Refilwe Nqatyelwa

IWA_19 Capstone

About IWA_19 Capstone

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

I had a week to make sure that the user story is provided is met so that the app can work as intended.

Contents

Exports and Imports.

2. Const.

3. Error Message.

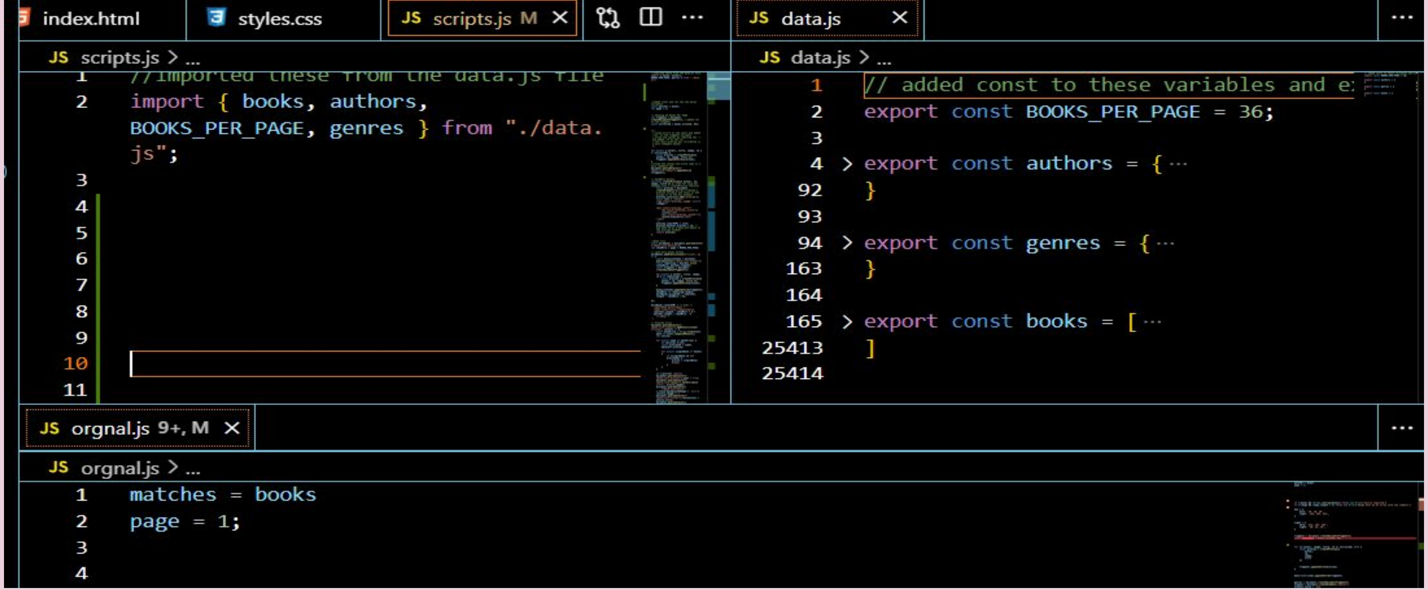
4. Fragment.

5. Genres.

6 Authors.

7. Show More Books.

8. Search Button



```
//imported these from the data.js file
import { books, authors, BOOKS_PER_PAGE, genres } from "./data.js";
```

```
// added const to these variables and e
export const BOOKS_PER_PAGE = 36;

export const authors = {···
}

export const genres = {···
}

export const books = [···
]
```

