

DWA_03.4 Knowledge Check_DWA3.1

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

MALMMA899_Bcl2023_BCL_od_IWA19

Book Connect App

This is an app that is run by a small startup group called Book Connect. Their app is run on social media and it specifically targets book enthusiasts. As part of my final capstone project I was tasked with fixing the code that was done by a freelance developer

Features

User friendly -Allows user to search and preview different books -Has filter search -Has dark and light mode

Getting started

The following is required:

- Visual Studio code or any other IDE
- Basic Html,CSS and Javascript skills (<https://developer.mozilla.org/en-US/docs/Learn>)
- Browser (<https://www.google.com/chrome>)

User stories

- As a user, I want to view a list of book previews, by title and author, so that I can discover new books to read.
- As a user, I want an image associated with all book previews so that I can recognize a book by the cover even if I forgot the name.
- As a user, I want to have the option of reading a summary of the book so that I can decide whether I want to read it.
- As a user, I want to have the option of seeing the date that a book was published so that I can determine how easy it is to obtain second-hand.
- As a user, I want to find books based on specific text phrases so that I don't need to remember the entire title of a book.
- As a user, I want to filter books by author so that I can find books to read by authors that I enjoy.
- As a user, I want to filter books by genre so that I can find books to read in genres that I enjoy.
- As a user, I want to toggle between dark and light modes so that I can use the app comfortably at night.

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

```

/**
 * A handler that allows the user to get different author options once they open the search button. In order to allow this to happen
 * the options must be appended to the HTML. Then a for loop allows the different search options for the author to be viewed. This will
 * all the different book titles and names of the authors.
 * @returns
 */
const createAuthorOptionsHtml = () =>{
  const fragment = document.createDocumentFragment()

  const option = document.createElement('option') //added const
  option.value = "any" //changed element to option
  option.innerText = "All Authors" //added .innerText
  html.search.authors.appendChild(option)

  for (const [id, names] of Object.entries(genres)) {
    const element = document.createElement('option') //added const
    element.value = id //changed element to option
    element.innerText = names //added .innerText
    fragment.appendChild(element)
  }
  return fragment
}

```

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

```

//@ts-check
/**
 * A handler that fires when a user drags over any element inside a column. In
 * order to determine which column the user is dragging over the entire event
 * bubble path is checked with `event.path` (or `event.composedPath()` for
 * browsers that don't support `event.path`). The bubbling path is looped over
 * until an element with a `data-area` attribute is found. Once found both the
 * active dragging column is set in the `state` object in "data.js" and the HTML
 * is updated to reflect the new column.
 *
 * @param {Object} event
 */
const handleDragOver = (event) => {
  event.preventDefault();
  const path = event.path || event.composedPath()
  let column = null

  for (const element of path) {
    const { area } = element.dataset
    if (area) {
      column = area
      break;
    }
  }

  if (!column) return
  updateDragging({ over: column })
  updateDraggingHtml({ over: column })
}

```

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

```
//Import all the data to be used from the necessary javascripts
import { TABLES, COLUMNS, state, createOrderData, updateDragging } from "./data.js";
import { createOrderHtml, html, updateDraggingHtml, moveToColumn } from "./view.js";
```
