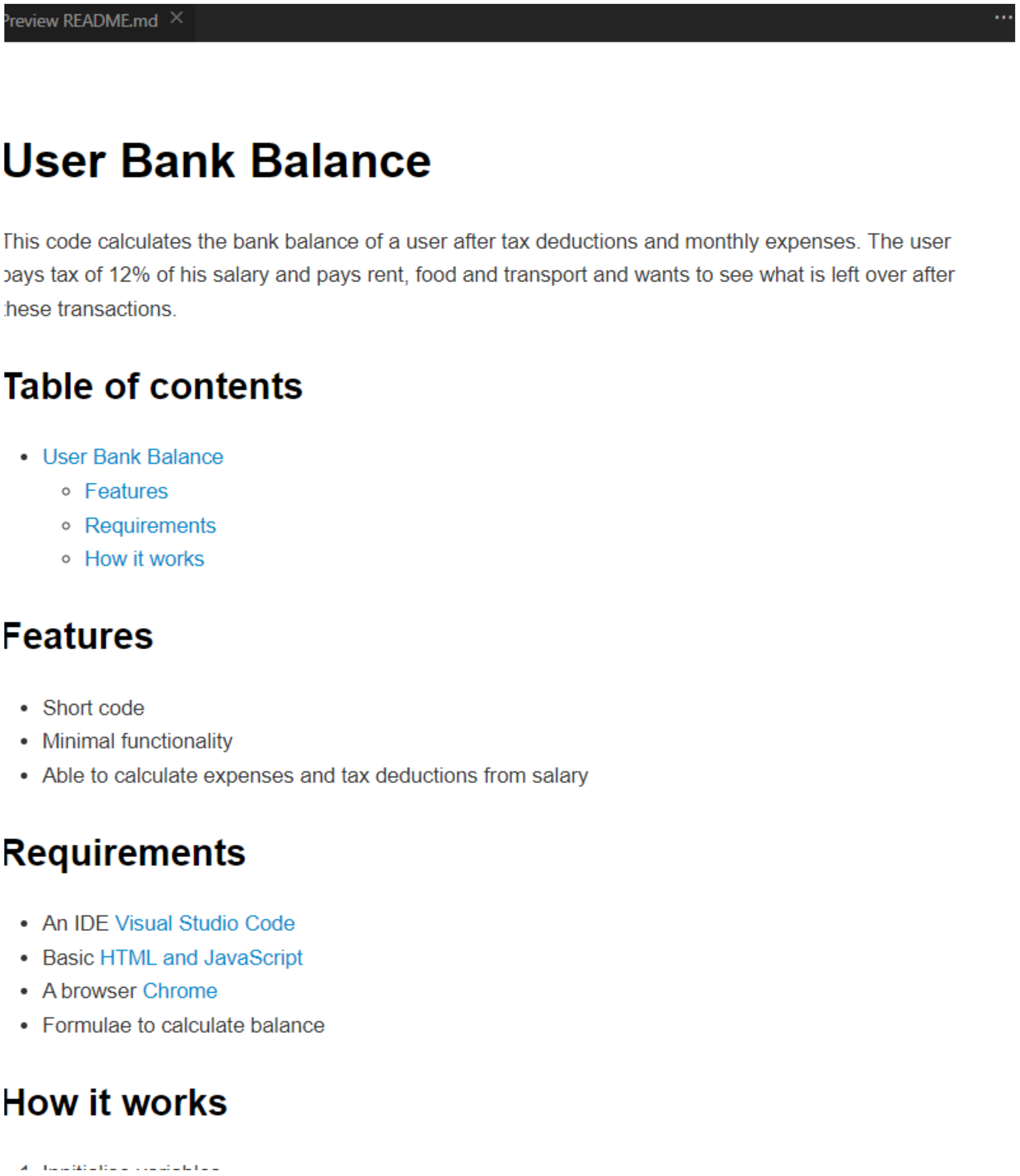
DWA_03.4 Knowledge Check_DWA3.1

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



User Bank Balance

This code calculates the bank balance of a user after tax deductions and monthly expenses. The user pays tax of 12% of his salary and pays rent, food and transport and wants to see what is left over after these transactions.