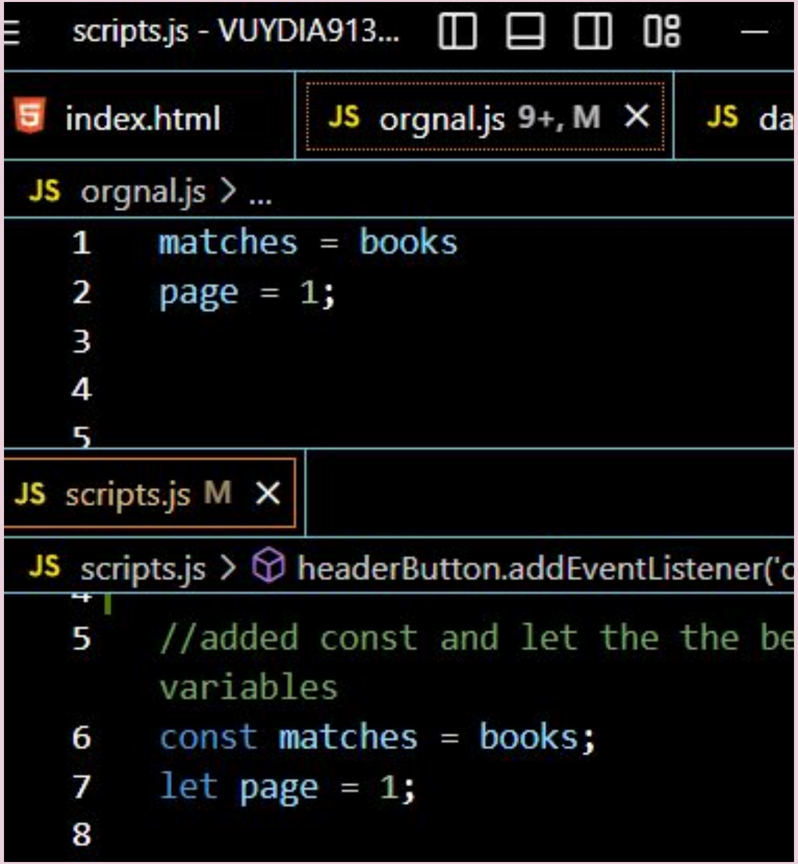
```
matches = books
page = 1;
```

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.

EXAMPLE OF CONST

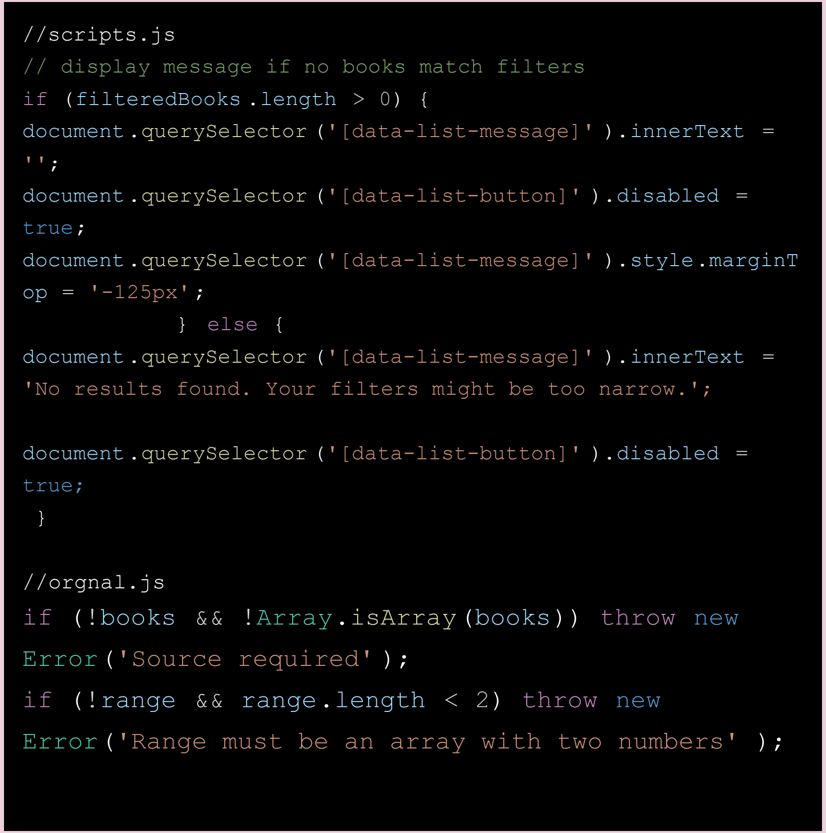
- Added ‘const’ to ‘matches’ on the ‘scripts.js’ file so that the variable can be defined.


scripts.js - VUYDIA913...
index.html
JS orgnal.js 9+, M
JS da
JS orgnal.js > ...
JS scripts.js M
JS scripts.js > headerButton.addEventListener('c

```
matches = books
page = 1;
```

```
//added const and let the the be
variables
const matches = books;
let page = 1;
```

EXAMPLE OF AN 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.marginT
op = '-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

