DWA_03.4 Knowledge Check_DWA3.1

---

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

```
/**
 * Calculates the area of a circle given its radius.
 * @param {number} radius - The radius of the circle.
 * @returns {number} The area of the circle.
 */
function calculateArea(radius) {
    return Math.PI * radius * radius;
}
```

---

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

```
/**
 * Represents a person.
 * @constructor
 * @param {string} name - The name of the person.
 * @param {number} age - The age of the person.
 */
function Person(name, age) {
    /**
     * The name of the person.
     * @type {string}
     */
    this.name = name;
```

```
    /**
     * The age of the person.
     * @type {number}
     */
    this.age = age;
}

/**
```

---

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

```
// @ts-check
/**
 * Adds two numbers.
 * @param {number} a - The first number.
 * @param {number} b - The second number.
 * @returns {number} The sum of the two numbers.
 */
function add(a, b) {
    return a + b;
}
```

---

4. As a BONUS, please show how you applied any other concept covered in the 'Documentation' module.
As my bonus i have learned about Adding "displaying" and "filters" to state

```
import React, { Component } from 'react';

class ItemList extends Component {
  constructor(props) {
    super(props);
    this.state = {
      items: [
        'Apple',
        'Banana',
        'Orange',
        'Mango',
        'Pineapple'
      ],
      displaying: [], // Holds items to be displayed
      filters: {
        searchQuery: '' // Holds the search query
      }
    };
  }

  componentDidMount() {
    // Initialize displaying state with all items initially
    this.setState({ displaying: this.state.items });
  }

  handleSearchChange = (e) => {
    const searchQuery = e.target.value.toLowerCase();
    const displaying = this.state.items.filter(item =>
      item.toLowerCase().includes(searchQuery)
    );
    this.setState({
      filters: {
        searchQuery: searchQuery
      },
```

```
      displaying: displaying
    });
  }

  render() {
    return (
      <div>
        <input
          type="text"
          value={this.state.filters.searchQuery}
          onChange={this.handleSearchChange}
          placeholder="Search items..."
        />
        <ul>
          {this.state.displaying.map((item, index) => (
            <li key={index}>{item}</li>
          ))}
        </ul>
      </div>
    );
  }
}

export default ItemList;
```