Table of contents

- User Bank Balance
  - Features
  - Requirements
  - How it works

Features

- Short code
- Minimal functionality
- Able to calculate expenses and tax deductions from salary

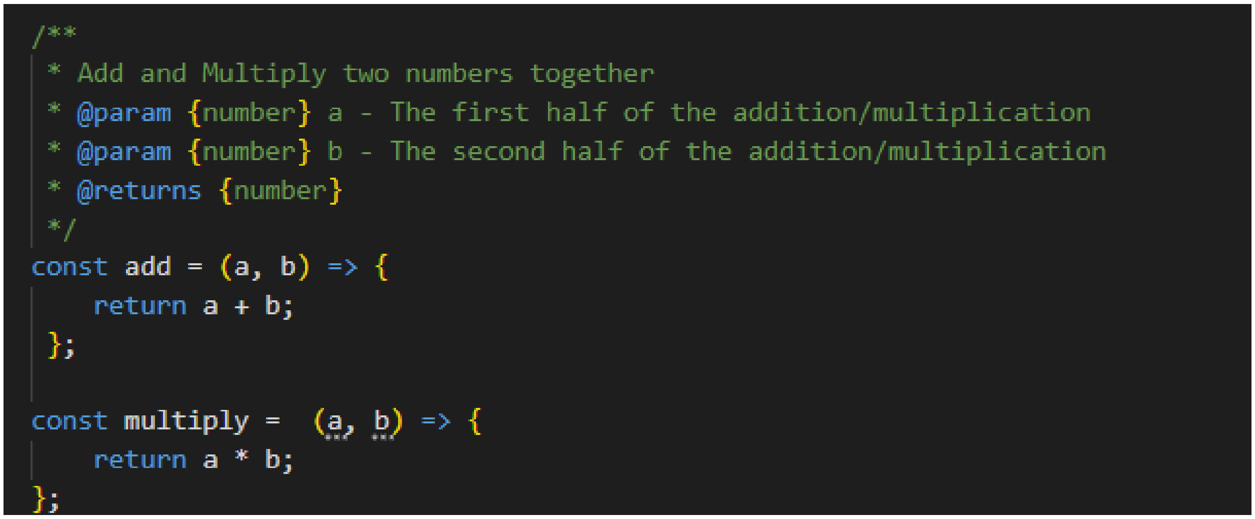
Requirements

- An IDE Visual Studio Code
- Basic HTML and JavaScript
- A browser Chrome
- Formulae to calculate balance

How it works

1. Innitialise variables

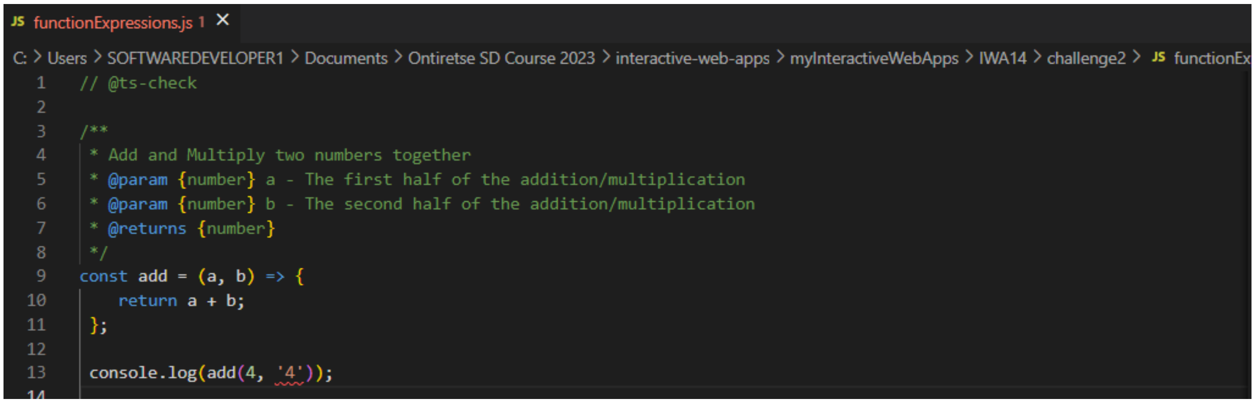
2. Please show how you applied JSDoc Comments to a piece of your code.



```
/**
 * Add and Multiply two numbers together
 * @param {number} a - The first half of the addition/multiplication
 * @param {number} b - The second half of the addition/multiplication
 * @returns {number}
 */
const add = (a, b) => {
    return a + b;
};

const multiply =  (a, b) => {
    return a * b;
};
```