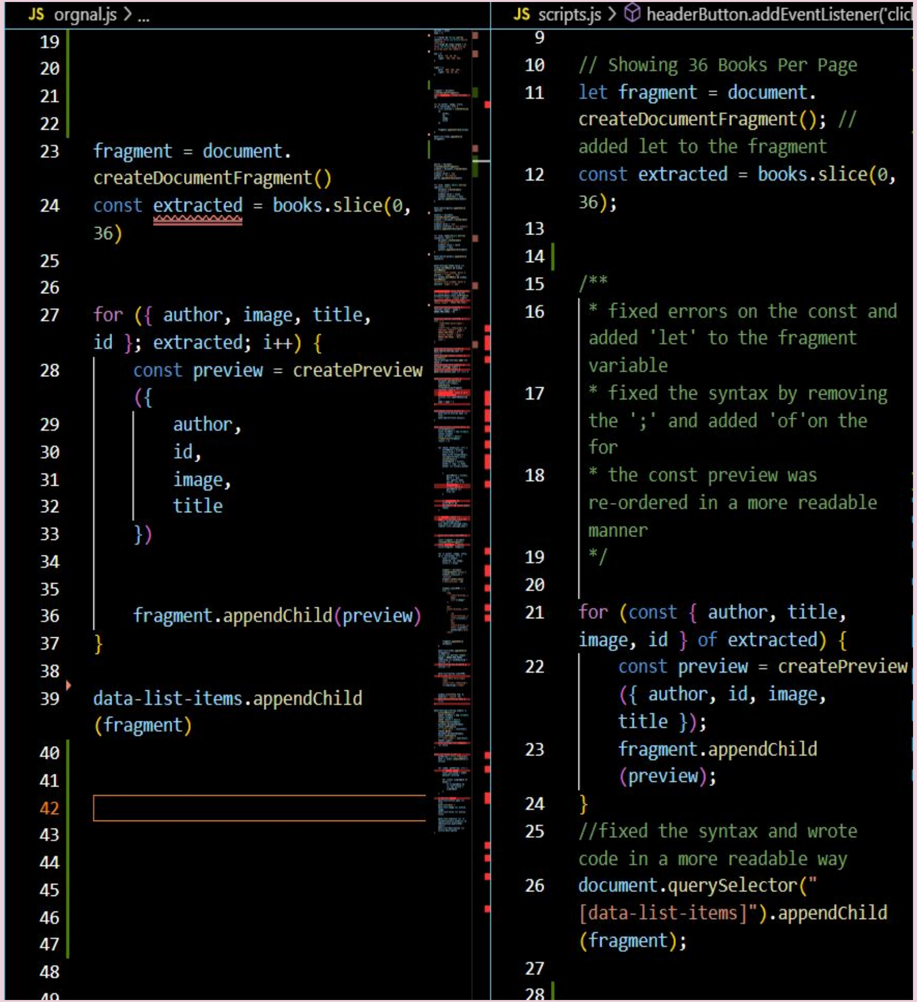
- Added ‘let’ to ‘fragment’ as it will be changing.

- 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.

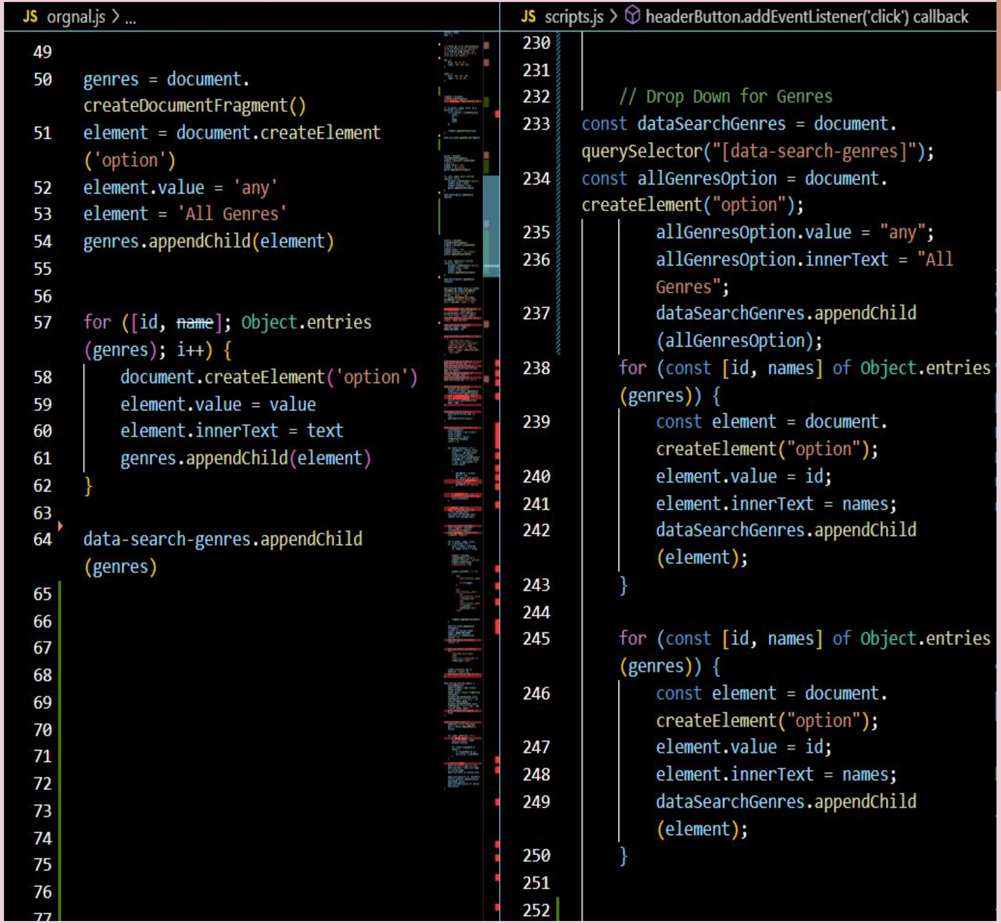
- The ‘const preview’ removed to be more clean and readable.

