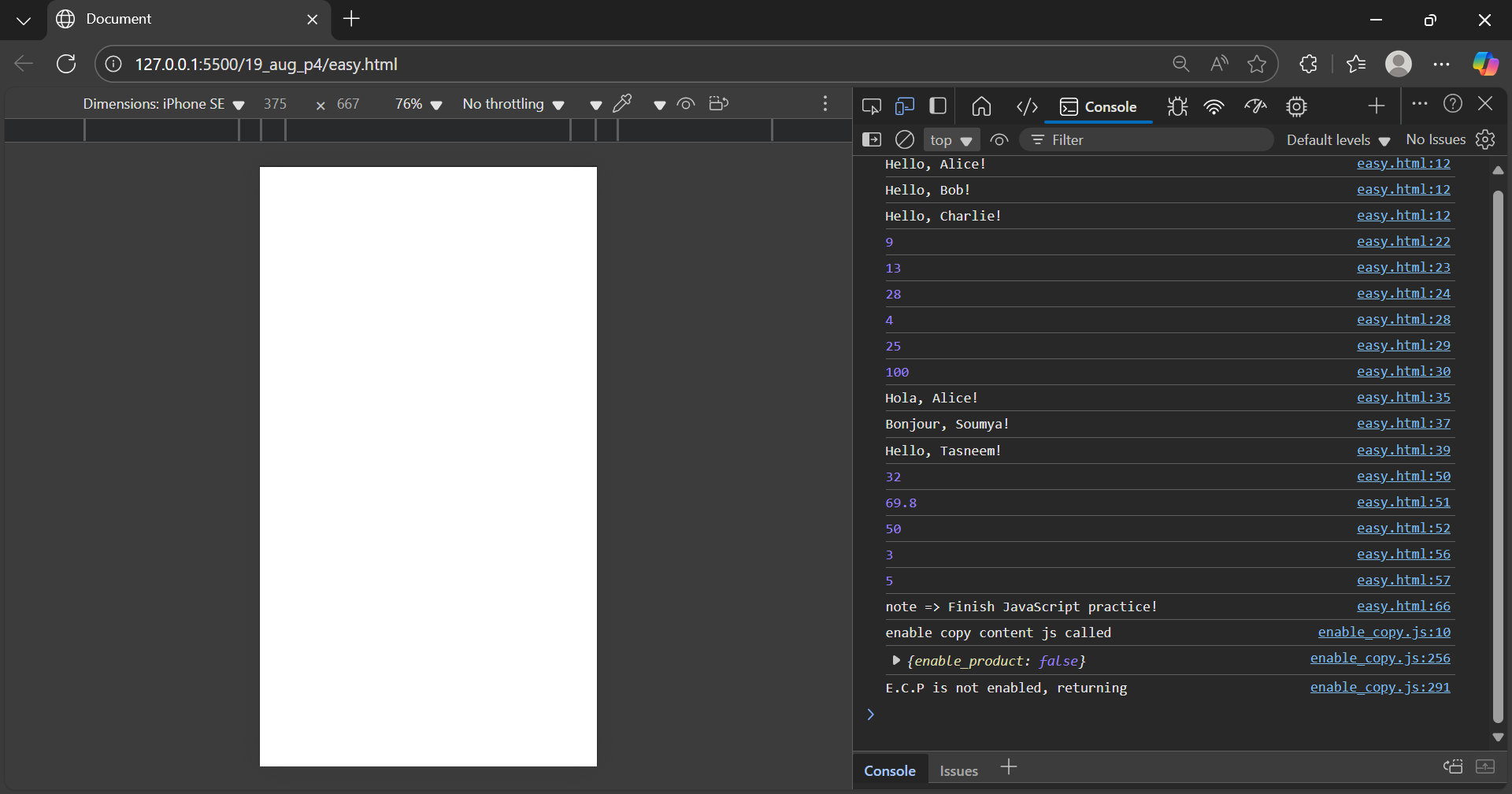
Name : Soumya Dixit

PRN : 23070521151

Batch : B2



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<script>

// 1. Function Declaration

function greet(name) {

console.log(`Hello, ${name}!`);

}

greet("Alice");

greet("Bob");

greet("Charlie");

// 2. Function Expression

const sum = function(a, b) {

return a + b;

