

Vuyiswa Diale

IWA_19 Capstone

ABOUT IWA_19 CAPSTONE

I was required to fix all critical issues, as well as to do a full audit of existing code. The goal is to highlight areas where improvements can be made to performance, readability and future maintainability.

When I was auditing the existing code I had to make some changes in order to meet the user's stories and for the code to run as intended.

Contents

- | | |
|-------------------------|---------------------|
| 1. Exports and Imports. | 9. Search Button. |
| 2. Const. | 10. Book List. |
| 3. Error Message. | 11. Filters. |
| 4. Fragment. | 12. Show More Books |
| 5. Genres. | |
| 6. Authors. | |
| 7. Themes. | |
| 8. Data Settings Theme. | |
| 9. Settings Button. | |

```
index.html | styles.css | JS scripts.js M | JS data.js | ...
```

```
JS scripts.js > ...
1 //imported these from the data.js file
2 import { books, authors,
  BOOKS_PER_PAGE, genres } from "./data.
  js";
3
4
5
6
7
8
9
10
11
```

```
JS data.js > ...
1 // added const to these variables and e
2 export const BOOKS_PER_PAGE = 36;
3
4 > export const authors = { ...
  92 }
  93
  94 > export const genres = { ...
  163 }
  164
  165 > export const books = [ ...
  25413 ]
  25414
```

```
JS orgnal.js 9+, M | ...
```

```
JS orgnal.js > ...
1 matches = books
2 page = 1;
3
4
```

EXPORTS AND IMPORTS

- 'BOOKS_PER_PAGE, books, authors, and genres' have to be imported from the 'data.js file'.
- Added 'const' and 'export' to BOOKS_PER_PAGE, books, authors, and genres.

CONST

- Added 'const' to 'matches' on the 'scripts.js' file.

```
scripts.js - VUYDIA913...  
index.html JS orgnal.js 9+, M X JS da  
JS orgnal.js > ...  
1 matches = books  
2 page = 1;  
3  
4  
5  
JS scripts.js M X  
JS scripts.js > 📦 headerButton.addEventListener('c  
4  
5 //added const and let the the be  
variables  
6 const matches = books;  
7 let page = 1;  
8
```

ERROR MESSAGE

```
//scripts.js
// display message if no books match filters
if (filteredBooks.length > 0) {
  document.querySelector('[data-list-message]').innerText =
  '';
  document.querySelector('[data-list-button]').disabled =
  true;
  document.querySelector('[data-list-message]').style.marginTop = '-125px';
} else {
  document.querySelector('[data-list-message]').innerText =
  'No results found. Your filters might be too narrow.';

  document.querySelector('[data-list-button]').disabled =
  true;
}

//orgnal.js
if (!books && !Array.isArray(books)) throw new
Error('Source required');
if (!range && range.length < 2) throw new
Error('Range must be an array with two numbers');
```

- This code checks if any books have passed the filters by checking the length of the `filteredBooks` array.
- If there are filtered books, it sets the text content of an element with the `[data-list-message]` attribute to an empty string, disables a button with the `[data-list-button]` attribute, and adjusts the margin of the `[data-list-message]` element.
- If there are no filtered books, it sets the text content of the `[data-list-message]` element to a message that tells the user that there are no results based on their filter criteria, and disables the `[data-list-button]` button.

FRAGMENT

- Added 'let' to 'fragment' as it will be changing.
- Fixed errors on the 'const' objects of the extracted and added 'let' to the fragment variable.
- Fixed the for loop syntax by removing the ';' and added 'of' for readability.
- The 'const preview' removed to be more clean and readable.
- Fixed the syntax and wrote code in a more correct and readable way.