- Fixed the syntax and wrote code in a more readable way.



```
fragment = document.createDocumentFragment()
const extracted = books.slice(0, 36)


for ({ author, image, title, id }; extracted; i++) {
    const preview = createPreview({
        author,
        id,
        image,
        title
    })


    fragment.appendChild(preview)
}

data-list-items.appendChild(fragment)
```

```

// Showing 36 Books Per Page
let fragment = document.createDocumentFragment(); // added let to the fragment
const extracted = books.slice(0, 36);


/**
 * fixed errors on the const and added 'let' to the fragment variable
 * fixed the syntax by removing the ';' and added 'of'on the for
 * the const preview was re-ordered in a more readable manner
 */

for (const { author, title, image, id } of extracted) {
    const preview = createPreview({ author, id, image, title });
    fragment.appendChild(preview);
}
//fixed the syntax and wrote code in a more readable way
document.querySelector("[data-list-items]").appendChild(fragment);
```



JS orgnal.js > ...

```
genres = document.createDocumentFragment()
element = document.createElement('option')
element.value = 'any'
element = 'All Genres'
genres.appendChild(element)


for ([id, name]; Object.entries(genres); i++) {
    document.createElement('option')
    element.value = value
    element.innerText = text
    genres.appendChild(element)
}

data-search-genres.appendChild(genres)
```

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

```


    // Drop Down for Genres
const dataSearchGenres = document.querySelector("[data-search-genres]");
const allGenresOption = document.createElement("option");
        allGenresOption.value = "any";
        allGenresOption.innerText = "All Genres";
        dataSearchGenres.appendChild(allGenresOption);
    for (const [id, names] of Object.entries(genres)) {
        const element = document.createElement("option");
        element.value = id;
        element.innerText = names;
        dataSearchGenres.appendChild(element);
    }

    for (const [id, names] of Object.entries(genres)) {
        const element = document.createElement("option");
        element.value = id;
        element.innerText = names;
        dataSearchGenres.appendChild(element);
    }


```

Genres

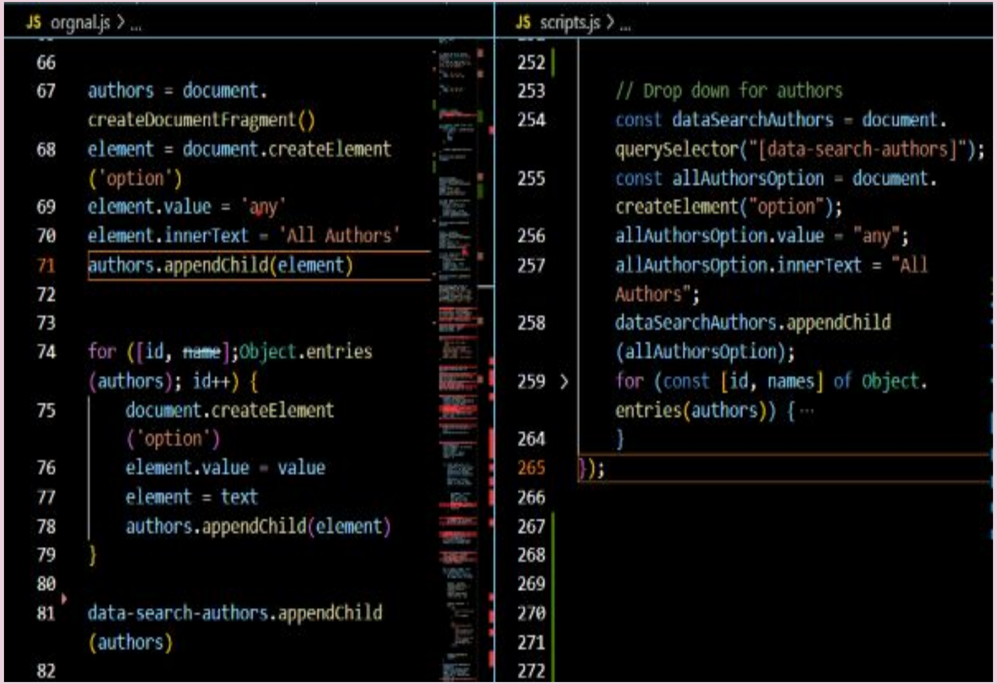
- Line element = 'All Genres' should be element.innerText = 'All Genres' to set the text of the option element.
-
- Fixed code in the line ‘element’ = 'All Genres' should be ‘element.innerText’ = 'All Genres' to set the text of the option element.

- 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.

