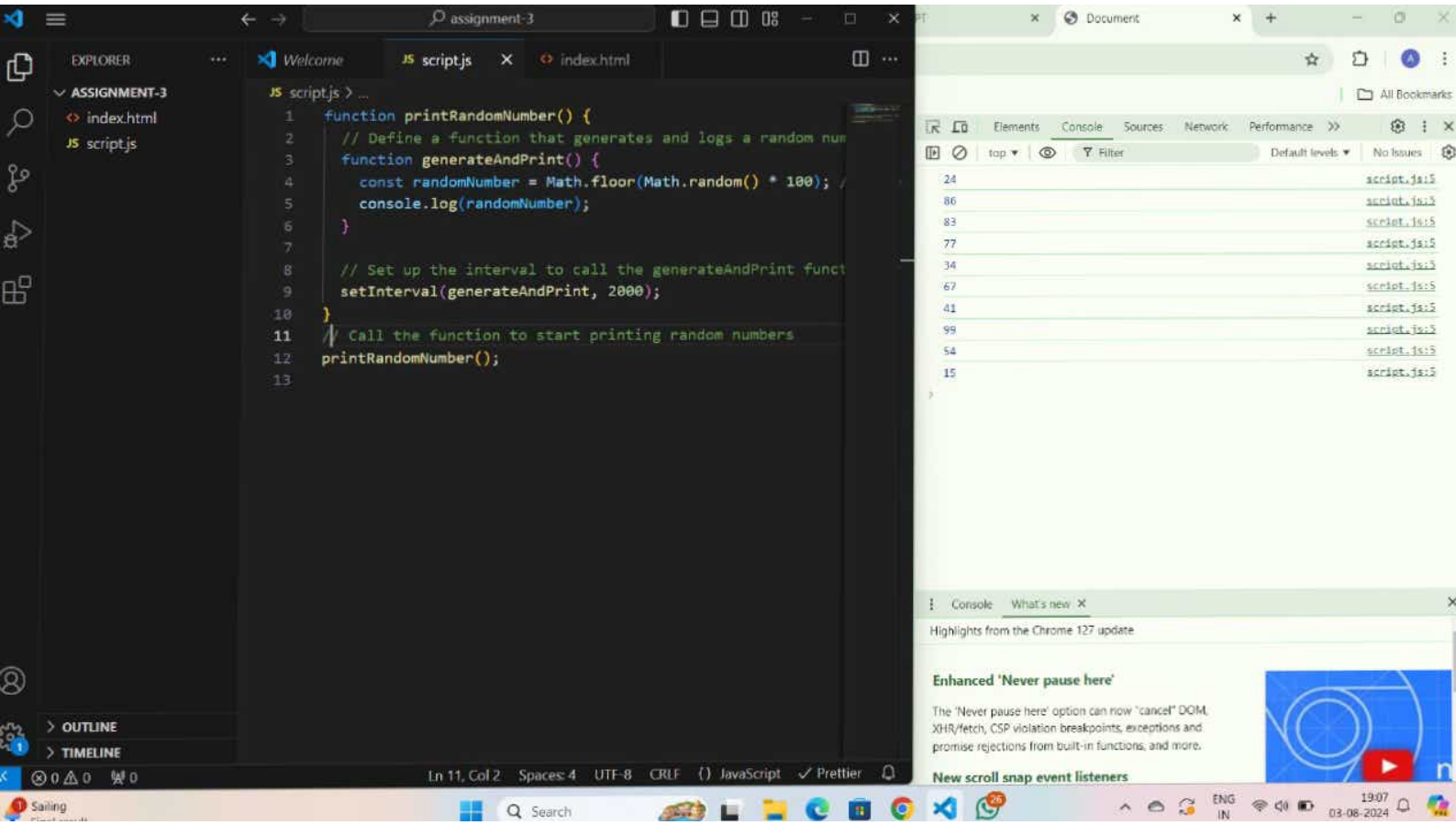
Question - 1

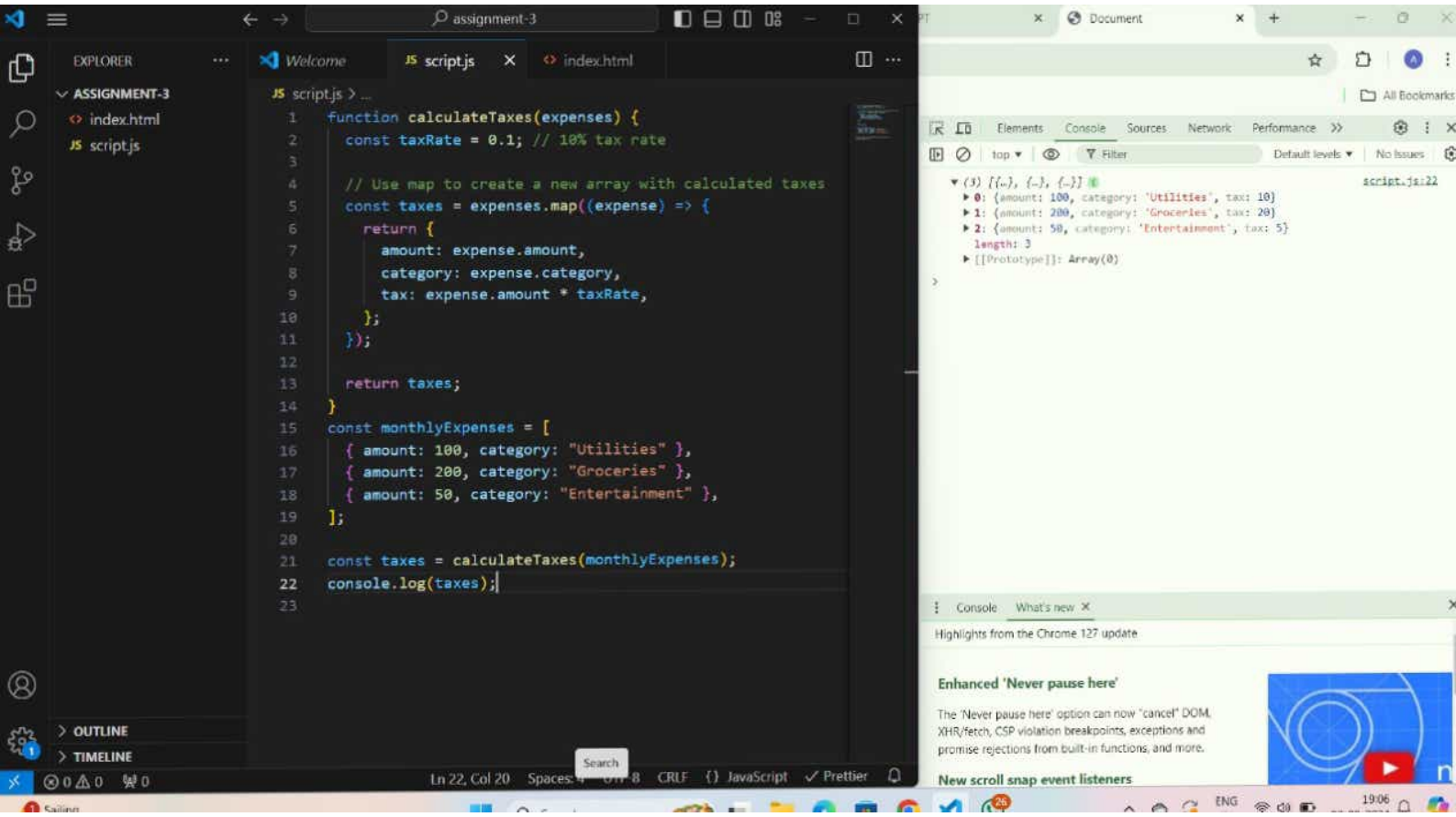

assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
function calculateTotalPrice(items) {
let total = 0;
items.forEach((item) => {
total += item.price;
});
return total;
}
// Example usage:
const orderItems = [
{ name: "Laptop", price: 120000 },
{ name: "Mobile", price: 7000 },
{ name: "Mobile Charger", price: 1500 },
{ name: "Laptop Charger", price: 10500 },
];
const totalPrice = calculateTotalPrice(orderItems);
console.log("Total Price:", totalPrice); // Output: Total
OUTLINE
TIMELINE
Ln 16, Col 41
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
ChatGPT
New Tab
Elements
Console
Sources
Network
Performance
Total Price: 139000
script.js:20
Console
What's new
Highlights from the Chrome 127 update
Enhanced 'Never pause here'
The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in functions, and more.
New scroll snap event listeners
Search
ENG
IN
18:46
03-08-2024

