

DWA_03.4 Knowledge Check_DWA3.1

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

- I used a README.md file to describe my book connect app. As shown below I have used the # to indicate a heading, the _ _ (underscore) to indicate italics, the * to indicate an unordered list, the ** ** to indicate that the word is in bold, and the 1. To indicate an ordered list.

Author:

Sumaiya Mohamed

Title:

Sumaiya-Mohamed/SUMMOH081_FTC2301_GROUPB_SumaiyaMohamed_ITW19
(Book Connect App)

Technologies Used:

* HTML

* CSS

* JavaScript

Description:

Book Connect is a versatile book app that makes finding new books easier for new readers or bookworms. With its intuitive interface and powerful features, Book Connect helps you explore, filter, and access comprehensive information about a vast collection of books.

Key Features:

1. **Smart Book Filtering** : Book Connect allows you to filter books effortlessly by title, author, or genre. Whether you're searching for a specific book or exploring works by your favorite author, the app quickly narrows down the results to match your criteria.

1. **Theme Customization** : Personalize your reading environment with Book Connect's theme options. Seamlessly switch between light and dark modes to suit your preferences and create a comfortable reading experience.

1. **Comprehensive Book Information** : Book Connect offers detailed book information at your fingertips. Each book entry includes essential details such as author, published date, title, and a summary. This way, you can quickly assess the relevance and appeal of a book before deciding to delve into it.

1. **User-Friendly Interface** : The intuitive layout and smooth navigation ensure a seamless and enjoyable experience while interacting with the app.

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

- In one of my recent projects I used Jsdoc to describe what the function `createBookOverlay` does and to define its param value , which essentially states that the param `booksToRender` is an array and should receive array-like qualities

```
/** The code below deals with displaying the book lists.  
 * @param {object[]} booksToRender - Array of books to render  
 */
```

```
const bookList = document.querySelector('[data-list-items]');  
const fragment = document.createDocumentFragment();  
let currentPage = 1; // add currentPage variable to keep track of the current page
```

```
const createBookOverlay = (booksToRender) => {  
  for (let i = 0; i < booksToRender.length; i++) {  
    const { author: authorId, id, image, title } = booksToRender[i];
```

```
    const element = document.createElement('button');  
    element.className = 'preview';  
    element.setAttribute('data-preview', id);
```

```
    element.innerHTML = /* html */ `  
        
      <div class="preview__info">
```

```

    <h3 class="preview__title">${title}</h3>
    <div class="preview__author">${authors[authorId]}</div>
  </div>
`;

fragment.appendChild(element);

// Add a click event listener to the button element
element.addEventListener('click', (event) => {
  const button = event.target;
  console.log(button);
});
}
bookList.appendChild(fragment);
};

```

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

```

//@ts-check

/** The code below deals with displaying the book lists.
 * @param {object[]} booksToRender - Array of books to render
 */

```

- I added the @ts-check keyword after two forward slashes, which helped catch potential type errors and provided type suggestions and improvements, which enhanced my code quality and reduced bugs.

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

- I utilized the "@link" feature to indicate the specific location where the code for the "createOrderHtml" and "createOrderData" functions can be found, ensuring that my future self can easily access them as they are both in different files.

```
/**
 * This function handles the displaying of the form information
 * in the ordered column after the form is submitted.
 * @param {Event} event - The submit event object.
 * see {@link viewFile.createOrderHtml}
 * see {@link dataFile.createOrderData}
 */
const handleAddSubmit = (event) => {
  event.preventDefault();
  const overlay = viewFile.html.add.overlay;
  const formData = new FormData(event.target);
  const data = Object.fromEntries(formData.entries());
  const newData = dataFile.createOrderData(data);
  const htmlData = viewFile.createOrderHtml(newData);
  const append = document.querySelector('[data-column="ordered"]');
  event.target.reset();
  overlay.close();
  append.appendChild(htmlData);
};
```