AUTHORS

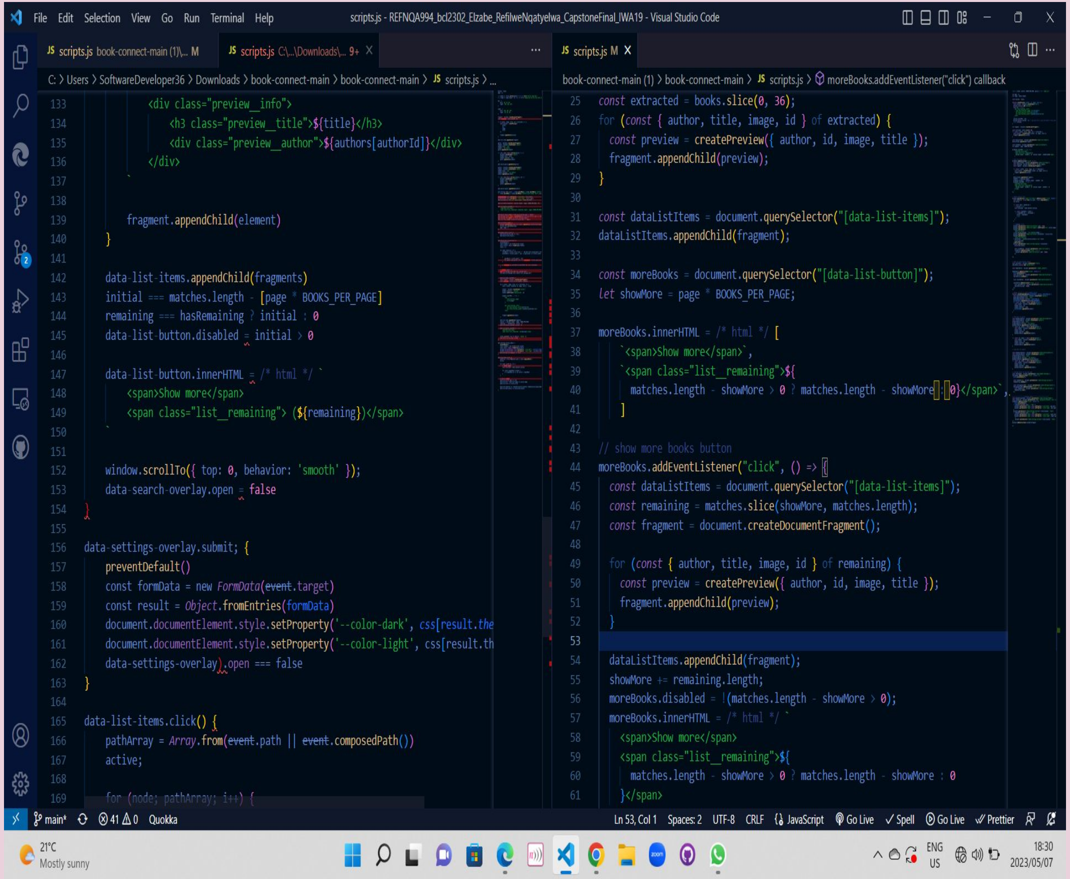
- Changed the variable names.
- Added the correct attribute, instead of ‘option’, to (‘[data-search-authors]’)’
- Changed a for...of loop to iterate through the key-value pairs in the genres object for the code to use ‘Object.entries’ to create an array of key-value pairs, and then uses a ‘for...of loop’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



```
authors = document.createDocumentFragment()
element = document.createElement('option')
element.value = 'any'
element.innerText = 'All Authors'
authors.appendChild(element)


for ([id, ~~name~~];Object.entries(authors); id++) {
    document.createElement('option')
    element.value = value
    element = text
    authors.appendChild(element)
}

data-search-authors.appendChild(authors)
```

```

    // Drop down for authors
    const dataSearchAuthors = document.querySelector("[data-search-authors]");
    const allAuthorsOption = document.createElement("option");
    allAuthorsOption.value = "any";
    allAuthorsOption.innerText = "All Authors";
    dataSearchAuthors.appendChild(allAuthorsOption);
    for (const [id, names] of Object.entries(authors)) {···
    }
});
```