Question - 2

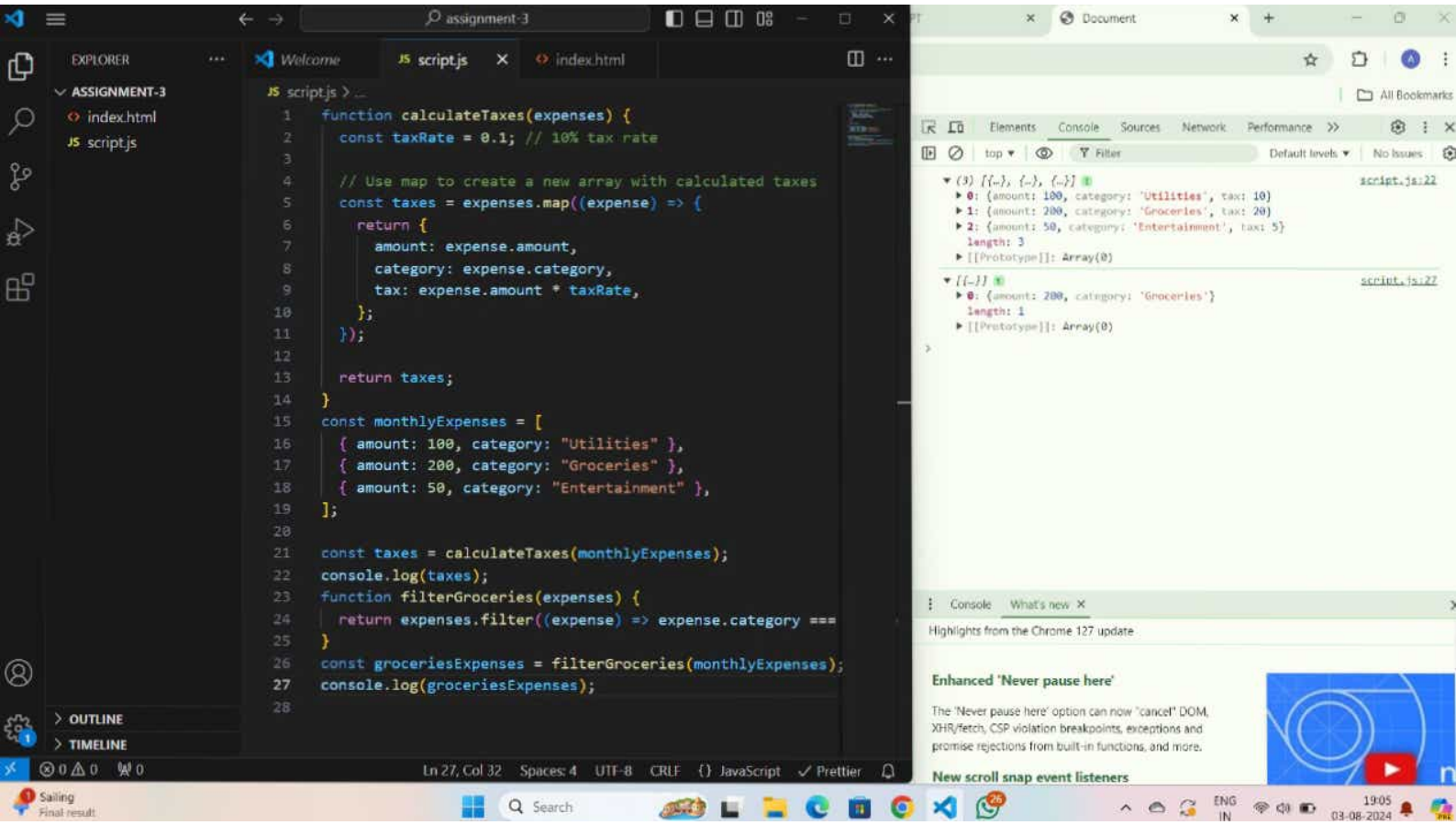

assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
function printRandomNumber() {
// Define a function that generates and logs a random num
function generateAndPrint() {
const randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);
}
// Set up the interval to call the generateAndPrint funct
setInterval(generateAndPrint, 2000);
}
// Call the function to start printing random numbers
printRandomNumber();
OUTLINE
TIMELINE
Ln 11, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
24
86
83
script.js:5
Console
What's new
Highlights from the Chrome 127 update
Enhanced 'Never pause here'
The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in functions, and more.
New scroll snap event listeners
Search
ENG
IN
19:07
03-08-2024