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



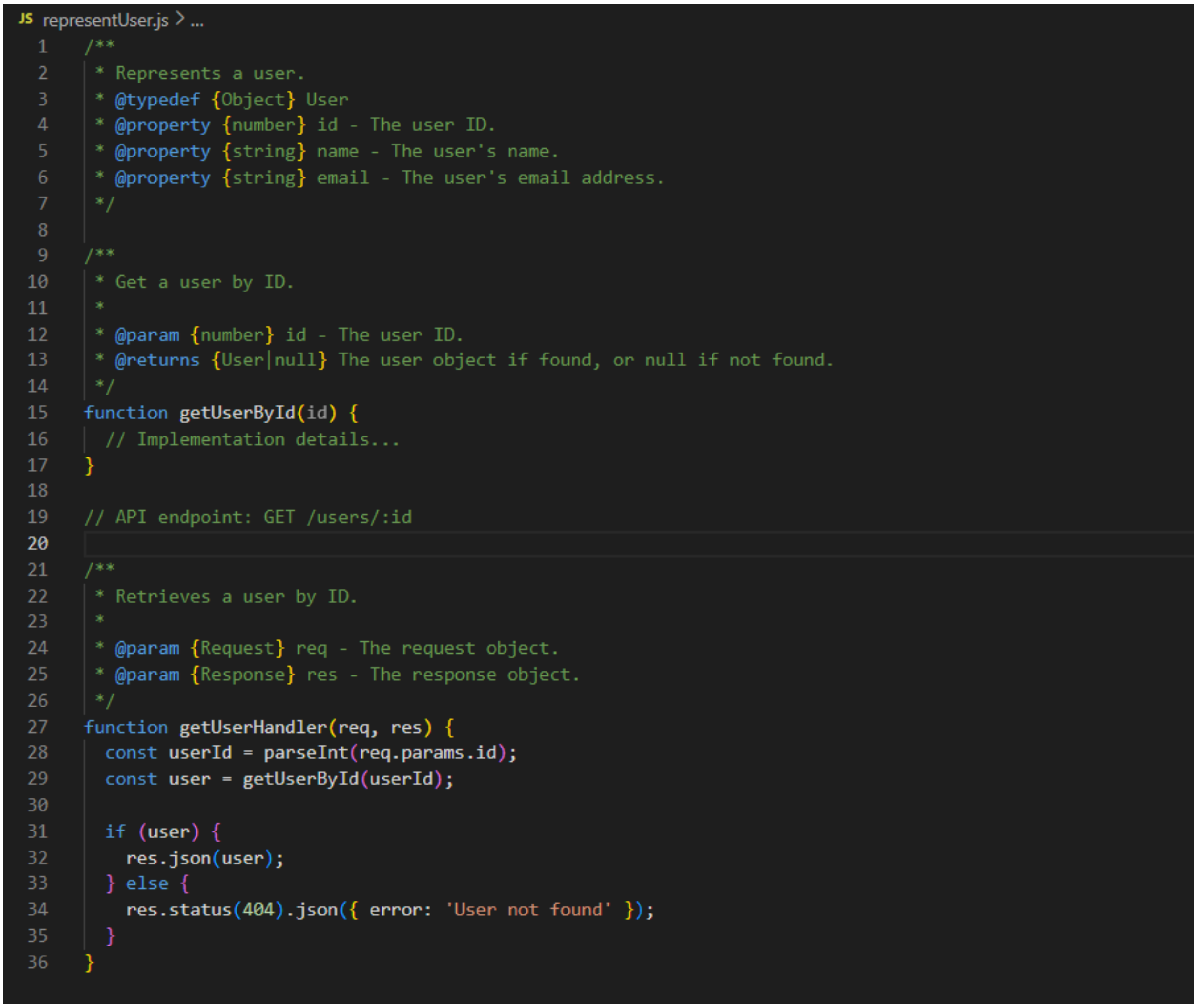
JS functionExpressions.js 1 ×
C: > Users > SOFTWAREDEVELOPER1 > Documents > Ontiretse SD Course 2023 > interactive-web-apps > myInteractiveWebApps > IWA14 > challenge2 > JS functionEx

```
// @ts-check

/**
 * Add and Multiply two numbers together
 * @param {number} a - The first half of the addition/multiplication
 * @param {number} b - The second half of the addition/multiplication
 * @returns {number}
 */
const add = (a, b) => {
    return a + b;
};

console.log(add(4, '4'));
```

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



```javascript
/**
 * Represents a user.
 * @typedef {Object} User
 * @property {number} id - The user ID.
 * @property {string} name - The user's name.
 * @property {string} email - The user's email address.
 */

/**
 * Get a user by ID.
 *
 * @param {number} id - The user ID.
 * @returns {User|null} The user object if found, or null if not found.
 */
function getUserById(id) {
  // Implementation details...
}

// API endpoint: GET /users/:id

/**
 * Retrieves a user by ID.
 *
 * @param {Request} req - The request object.
 * @param {Response} res - The response object.
 */
function getUserHandler(req, res) {
  const userId = parseInt(req.params.id);
  const user = getUserById(userId);

  if (user) {
    res.json(user);
  } else {
    res.status(404).json({ error: 'User not found' });
  }
}
```