JS orgnal.js > ...

```
19
20
21
22
23   fragment = document.
      createDocumentFragment()
24   const extracted = books.slice(0,
      36)
25
26
27   for ({ author, image, title,
      id }; extracted; i++) {
28     const preview = createPreview
      ({
29       author,
30       id,
31       image,
32       title
33     })
34
35
36     fragment.appendChild(preview)
37   }
38
39   data-list-items.appendChild
      (fragment)
40
41
42
43
44
45
46
47
48
49
```

JS scripts.js > headerButton.addEventListener("click",

```
9
10   // Showing 36 Books Per Page
11   let fragment = document.
      createDocumentFragment(); //
      added let to the fragment
12   const extracted = books.slice(0,
      36);
13
14
15   /**
16    * fixed errors on the const and
      added 'let' to the fragment
      variable
17    * fixed the syntax by removing
      the ';' and added 'of' on the
      for
18    * the const preview was
      re-ordered in a more readable
      manner
19    */
20
21   for (const { author, title,
      image, id } of extracted) {
22     const preview = createPreview
      ({ author, id, image,
      title });
23     fragment.appendChild
      (preview);
24   }
25   //fixed the syntax and wrote
      code in a more readable way
26   document.querySelector("
      [data-list-items]").appendChild
      (fragment);
27
28
```

GENRES

JS orgnal.js > ...

```
49
50 genres = document.
   createDocumentFragment()
51 element = document.createElement
   ('option')
52 element.value = 'any'
53 element = 'All Genres'
54 genres.appendChild(element)
55
56
57 for ([id, name]; Object.entries
   (genres); i++) {
58   document.createElement('option')
59   element.value = value
60   element.innerText = text
61   genres.appendChild(element)
62 }
63
64 data-search-genres.appendChild
   (genres)
65
66
67
68
69
70
71
72
73
74
75
76
77
```

JS scripts.js > headerButton.addEventListener('click') callback

```
230
231
232 // Drop Down for Genres
233 const dataSearchGenres = document.
   querySelector("[data-search-genres]");
234 const allGenresOption = document.
   createElement("option");
235   allGenresOption.value = "any";
236   allGenresOption.innerText = "All
   Genres";
237   dataSearchGenres.appendChild
   (allGenresOption);
238   for (const [id, names] of Object.entries
   (genres)) {
239     const element = document.
       createElement("option");
240     element.value = id;
241     element.innerText = names;
242     dataSearchGenres.appendChild
       (element);
243   }
244
245   for (const [id, names] of Object.entries
   (genres)) {
246     const element = document.
       createElement("option");
247     element.value = id;
248     element.innerText = names;
249     dataSearchGenres.appendChild
       (element);
250   }
251
252
```

- Line `element = 'All Genres'` should be `element.innerText = 'All Genres'` to set the text of the option element.
- Fixed code in the line `'element' = 'All Genres'` should be `'element.innerText' = 'All Genres'` to set the text of the option element.
- Changed a `for...of` loop to iterate through the key-value pairs in the `genres` object for the code to use `'Object.entries'` to create an array of key-value pairs, and then uses a `'for...of loop'` to iterate through that array.
- Changed the `'var'` to be `'const'`, to declare the loop variables.

- Changed the variable names.
- Added the correct attribute, instead of 'option', to ('[data-search-authors]')'
- Changed a for...of loop to iterate through the key-value pairs in the genres object for the code to use 'Object.entries' to create an array of key-value pairs, and then uses a 'for...of loop' to iterate through that array.
- Fixed the for...of loop and repeated the same code as for the genre code.
- Used 'dataSearchAuthors' as an element that will be added instead of 'data-search-authors'.

AUTHORS

JS original.js > ...	JS scripts.js > ...
66	252
67 authors = document. createDocumentFragment()	253 // Drop down for authors
68 element = document.createElement ('option')	254 const dataSearchAuthors = document. querySelector("[data-search-authors]");
69 element.value = 'any'	255 const allAuthorsOption = document. createElement("option");
70 element.innerText = 'All Authors'	256 allAuthorsOption.value = "any";
71 authors.appendChild(element)	257 allAuthorsOption.innerText = "All Authors";
72	258 dataSearchAuthors.appendChild (allAuthorsOption);
73	259 > for (const [id, names] of Object. entries(authors)) { ...
74 for ([id, name]; Object.entries (authors); id++) {	264 }
75 document.createElement ('option')	265 });
76 element.value = value	266
77 element = text	267
78 authors.appendChild(element)	268
79 }	269
80	270
81 data-search-authors.appendChild (authors)	271
82	272