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
script.js > ...
function printRandomNumber() {
// Define a function that generates and logs a random num
function generateAndPrint() {
const randomNumber = Math.floor(Math.random() * 100);
console.log(randomNumber);
}
// Set up the interval to call the generateAndPrint funct
setInterval(generateAndPrint, 2000);
}
// Call the function to start printing random numbers
printRandomNumber();
OUTLINE
TIMELINE
Ln 11, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
All Bookmarks
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
24 script.js:5
86 script.js:5
83 script.js:5
77 script.js:5
34 script.js:5
67 script.js:5
41 script.js:5
99 script.js:5
54 script.js:5
15 script.js:5
Console
What's new
Highlights from the Chrome 127 update
Enhanced 'Never pause here'
The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in functions, and more.
New scroll snap event listeners
Sailing
Search
ENG
IN
19:07
03-08-2024

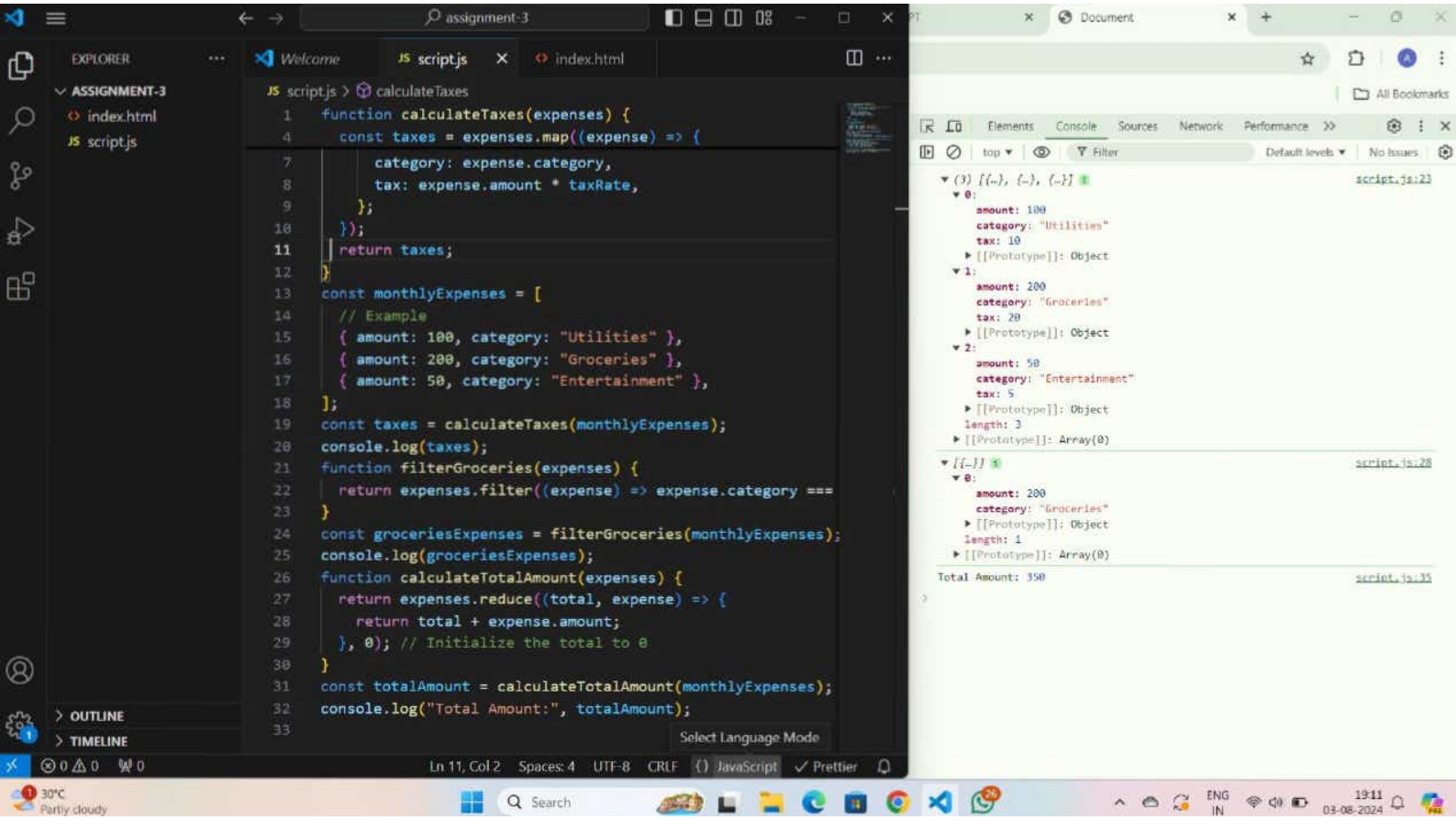
Question - 3


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
function calculateTaxes(expenses) {
const taxRate = 0.1; // 10% tax rate
// Use map to create a new array with calculated taxes
const taxes = expenses.map((expense) => {
return {
amount: expense.amount,
category: expense.category,
tax: expense.amount * taxRate,
};
});
return taxes;
}
const monthlyExpenses = [
{ amount: 100, category: "Utilities" },
{ amount: 200, category: "Groceries" },
{ amount: 50, category: "Entertainment" },
];
const taxes = calculateTaxes(monthlyExpenses);
console.log(taxes);
OUTLINE
TIMELINE
Ln 22, Col 20
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
script.js:22
0: {amount: 100, category: 'Utilities', tax: 10}
1: {amount: 200, category: 'Groceries', tax: 20}
2: {amount: 50, category: 'Entertainment', tax: 5}
length: 3
[[Prototype]]: Array(0)
Console
What's new
Highlights from the Chrome 127 update
Enhanced 'Never pause here'
The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in functions, and more.
New scroll snap event listeners

