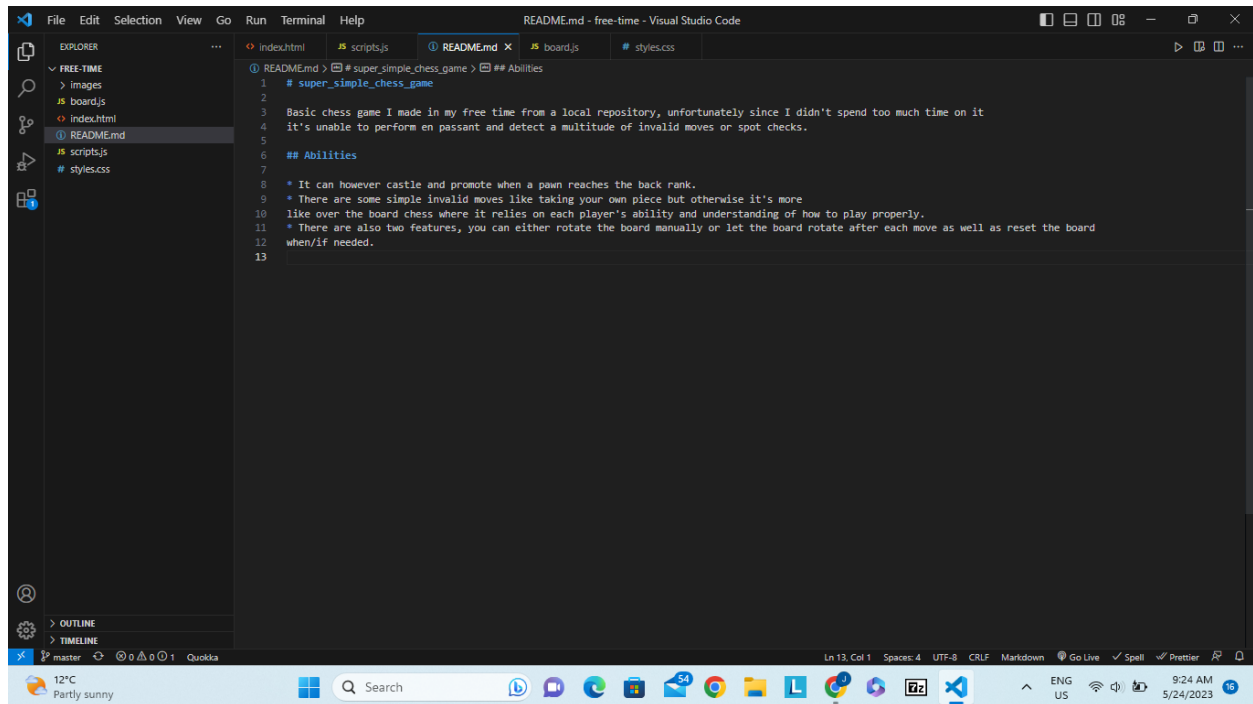
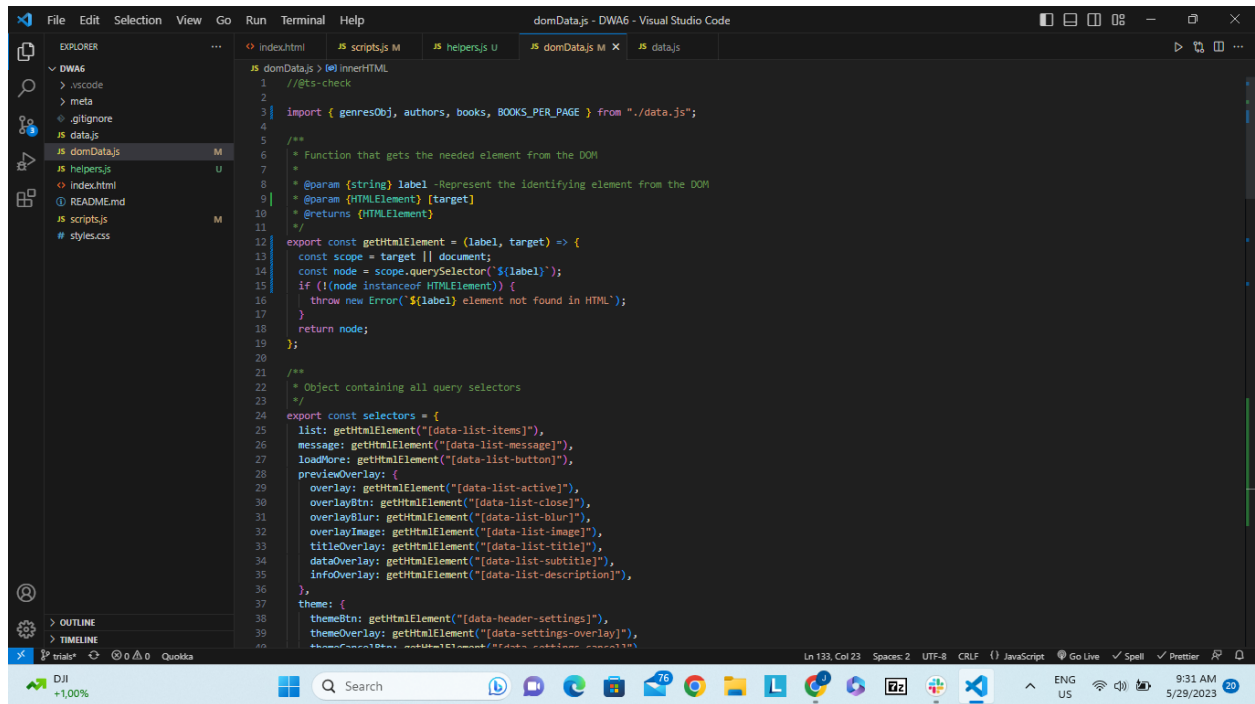


DWA_03.4 Knowledge Check_DWA3.1

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



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



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

```
1 // @ts-check
2
3 import { genresObj, authors, books, BOOKS_PER_PAGE } from './data.js';
4
5 /**
6  * Function that gets the needed element from the DOM
7  *
8  * @param {string} label -Represent the identifying element from the DOM
9  * @param {HTMLElement} [target]
10  * @returns {HTMLElement}
11  */
12 export const getHtmlElement = (label, target) => {
13   const scope = target || document;
14   const node = scope.querySelector(`${label}`);
15   if (!node instanceof HTMLElement) {
16     throw new Error(`${label} element not found in HTML`);
17   }
18   return node;
19 };
20
21 /**
22  * Object containing all query selectors
23  */
24 export const selectors = {
25   list: getHtmlElement("[data-list-items]"),
26   message: getHtmlElement("[data-list-message]"),
27   loadMore: getHtmlElement("[data-list-button]"),
28   previewOverlay: {
29     overlay: getHtmlElement("[data-list-active]"),
30     overlayBtn: getHtmlElement("[data-list-close]"),
31     overlayBlur: getHtmlElement("[data-list-blur]"),
32     overlayImage: getHtmlElement("[data-list-image]"),
33     titleOverlay: getHtmlElement("[data-list-title]"),
34     dataOverlay: getHtmlElement("[data-list-subtitle]"),
35     infoOverlay: getHtmlElement("[data-list-description]"),
36   },
37   theme: {
38     themeBtn: getHtmlElement("[data-header-settings]"),
39     themeOverlay: getHtmlElement("[data-settings-overlay]"),
40     themeFormBtn: getHtmlElement("[data-settings-submit]"),
41   }
42 };
43
44 // ... (rest of the code) ...
```

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