THEME

JS orgnal.js > ...

```
10
11 |
12   day = {
13     dark: '10, 10, 20',
14     light: '255, 255, 255',
15   }
16
17
18   night = {
19     dark: '255, 255, 255',
20     light: '10, 10, 20',
21   }
22
23
```

JS scripts.js > ...

```
281
282   // Define the color values for the day and night
  themes
283   const css = {
284     day: {
285       dark: '10, 10, 20',
286       light: '255, 255, 255',
287     },
288     night: {
289       dark: '255, 255, 255',
290       light: '10, 10, 20',
291     },
292   };
293
```

- Added const to the day and night properties and made them css properties.

```

data-settings-theme.value ===
window.matchMedia && window.
matchMedia('
(prefers-color-scheme: dark)').
matches ? 'night' : 'day'
v = window.matchMedia &&
window.matchMedia('
(prefers-color-scheme: dark)').
matches? 'night' | 'day'

```

```

JS scripts.js > ...
294 const dataSettingsTheme = document.querySelector
    ('[data-settings-theme]')
295 const saveButton = document.querySelector("
    [data-settings-overlay] button.
    overlay__button_primary")
296
297 // Add a click event listener to the save button
298 saveButton.addEventListener('click', (event) => {
299     event.preventDefault();
300     // Get the selected theme from the theme
        select element
301     const selectedTheme = dataSettingsTheme.
        value;
302     // Set the body element's background colors
        based on the selected theme
303     if (selectedTheme === 'day') {
304         document.querySelector('body').style.
            setProperty('--color-dark', css.day.
            dark);
305         document.querySelector('body').style.
            setProperty('--color-light', css.day.
            light);
306     } else if (selectedTheme === 'night') {
307         document.querySelector('body').style.
            setProperty('--color-dark', css.night.
            dark);
308         document.querySelector('body').style.
            setProperty('--color-light', css.night.
            light);
309     };
310
311 // Hide the settings overlay
312 document.querySelector("
    [data-settings-overlay]").style.display =
    "none"
313 });

```

DATA SETTINGS THEME

- Used `querySelector` to get the `data-settings-theme` attribute.
- Added an `addEventListener` to the save button so it can submit the changes.
- Used an `if else` statement for when the user selects the day theme it should use the `css` property for day which is the light color.
- To continue the `if else` statement would then be `if` a user selects night then it also use the `css` property for night which will be the dark color.

JS scripts.js > ...

```
274 // Theme Mode
275 //Buttons
276 const settingsButton = document.querySelector
277 ('[data-header-settings]');
278 settingsButton.addEventListener('click',
279 (event) => {
280     event.preventDefault();
281     const themeOverlay = document.
282     querySelector('[data-settings-overlay]');
283     themeOverlay.showModal();
284
285     // Get the cancel button and add a click
286     event listener to close the theme
287     overlay dialog
288     const settingsCancelBtn = document.
289     querySelector('[data-settings-cancel]');
290     settingsCancelBtn.addEventListener
291     ('click', () => {
292         const themeOverlay = document.
293         querySelector('
294         [data-settings-overlay]');
295         themeOverlay.close();
296     });
297 });
298 });
299
```

SETTINGS BUTTONS

- Created a settings button to open the data settings overlay.
- Added a setting cancel button to close the data settings overlay.
- Added eventListeners to these buttons.

SEARCH BUTTON

```
// Open the Search Button overlay
const searchForm = document.createElement('form');
searchForm.classList.add('search-form');

//Header Button
const headerButton = document.querySelector('.header__button');

headerButton.addEventListener('click', (event) => {
  event.preventDefault();
  const searchOverlay =
document.querySelector('[data-search-overlay]');
  searchOverlay.showModal();

  const cancelButton =
document.querySelector('[data-search-cancel]');
  cancelButton.addEventListener('click', () => {
    const searchOverlay =
document.querySelector('[data-search-overlay]');
    searchOverlay.close();
  });

  const searchButton =
document.querySelector('[data-search-form]');
  searchButton.addEventListener('submit', (event) => {
    event.preventDefault();

    const searchTotal =
document.querySelector('[data-search-title]');
    console.log(searchTotal.value)
  })
  console.log(searchButton);
```

- Created a search button in order to allow the user to search for more books.
- Opened a search button overlay using the searchForm variable I have created.
- Added an event listener to headerButton so that when it is clicked the overlay will show and when the cancel button is clicked the overlay will close.
- Used the data-search-form attribute in order to submit the data that is searched.