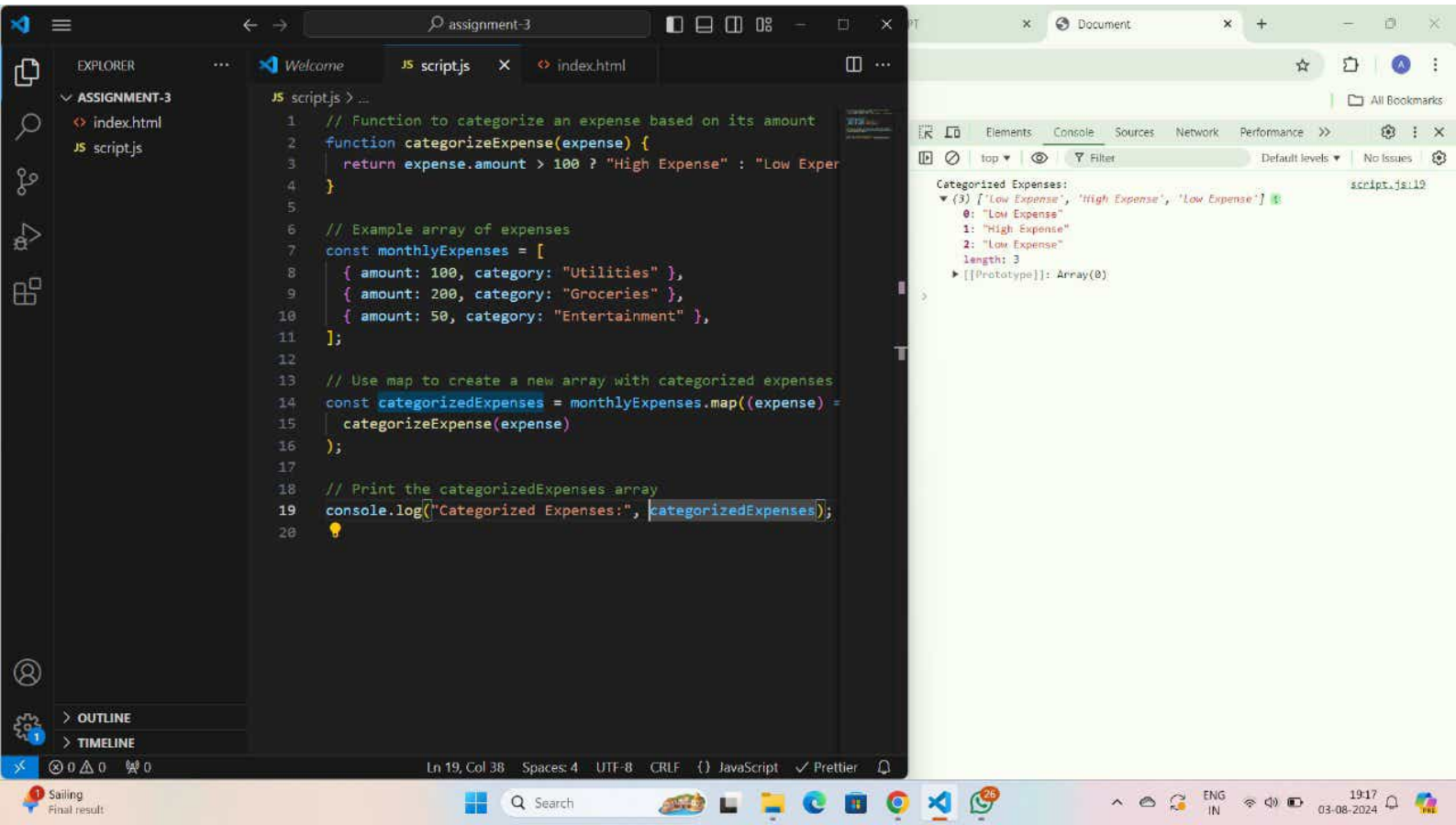
Question - 4


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
function calculateTaxes(expenses) {
const taxRate = 0.1; // 10% tax rate
// Use map to create a new array with calculated taxes
const taxes = expenses.map((expense) => {
return {
amount: expense.amount,
category: expense.category,
tax: expense.amount * taxRate,
};
});
return taxes;
}
const monthlyExpenses = [
{ amount: 100, category: "Utilities" },
{ amount: 200, category: "Groceries" },
{ amount: 50, category: "Entertainment" },
];
const taxes = calculateTaxes(monthlyExpenses);
console.log(taxes);
function filterGroceries(expenses) {
return expenses.filter((expense) => expense.category ===
}
const groceriesExpenses = filterGroceries(monthlyExpenses);
console.log(groceriesExpenses);
OUTLINE
TIMELINE
Ln 27, Col 32
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
Default levels
No Issues
script.js:22
script.js:27
Console
What's new
Highlights from the Chrome 127 update
Enhanced 'Never pause here'
The 'Never pause here' option can now "cancel" DOM, XHR/fetch, CSP violation breakpoints, exceptions and promise rejections from built-in functions, and more.
New scroll snap event listeners
Search
ENG
IN
19:05
03-08-2024