};

console.log(sum(2, 7));

console.log(sum(10, 3));

console.log(sum(-5, 33));

// 3. Arrow Function

const square = (n) => n \* n;

console.log(square(2));

console.log(square(5));

console.log(square(10));

// 4. Function Declaration with Default Parameter

function greetLang(name, language = "English") {

if (language === "Spanish") {

console.log(`Hola, ${name}!`);

} else if (language === "French") {

console.log(`Bonjour, ${name}!`);

} else {

console.log(`Hello, ${name}!`);

}

}

greetLang("Alice", "Spanish");

greetLang("Soumya", "French");

greetLang("Tasneem");

// 5. Function Expression – Celsius to Fahrenheit

const toFahrenheit = function(celsius) {

return (celsius \* 9/5) + 32;

};

console.log(toFahrenheit(0));

console.log(toFahrenheit(21));

console.log(toFahrenheit(10));

// 6. Arrow Function – Word Count

const wordCount = (str) => str.split(" ").length;

console.log(wordCount("I love JavaScript"));

console.log(wordCount("This is a test sentence"));

// 7. Simple Closure – Personal Note

function createNote(note) {

return function() {

return `note => ${note}`;

};

}

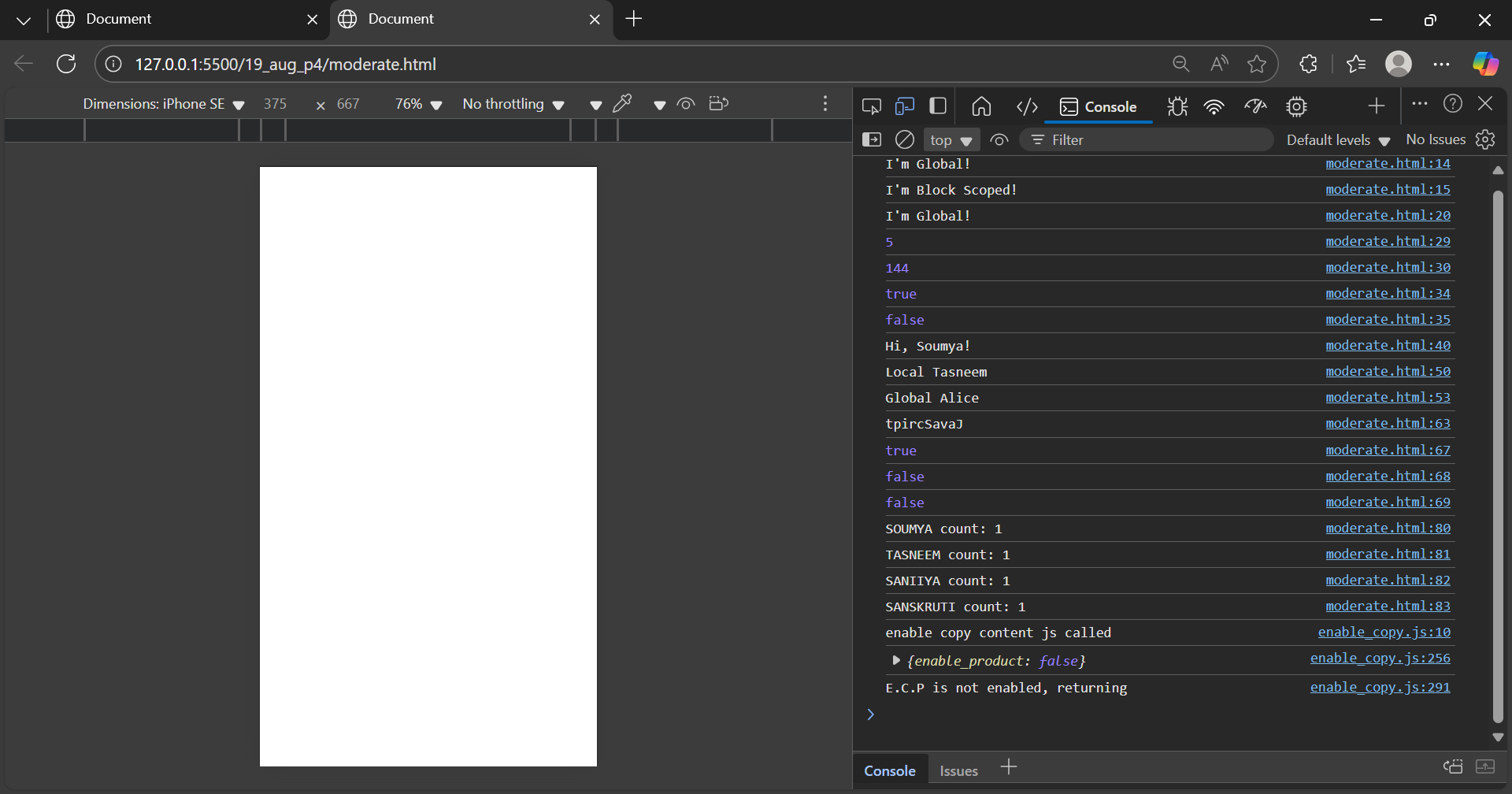
const myNote = createNote("Finish JavaScript practice!");