Show more books

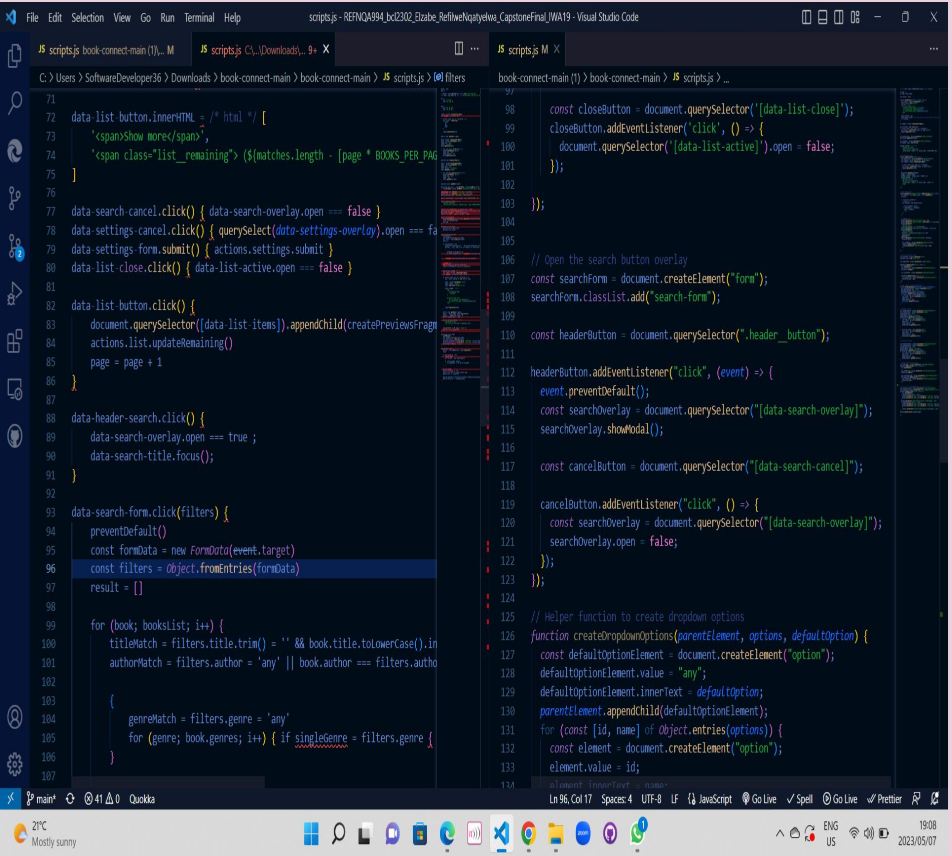

File Edit Selection View Go Run Terminal Help
scripts.js - REFNQA994_bcl2302_Elzabe_RefilweNqatyelwa_CapstoneFinal_IWA19 - Visual Studio Code
JS scripts.js book-connect-main (1)\... M
JS scripts.js C:\...\Downloads\... 9+
JS scripts.js M
C: > Users > SoftwareDeveloper36 > Downloads > book-connect-main > book-connect-main > JS scripts.js > ...
book-connect-main (1) > book-connect-main > JS scripts.js > moreBooks.addEventListener("click") callback
main* ⊗ 41 ⚠ 0 Quokka
Ln 53, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live ✓ Spell Go Live ✓ Prettier
21°C Mostly sunny
ENG US
18:30 2023/05/07

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

    fragment.appendChild(element)
  }

  data-list-items.appendChild(fragments)
  initial === matches.length - [page * BOOKS_PER_PAGE]
  remaining === hasRemaining ? initial : 0
  data-list-button.disabled = initial > 0

  data-list-button.innerHTML = /* html */ `
    <span>Show more</span>
    <span class="list__remaining"> (${remaining})</span>
  `

  window.scrollTo({ top: 0, behavior: 'smooth' });
  data-search-overlay.open = false
}

data-settings-overlay.submit; {
  preventDefault()
  const formData = new FormData(event.target)
  const result = Object.fromEntries(formData)
  document.documentElement.style.setProperty('--color-dark', css[result.the
  document.documentElement.style.setProperty('--color-light', css[result.th
  data-settings-overlay).open === false
}

data-list-items.click() {
  pathArray = Array.from(event.path || event.composedPath())
  active;

  for (node; pathArray; i++) {
```

```
const extracted = books.slice(0, 36);
for (const { author, title, image, id } of extracted) {
  const preview = createPreview({ author, id, image, title });
  fragment.appendChild(preview);
}

const dataListItems = document.querySelector("[data-list-items]");
dataListItems.appendChild(fragment);

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

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

// show more books button
moreBooks.addEventListener("click", () => {
  const dataListItems = 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);
  }

  dataListItems.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 a let to fragment as it will be changing.
- The "showMore" variable tracks the number of items currently displayed and updates accordingly with each click. The "moreBooks" button is disabled if all book items have been displayed and its text changes to reflect how many items remain to be displayed.
- The "showMore" variable tracks the number of items currently displayed and updates accordingly with each click. The "moreBooks" button is disabled if all book items have been displayed and its text changes to reflect how many items remain to be displayed.
- // handle preview click is where i set the event listener for click events and searches through an array to find books that matches the preview dataset value, if it matches it populates the ovalay with the matched book element.