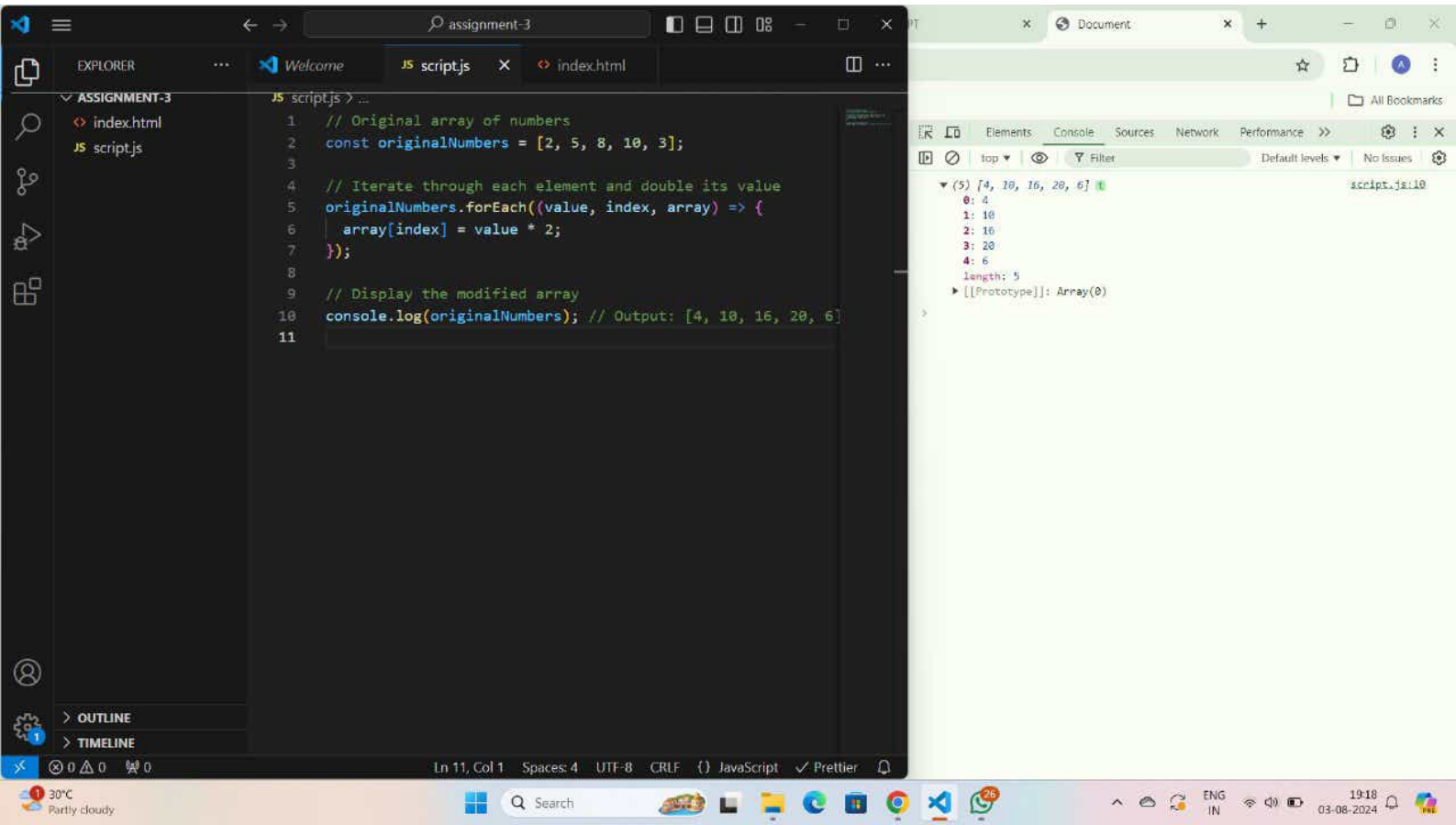
Question - 5


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
function calculateTaxes(expenses) {
const taxes = expenses.map((expense) => {
category: expense.category,
tax: expense.amount * taxRate,
});
return taxes;
}
const monthlyExpenses = [
// Example
{ amount: 100, category: "Utilities" },
{ amount: 200, category: "Groceries" },
{ amount: 50, category: "Entertainment" },
];
const taxes = calculateTaxes(monthlyExpenses);
console.log(taxes);
function filterGroceries(expenses) {
return expenses.filter((expense) => expense.category ===
}
const groceriesExpenses = filterGroceries(monthlyExpenses);
console.log(groceriesExpenses);
function calculateTotalAmount(expenses) {
return expenses.reduce((total, expense) => {
return total + expense.amount;
}, 0); // Initialize the total to 0
}
const totalAmount = calculateTotalAmount(monthlyExpenses);
console.log("Total Amount:", totalAmount);
OUTLINE
TIMELINE
Select Language Mode
Ln 11, Col 2
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
Filter
Default levels
No Issues
script.js:23
amount: 100
category: "Utilities"
tax: 10
amount: 200
category: "Groceries"
tax: 20
amount: 50
category: "Entertainment"
tax: 5
length: 3
length: 1
Total Amount: 350
script.js:28
script.js:35
30°C
Partly cloudy
Search
ENG
IN
19:11
03-08-2024

