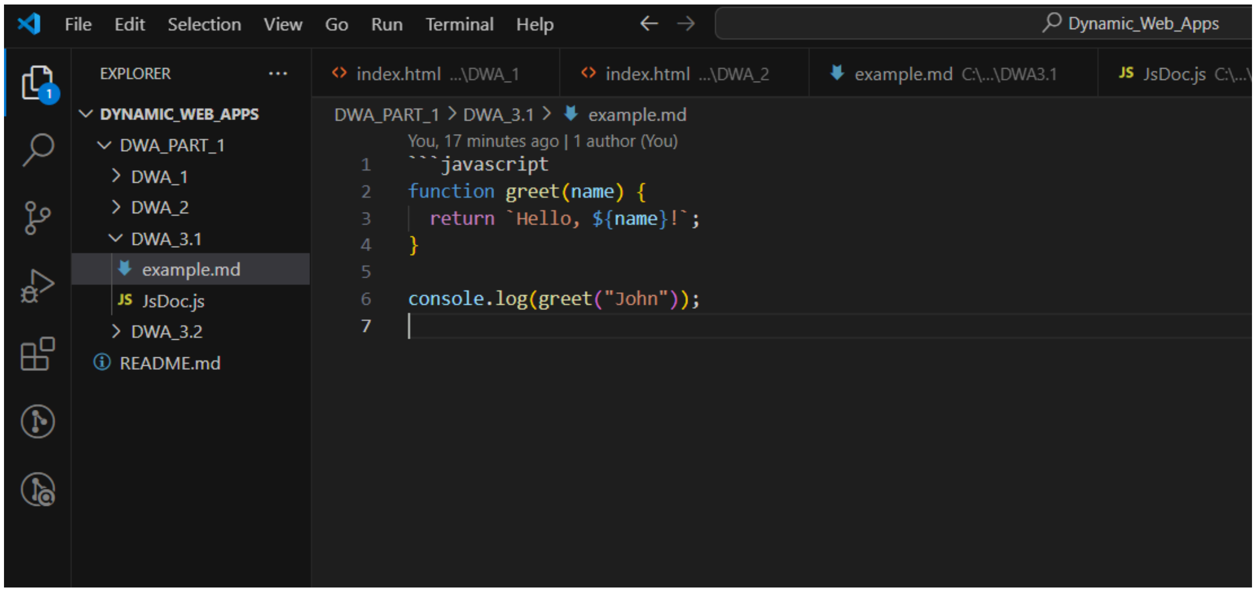
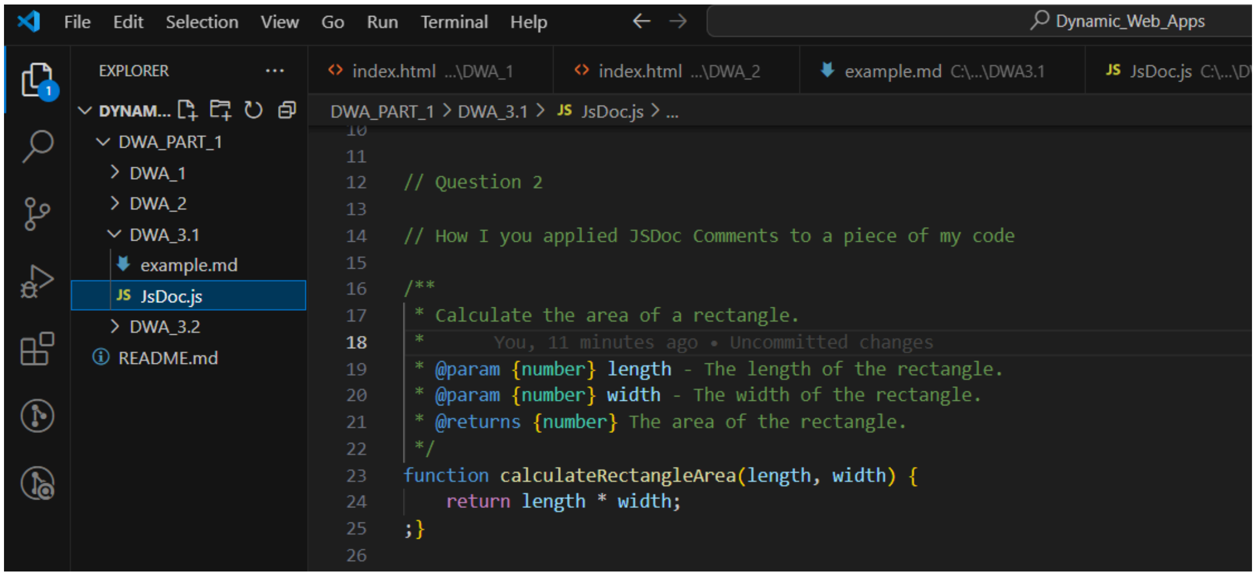
DWA_03.4 Knowledge Check_DWA3.1