console.log(myNote());

</script>

</body>

</html>



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<script>

// 1. Scope Practice

let globalVar = "I'm Global!";

{

let blockVar = "I'm Block Scoped!";

console.log(globalVar);

console.log(blockVar);

}

function scopeTest() {

console.log(globalVar);

}

scopeTest();

// 2. Function Expression + Recursion (Fibonacci)

const fibonacci = function fib(n) {

if (n <= 1) return n;

return fib(n - 1) + fib(n - 2);

};

console.log(fibonacci(5));

console.log(fibonacci(12));

// 3. Arrow Function String Check

const endsWithExclamation = (str) => str.endsWith("!");

console.log(endsWithExclamation("Hello!"));

console.log(endsWithExclamation("Hello"));

// 4. Simple Closure – Greeter

function createGreeter(message) {

return function(name) {

console.log(`${message}, ${name}!`);

};

}

const greetHi = createGreeter("Hi");

greetHi("Soumya");

// 5. Scope Example – Shadowing

let name = "Global Alice";

function testShadow() {

let name = "Local Tasneem";

console.log(name);

}

testShadow();

console.log(name);

// 6. Function Expression – Reverse String

const reverseStr = function(str) {

let reversed = "";

for (let i = str.length - 1; i >= 0; i--) {

reversed += str[i];

}

return reversed;

};

console.log(reverseStr("JavaScript")); // tpircSavaJ

// 7. Arrow Function – Multiple Checks

const checkNumber = (n) => n % 2 === 0 && n > 0;

console.log(checkNumber(4));

console.log(checkNumber(-2));

console.log(checkNumber(3));

// 8. Closure – Personalized Counter

function nameCounter() {

let counts = {};

return function(person) {

counts[person] = (counts[person] || 0) + 1;

return `${person} count: ${counts[person]}`;

};

}

const counter = nameCounter();

console.log(counter("SOUMYA"));

console.log(counter("TASNEEM"));