SEARCH BUTTON


File Edit Selection View Go Run Terminal Help
scripts.js - REFNQA994_bcl2302_Elzabe_RefilweNqatyelwa_CapstoneFinal_IWA19 - Visual Studio Code
scripts.js book-connect-main (1)\... M
scripts.js C:\...\Downloads\... 9+
scripts.js M
C: > Users > SoftwareDeveloper36 > Downloads > book-connect-main > book-connect-main > scripts.js > filters
book-connect-main (1) > book-connect-main > scripts.js > ...
main* 41 0 Quokka
Ln 96, Col 17 Spaces: 4 UTF-8 LF JavaScript Go Live Spell Go Live Prettier
21°C
Mostly sunny
ENG
US
19:08
2023/05/07

```
data-list-button.innerHTML = /* html */ [
    '<span>Show more</span>',
    '<span class="list__remaining"> (${matches.length - [page * BOOKS_PER_PAG
]

data-search-cancel.click() { data-search-overlay.open === false }
data-settings-cancel.click() { querySelect(data-settings-overlay).open === fa
data-settings-form.submit() { actions.settings.submit }
data-list-close.click() { data-list-active.open === false }

data-list-button.click() {
    document.querySelector([data-list-items]).appendChild(createPreviewsFragm
    actions.list.updateRemaining()
    page = page + 1
}

data-header-search.click() {
    data-search-overlay.open === true ;
    data-search-title.focus();
}

data-search-form.click(filters) {
    preventDefault()
    const formData = new FormData(~~event~~.target)
    const filters = Object.fromEntries(formData)
    result = []

    for (book; booksList; i++) {
        titleMatch = filters.title.trim() = '' && book.title.toLowerCase().in
        authorMatch = filters.author = 'any' || book.author === filters.autho

        {
            genreMatch = filters.genre = 'any'
            for (genre; book.genres; i++) { if singleGenre = filters.genre {
        }
```

```
    const closeButton = document.querySelector('[data-list-close]');
    closeButton.addEventListener('click', () => {
      document.querySelector('[data-list-active]').open = false;
    });

});


// Open the search button overlay
const searchForm = document.createElement("form");
searchForm.classList.add("search-form");

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.open = false;
  });
});

// Helper function to create dropdown options
function createDropdownOptions(parentElement, options, defaultOption) {
  const defaultOptionElement = document.createElement("option");
  defaultOptionElement.value = "any";
  defaultOptionElement.innerText = defaultOption;
  parentElement.appendChild(defaultOptionElement);
  for (const [id, name] of Object.entries(options)) {
    const element = document.createElement("option");
    element.value = id;
    element.innerText = name;
```

- Created a search overlay form

- When button is clicked the search overlay is opened by calling its showModal() method.

- The code allows the user to search for books either by any, genre or authour.

- The cancel button also has an event listener. When it is clicked, the event listener retrieves the search overlay element again and sets its "open" attribute to false, which closes the search overlay.

- The helper function is for the dropdown, the element takes the parent element to be added to the dropdown,The function creates an option element for the default option with a value of "any" and adds it to the parent element. It then iterates through the object of options and creates an option element for each option with a value of the option's id and text content of the option's name. These option elements are added to the parent element.