1. Please show how you applied a Markdown File to a piece of your code.



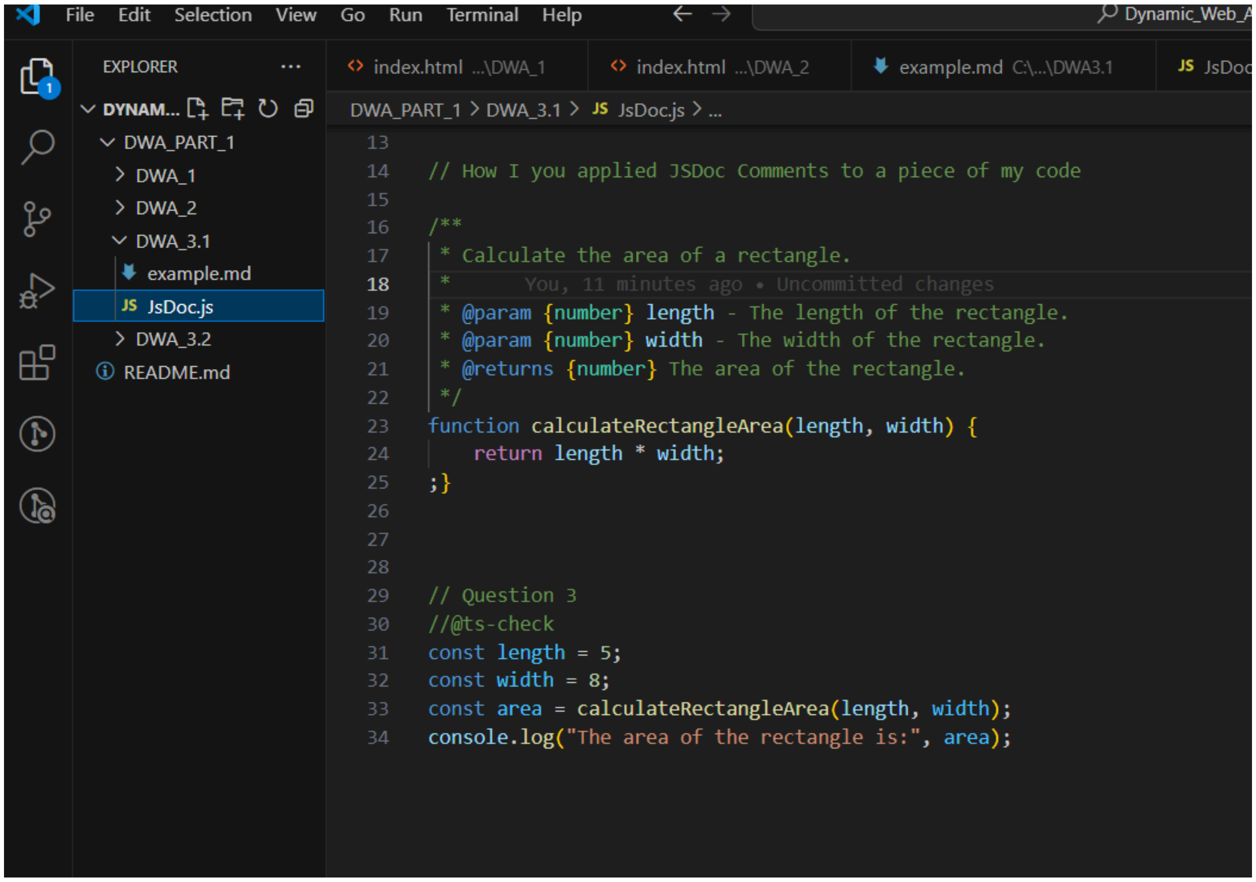
~~~
```javascript
function greet(name) {
  return `Hello, ${name}!`;
}

console.log(greet("John"));

~~~

2. Please show how you applied JSDoc Comments to a piece of your code.