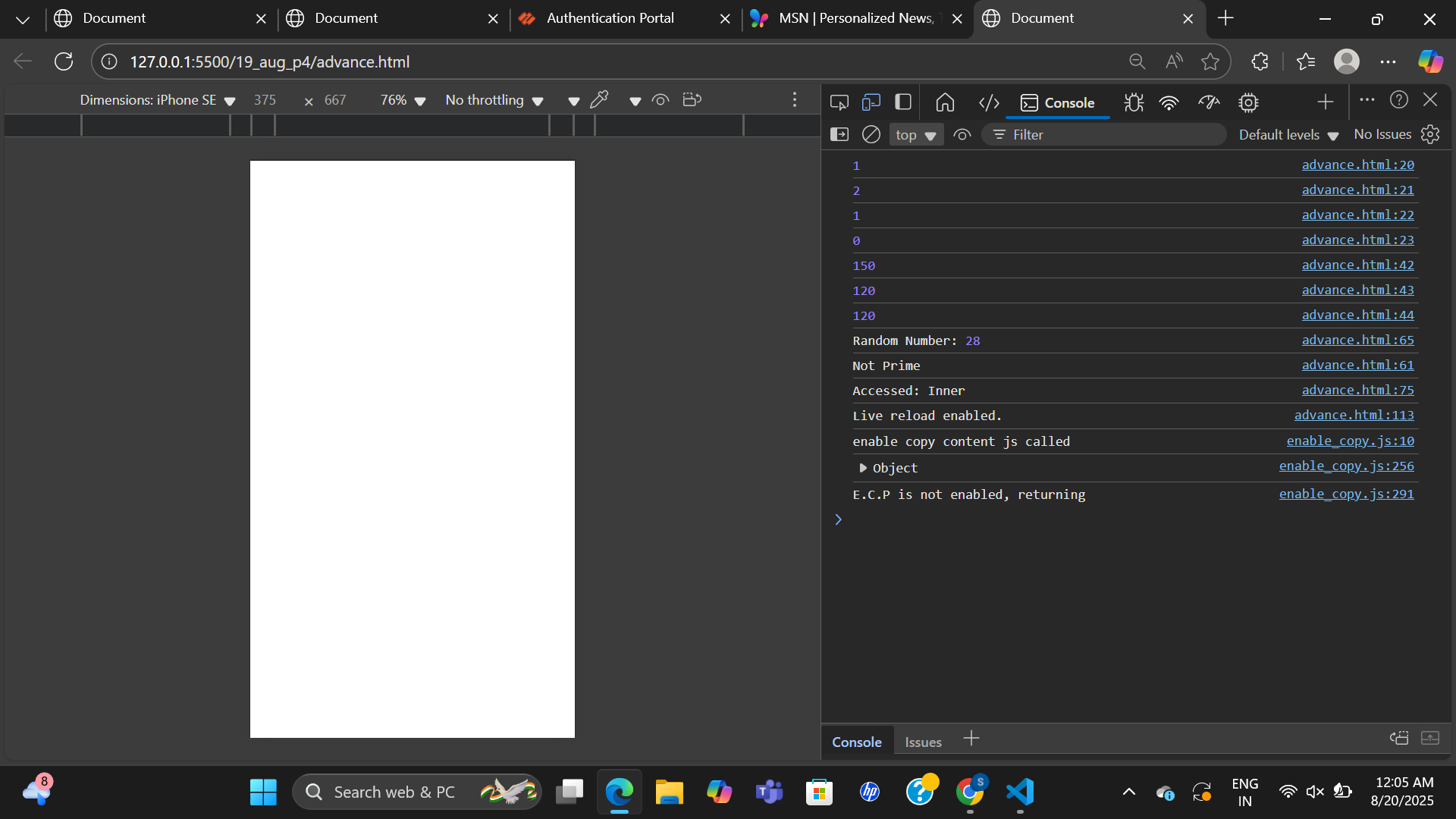
console.log(counter("SANIIYA"));

console.log(counter("SANSKRUTI"));

</script>

</body>

</html>

  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

</head>

<body>

<script>

// 1. Closure for Counter

function createCounter() {

let count = 0;

return {

increment: () => ++count,

decrement: () => --count,

reset: () => { count = 0; return count; }

};

}

const counter1 = createCounter();

console.log(counter1.increment());

console.log(counter1.increment());

console.log(counter1.decrement());

console.log(counter1.reset());

// 2. Private Data with Closure

function createBankAccount(initialBalance = 0) {

let balance = initialBalance;

return {

deposit: (amount) => { balance += amount; return balance; },

withdraw: (amount) => {

if (amount <= balance) {

balance -= amount;

return balance;

} else {

return "Insufficient funds!";

}

},

checkBalance: () => balance

};

}

const account = createBankAccount(100);

console.log(account.deposit(50));

console.log(account.withdraw(30));

console.log(account.checkBalance());

// 3. Mixed Function Types

// Function Declaration: Random Number

function getRandomNumber() {

return Math.floor(Math.random() \* 50) + 1;

}

// Function Expression: Check Prime

const isPrime = function(num) {

if (num < 2) return false;

for (let i = 2; i <= Math.sqrt(num); i++) {

if (num % i === 0) return false;

}

return true;

};

// Arrow Function: Print Result

const printPrime = (num) => console.log(isPrime(num) ? "Prime" : "Not Prime");

// Run sequence

let randomNum = getRandomNumber();

console.log("Random Number:", randomNum);

printPrime(randomNum);

// 4. Scope Chain Debugging

function outer() {

let name = "Outer";

function middle() {

let name = "Middle";

function inner() {

let name = "Inner";

console.log("Accessed:", name); // "Inner" (closest scope)

}

inner();

}

middle();

}

outer();

</script>

</body>

</html>