DWA_03.4 Knowledge Check_DWA3.1

---

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

```
# Book Connect
## Intro

BookConnect is a social media platform for book enthusiasts. It aims to provide a better user experience by allowing users to load more books, search for specific books based on criteria like title, author, and genre, and receive relevant search results.

## Code Functionality

This codebase provides functionality for a book preview system aimed at a target audience of book enthusiasts. It includes features such as a light/dark mode toggle, book search and filtering, and the default display of book previews. The code is written in JavaScript and uses HTML elements and CSS styles to create an interactive and visually appealing book preview experience.

The light/dark mode toggle enables users to switch between light and dark themes. The theme settings are stored in an object and applied to the document's body using CSS custom properties. Users can save their selected theme or cancel the changes.

The book search and filtering feature enables users to search for books based on title, author, and genre. The search form captures user input, filters the books array accordingly, and displays the matching results. Pagination is implemented to limit the number of displayed results per page.

The default book preview display showcases a set number of books by default. It creates preview elements dynamically using the provided book data. Each preview element includes the book's image, title, and author. The default display is initialized upon page load.

This codebase aims to provide an interactive and user-friendly book preview system, allowing users to explore and discover books easily.

## Features

- **Light/Dark Mode Toggle:** Users can switch between light and dark themes for a personalized viewing experience.
- **Book Search and Filtering:** Users can search for books based on title, author, and genre, and the system filters and displays the matching results.
- **Default Book Previews:** The system displays a set number of book previews by default, providing an initial overview of the available books.
- User-friendly

## Technologies Used

The project uses the following technologies:

- **HTML:** Structuring the web pages and user interface elements.
- **CSS:** Styling the user interface and creating visual effects.
- **JavaScript:** Implementing the logic and interactivity of the book preview system.

## Code Structure

The codebase consists of several JavaScript functions, event listeners, and HTML elements that work together to create the book preview system. Here is a high-level overview of the code structure:

- **Light/Dark Mode Toggle:** Implements the functionality to switch between light and dark themes based on user selection.
- **Book Search and Filtering:** Handles the user input from the search form, filters the book collection based on the specified criteria, and displays the filtered results.
- **Default Book Previews:** Initializes the default book previews upon page load, creating preview elements dynamically based on the book data.

## Usage

```

```
43      To use the book preview system, please follow these steps:
44
45      - Clone the repository to your local machine.
46      - Open the index.html file in a web browser.
47      - Explore the book previews, toggle between light and dark modes, and use the search form to filter the
        book collection.
48
49      ## Contribution
50
51      Contributions to the project are welcome. If you find any issues or have suggestions for improvements,
        please open an issue or submit a pull request on the project repository.
52
53      ### Requirements
54
55      - Basic {HTML, CSS and JavaScript} skills (https://www.w3schools.com)
56      - IDE eg. {Visual Studio Code}(https://code.visualstudio.com)
57      - {Chrome} browser (https://www.google.com/chrome)
```

---

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

```
1     /**
2      * Handles the light/dark mode toggle functionality based on user selection.
3      * @param selectedTheme - The selected theme ('day' or 'night').
4      * @type {string}
5      */
6
7     const themeSettings = {
8       day: {
9         dark: "10, 10, 20",
10        light: "255, 255, 255",
11      },
12      night: {
13        dark: "255, 255, 255",
14        light: "10, 10, 20",
15      },
16    };
```

---

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

```
1     // @ts-check
2
3     const themeSettings = {
4       day: {
5         dark: "10, 10, 20",
6         light: "255, 255, 255",
7       },
8       night: {
9         dark: "255, 255, 255",
10        light: "10, 10, 20",
11      },
12    };
13
```

---

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

```
// Show Book Details Function

function showBookDetails(event) {
  const overlay = document.querySelector("[data-list-active]");
  const title = document.querySelector("[data-list-title]");
  const subtitle = document.querySelector("[data-list-subtitle]");
  const description = document.querySelector("[data-list-description]");
  const image = document.querySelector("[data-list-image]");
  const imageBlur = document.querySelector("[data-list-blur]");

  if (event.target.dataset.id) {
    overlay.show();
  }

  if (event.target.dataset.title) {
    title.innerHTML = event.target.dataset.title;
  }

  if (event.target.dataset.subtitle) {
    subtitle.innerHTML = event.target.dataset.subtitle;
  }

  if (event.target.dataset.description) {
    description.innerHTML = event.target.dataset.description;
  }

  if (event.target.dataset.image) {
    image.setAttribute("src", event.target.dataset.image);
    imageBlur.setAttribute("src", event.target.dataset.image);
  }
}

document.querySelector("[data-list-items]")
document.addEventListener("click", showBookDetails);

document.querySelector("[data-list-close]").addEventListener("click", () => {
  document.querySelector("[data-list-active]").close();
});
```