```

// Question 2

// How I you applied JSDoc Comments to a piece of my code

/**
 * Calculate the area of a rectangle.
 *       You, 11 minutes ago • Uncommitted changes
 * @param {number} length - The length of the rectangle.
 * @param {number} width - The width of the rectangle.
 * @returns {number} The area of the rectangle.
 */
function calculateRectangleArea(length, width) {
    return length * width;
;}

```

3. Please show how you applied the @ts-check annotation to a piece of your code.



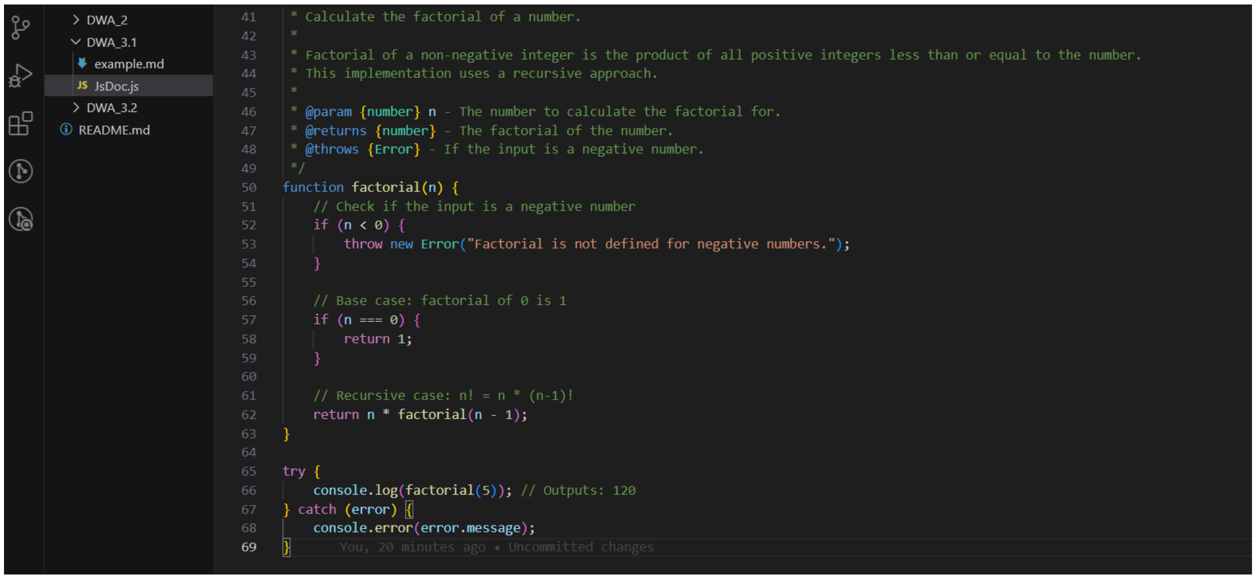
```

// How I you applied JSDoc Comments to a piece of my code

/**
 * Calculate the area of a rectangle.
 *       You, 11 minutes ago • Uncommitted changes
 * @param {number} length - The length of the rectangle.
 * @param {number} width - The width of the rectangle.
 * @returns {number} The area of the rectangle.
 */
function calculateRectangleArea(length, width) {
    return length * width;
;}



// Question 3
//@ts-check
const length = 5;
const width = 8;
const area = calculateRectangleArea(length, width);
console.log("The area of the rectangle is:", area);
```

---

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.

I will show In line Comments:

They explain specific parts of the factorial function, such as the base case and recursive case. And they also help clarify the logic and purpose of the code.


DWA_2
DWA_3.1
example.md
JsDoc.js
DWA_3.2
README.md

```
 * Calculate the factorial of a number.
 *
 * Factorial of a non-negative integer is the product of all positive integers less than or equal to the number.
 * This implementation uses a recursive approach.
 *
 * @param {number} n - The number to calculate the factorial for.
 * @returns {number} - The factorial of the number.
 * @throws {Error} - If the input is a negative number.
 */
function factorial(n) {
    // Check if the input is a negative number
    if (n < 0) {
        throw new Error("Factorial is not defined for negative numbers.");
    }

    // Base case: factorial of 0 is 1
    if (n === 0) {
        return 1;
    }

    // Recursive case: n! = n * (n-1)!
    return n * factorial(n - 1);
}

try {
    console.log(factorial(5)); // Outputs: 120
} catch (error) {
    console.error(error.message);
}      You, 20 minutes ago • Uncommitted changes
```