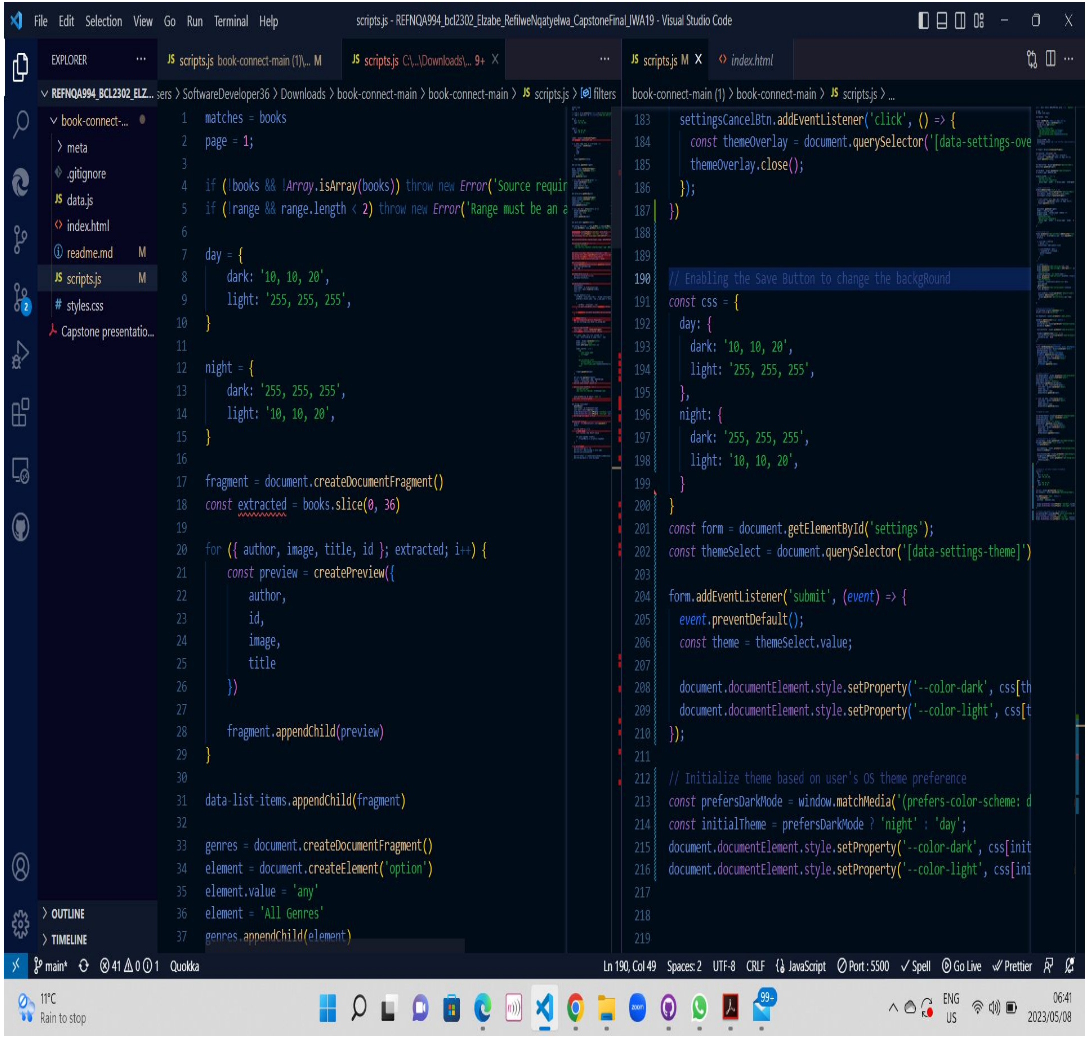
SETTING BUTTON

Allows the user to change the background colors of a webpage by selecting a theme from a dropdown menu and clicking the save button. The css object contains two themes, 'day' and 'night', each with a dark and light color value.

The form variable selects the form element with the ID 'settings', and the themeSelect variable selects the dropdown menu element with the attribute 'data-settings-theme'.

An event listener is added to the form variable, which listens for a form submission. When the form is submitted, the selected theme value is obtained from the themeSelect element, and the --color-dark and --color-light CSS variables of the document.documentElement element are set to the corresponding dark and light color values of the selected theme.

The code also initializes the theme based on the user's OS theme preference, either 'day' or 'night', by using the window.matchMedia method and setting the --color-dark and --color-light CSS variables to the corresponding dark and light color values of the initial theme


File Edit Selection View Go Run Terminal Help
scripts.js - REFNQA994_bcl2302_Elzabe_RefilweNqatyelwa_CapstoneFinal_IWA19 - Visual Studio Code
EXPLORER
REFNQA994_BCL2302_ELZ...
book-connect-...
meta
.gitignore
data.js
index.html
readme.md M
scripts.js M
styles.css
Capstone presentatio...
OUTLINE
TIMELINE
scripts.js book-connect-main (1)\... M
scripts.js C:\...\Downloads\... 9+
...sers > SoftwareDeveloper36 > Downloads > book-connect-main > book-connect-main > scripts.js > filters
scripts.js M
index.html
book-connect-main (1) > book-connect-main > scripts.js > ...
main* ⊗41 ⚠0 ⓘ1 Quokka
Ln 190, Col 49 Spaces: 2 UTF-8 CRLF {} JavaScript Port : 5500 ✓ Spell Go Live ✓ Prettier
11°C
Rain to stop
ENG
US
06:41
2023/05/08

```javascript
matches = books
page = 1;

if (!books && !Array.isArray(books)) throw new Error('Source requir
if (!range && range.length < 2) throw new Error('Range must be an a

day = {
    dark: '10, 10, 20',
    light: '255, 255, 255',
}

night = {
    dark: '255, 255, 255',
    light: '10, 10, 20',
}

fragment = document.createDocumentFragment()
const extracted = books.slice(0, 36)

for ({ author, image, title, id }; extracted; i++) {
    const preview = createPreview({
        author,
        id,
        image,
        title
    })

    fragment.appendChild(preview)
}

data-list-items.appendChild(fragment)

genres = document.createDocumentFragment()
element = document.createElement('option')
element.value = 'any'
element = 'All Genres'
genres.appendChild(element)
```

```javascript
  settingsCancelBtn.addEventListener('click', () => {
    const themeOverlay = document.querySelector('[data-settings-ove
    themeOverlay.close();
  });
})


// Enabling the Save Button to change the backgRound
const css = {
  day: {
    dark: '10, 10, 20',
    light: '255, 255, 255',
  },
  night: {
    dark: '255, 255, 255',
    light: '10, 10, 20',
  }
}
const form = document.getElementById('settings');
const themeSelect = document.querySelector('[data-settings-theme]')

form.addEventListener('submit', (event) => {
  event.preventDefault();
  const theme = themeSelect.value;

  document.documentElement.style.setProperty('--color-dark', css[th
  document.documentElement.style.setProperty('--color-light', css[t
});

// Initialize theme based on user's OS theme preference
const prefersDarkMode = window.matchMedia('(prefers-color-scheme: d
const initialTheme = prefersDarkMode ? 'night' : 'day';
document.documentElement.style.setProperty('--color-dark', css[init
document.documentElement.style.setProperty('--color-light', css[ini
```

CONCLUSSION

Ater I have debugged and fixed all the critical issues the app should be able to do all that the user asked for which are the following:

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. As a user, I want to filter books by author so that I can find books to read by authors that I enjoy.
7. As a user, I want to filter books by genre so that I can find books to read in genres that I enjoy.
8. As a user, I want to toggle between dark and light modes so that I can use the app comfortably at night.