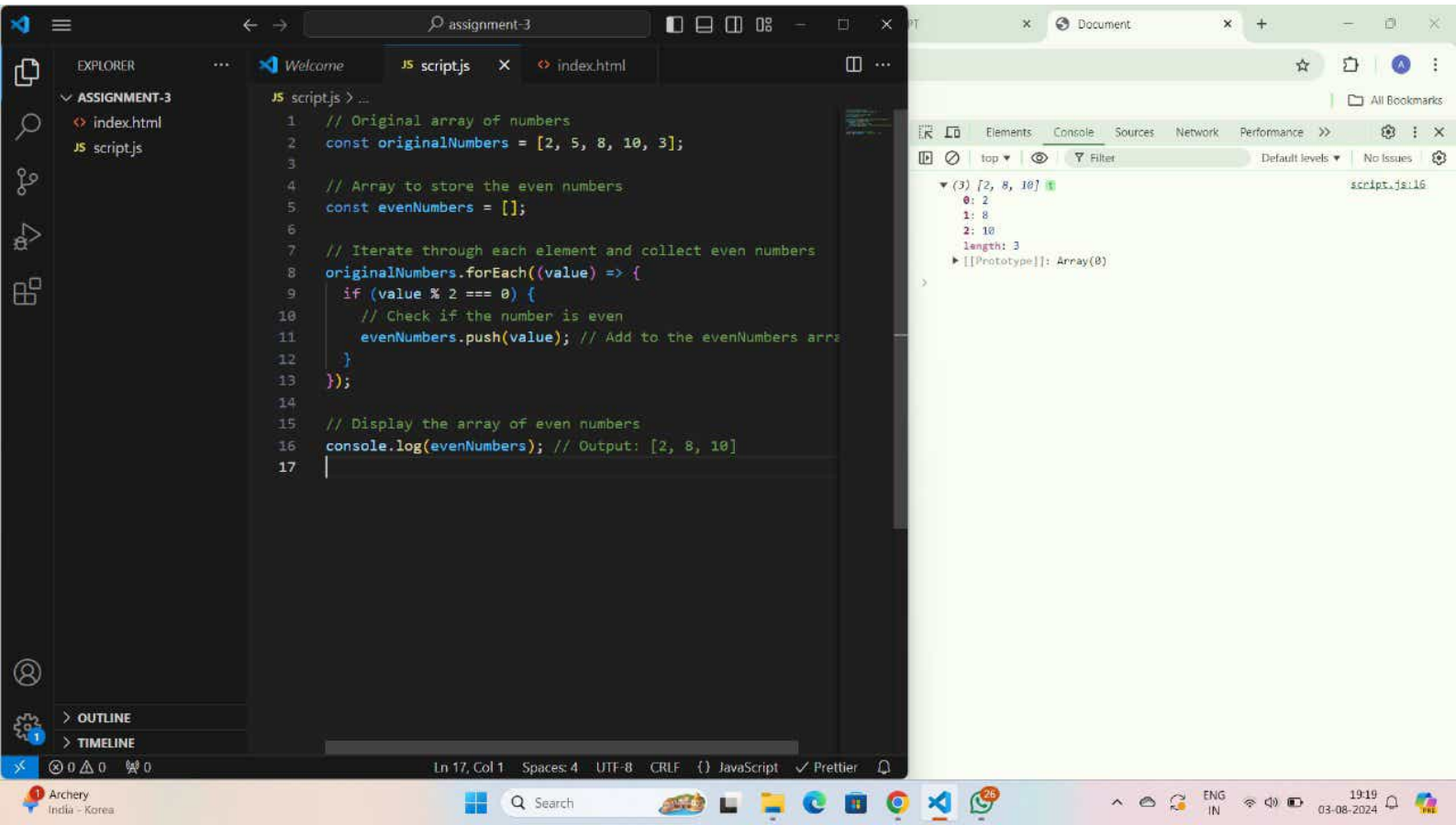
Question - 6


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
// Function to categorize an expense based on its amount
function categorizeExpense(expense) {
return expense.amount > 100 ? "High Expense" : "Low Exper
}
// Example array of expenses
const monthlyExpenses = [
{ amount: 100, category: "Utilities" },
{ amount: 200, category: "Groceries" },
{ amount: 50, category: "Entertainment" },
];
// Use map to create a new array with categorized expenses
const categorizedExpenses = monthlyExpenses.map((expense) =
categorizeExpense(expense)
);
// Print the categorizedExpenses array
console.log("Categorized Expenses:", categorizedExpenses);
OUTLINE
TIMELINE
Ln 19, Col 38
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
Categorized Expenses:
script.js:19
(3) ['Low Expense', 'High Expense', 'Low Expense']
0: "Low Expense"
1: "High Expense"
2: "Low Expense"
length: 3
[[Prototype]]: Array(0)
Search
ENG
IN
19:17
03-08-2024

Question - 7


assignment-3
EXPLORER
Welcome
script.js
index.html
ASSIGNMENT-3
index.html
script.js
// Original array of numbers
const originalNumbers = [2, 5, 8, 10, 3];
// Iterate through each element and double its value