BOOK LIST

```
//Book list
const moreBooks =
document.querySelector("[data-list-button]");
let showMore = page * BOOKS_PER_PAGE;

// show more books button
moreBooks.addEventListener("click", () => {
  const dataListItem =
document.querySelector("[data-list-items]");
  const remaining = matches.slice(showMore,
matches.length);
  const fragment = document.createDocumentFragment();

  for (const { author, title, image, id } of remaining) {
    const preview = createPreview({ author, id, image,
title });
    fragment.appendChild(preview);
  }
  dataListItem.appendChild(fragment);
  showMore += remaining.length;

  moreBooks.disabled = !(matches.length - showMore > 0);
});

moreBooks.innerHTML = /* html */ `
  <span>Show more</span> (
  <span class="list__remaining">${matches.length - showMore
> 0 ? matches.length - showMore : 0
  }</span> )
`;
```

- Added the book list on showMore button so that when it is clicked more books lists will appear.
- Added a const moreBooks and fetched it from the html file using the querySelector and the attribute.
- Added let showMore to be assigned to page and BOOKS_PER_PAGE in order to display a few number of books .
- Other books should be disabled only to show the 36 books per page.
- Appended the fragment to the dataListItem container, which contains the list of books that are currently displayed.

FILTERS

```

141 // Search specific books
142 const searchFilter = document.querySelector('[data-search-form]')
143 // add event listener to search form
144 searchFilter.addEventListener('submit', (event) => {
145     event.preventDefault();
146     // hide book list
147     document.querySelector('[data-list-items]').style.display = 'none'
148     // clear message area
149     document.querySelector('[data-list-message]').innerHTML = ''
150
151     // To Get the Data
152     const formData = new FormData(event.target)
153     const titleA = formData.get('title');
154     const genreA = formData.get('genre');
155     const authorA = formData.get('author');
156
157     // array to store filtered books
158     const filteredBooks = [];
159     // loop through all books
160     for (let i = 0; i < books.length; i++) { ...
161 }
162 // display filtered books
163 document.querySelector('[class="list__message"]').style.display = 'block'
164 // create fragment to hold filtered books
165 const fragment2 = document.createDocumentFragment()
166 for (const { author, image, title, id, description, published } of filteredBooks) { ...
167 }
168 // To add Filtered Books to Message Area
169 const bookList2 = document.querySelector('[class="list__message"]')
170 bookList2.append(fragment2)
171 document.querySelector('[data-search-form]').reset()
172 document.querySelector("[data-search-overlay]").close()
173 });

```

- This code handles the event of a search form being submitted.
- It gets the user input from the form, filters the books based on the input, and displays the filtered books in a preview format.
- It also adds the preview buttons to a fragment, which is then appended to the message area.
- Finally, it resets the search form and closes the search overlay. If no books match the filters, it displays a message indicating this.

SHOW MORE BOOKS

```
// ShowMore Button
function createPreview({ author, id, image, title
}) { //the for loop was changed into a
createPreview function
    const preview =
document.createElement("div");//created a preview
variable and const it and assign it to the div
element
    preview.classList.add("preview");
    const info = /*html*/ `


    <div class="preview__info">
        <h3 class="preview__title">${title}</h3>
        <div
class="preview__author">${authors[author]}</div>
        </div>
    `;
    preview.innerHTML = info;
    preview.dataset.preview = id; // add book ID
as a data attribute to the preview element
    return preview;
}
```

- **Changed a for loop into a createPreview function.**
- **Created a preview variable and const it and assigned it to the div element.**
- **Used the html elements for the display of the preview.**
- **Added a book ID as a data attribute to the preview element**