originalNumbers.forEach((value, index, array) => {
array[index] = value * 2;
});
// Display the modified array
console.log(originalNumbers); // Output: [4, 10, 16, 20, 6]
OUTLINE
TIMELINE
Ln 11, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
(5) [4, 10, 16, 20, 6]
script.js:10
0: 4
1: 10
2: 16
3: 20
4: 6
length: 5
[[Prototype]]: Array(0)
30°C
Partly cloudy
Search
ENG
IN
19:18
03-08-2024

Question - 8


assignment-3
EXPLORER
ASSIGNMENT-3
index.html
script.js
Welcome
script.js
index.html
// Original array of numbers
const originalNumbers = [2, 5, 8, 10, 3];
// Array to store the even numbers
const evenNumbers = [];
// Iterate through each element and collect even numbers
originalNumbers.forEach((value) => {
if (value % 2 === 0) {
// Check if the number is even
evenNumbers.push(value); // Add to the evenNumbers arra
}
});
// Display the array of even numbers
console.log(evenNumbers); // Output: [2, 8, 10]
OUTLINE
TIMELINE
Ln 17, Col 1
Spaces: 4
UTF-8
CRLF
JavaScript
Prettier
Document
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
(3) [2, 8, 10]
script.js:16
0: 2
1: 8
2: 10
length: 3
[[Prototype]]: Array(0)
Archery
India - Korea
Search
ENG
IN
19:19
03-08-2024