

# IWA19 FINAL CAPSTONE PROJECT - AUDIT & FIX "BOOK CONNECT" APP



Stephany Truscott

STETRU007\_PTO2301\_GroupB

# Introduction

As a junior JavaScript developer, you have been approached by a small startup called Book Connect. They run a social media platform specifically targeted at book enthusiasts. A basic version of the product was created by another freelance developer.

The team was extremely impressed by their HTML and CSS code. However, it eventually became clear that the developer was not entirely honest about their competency with JavaScript. While the client is happy with the current HTML and CSS, they have asked you to review the existing JavaScript code. Currently, it throws several errors preventing the code from even running. You will have to fix all critical issues, as well as do a full audit of the existing code. The goal is to highlight areas where improvements can be made to performance, readability and future maintainability. The client has provided you with prioritised user stories by means of the current backlog. These user stories express the product behaviour as intended, and the app will not be considered as fixed unless all these user stories are met





# AGENDA

Identify Existing Issues

Implement Fixes

Implement User Stories

Final Product

# Identify Existing Issues

- Undeclared Variables
- No flow
- Incorrect Syntax Structures
- Incomplete or missing functions / statements etc.
- Incorrect DOM calling.
- Incorrect HTML TAGS.

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

```
<script src="./scripts.js"></script>
<script src="./data.js"></script>
```

```
for (node; pathArray; i++) {
  if active break;
  const previewId = node?.dataset?.preview

  for (const singleBook of books) {
    if (singleBook.id === id) active = singleBook
  }
}
```

```
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)
```

# Implement Fixes

- Declared all Variables
- Rearranged code stacks in correct order for needed flow
- Corrected Syntax Structures
- Completed & fixed functions / statements etc.
- Corrected DOM calling.
- Corrected HTML TAGS.
- Added comments for better readability

```
import {BOOKS_PER_PAGE,authors,books,genres} from "../data.js"
```

```
//SEARCH FUNCTION - Fetch HTML DOM Elements
const search = document.querySelector("[data-header-search]");
const searchCancel = document.querySelector("[data-search-cancel]");
const searchOverlay = document.querySelector("[data-search-overlay]");
const searchForm = document.querySelector("[data-search-form]");
const searchTitle = document.querySelector("[data-search-title]");
const searchGenres = document.querySelector("[data-search-genres]");
const searchAuthors = document.querySelector("[data-search-authors]");
```

```
settingsForm.addEventListener("submit", (event) => {
  // user selects dark/light
  event.preventDefault();
});
```

```
for (let i = 0; i < genreList.length; i++) {
  const [id, name] = genreList[i];
  const genreOption = document.createElement("option");
  genreOption.value = id;
  genreOption.textContent = name;
  genresFragment.appendChild(genreOption);
}
searchGenres.appendChild(genresFragment);
```

# IMPLEMENT USER STORIES

```
//create range of books to display
const fragment = document.createDocumentFragment();
const extracted = books.slice(0, 36);

for (let i = 0; i < extracted.length; i++) {
  const { author: authorId, id, image, title } = extracted[i];

  const extractedList = document.createElement("button"); //button effect for preview
  extractedList.classList = "preview";
  extractedList.setAttribute("data-preview", id);

  extractedList.innerHTML = /* html */ `
    

    <div class="preview_info">
      <h3 class="preview_title">${title}</h3>
      <div class="preview_author">${authors[authorId]}</div>
    </div>
  `;

  fragment.appendChild(extractedList);
}
listItems.appendChild(fragment);
```

## Book & Image Previews

### 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.

# IMPLEMENT USER STORIES

```
//summary preview setting setup
listItems.addEventListener("click", (event) => {
  const pathArray = Array.from(event.path || event.composedPath());
  let active = null;

  for (let i = 0; i < pathArray.length; i++) {
    const node = pathArray[i];
    if (active) {
      break;
    }
    const previewId = node?.dataset?.preview;

    for (let i = 0; i < books.length; i++) {
      const singleBook = books[i];
      if (singleBook.id === previewId) {
        active = singleBook;
        break;
      }
    }
  }

  if (!active) {
    return;
  }

  listActive.open = true;
  listBlur.src = active.image;
  listImage.src = active.image;
  listTitle.textContent = active.title;

  listSubtitle.textContent = `${authors[active.author]} (${new Date(
    active.published
  ).getFullYear()})`;
  listDescription.textContent = active.description;
});
```



## Summary & Date Previews

### 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.



# IMPLEMENT USER STORIES

```
//SEARCH FUNCTION - Fetch HTML DOM Elements
const search = document.querySelector("[data-header-search]");
const searchCancel = document.querySelector("[data-search-cancel]");
const searchOverlay = document.querySelector("[data-search-overlay]");
const searchForm = document.querySelector("[data-search-form]");
const searchTitle = document.querySelector("[data-search-title]");
const searchGenres = document.querySelector("[data-search-genres]");
const searchAuthors = document.querySelector("[data-search-authors]");
```

+

●

```
//search function
search.addEventListener("click", () => {
  //user clicks on search icon & search bar opens
  searchOverlay.showModal();
  searchTitle.focus();
});

searchCancel.addEventListener("click", () => {
  //user clicks "cancel" button & searchbar closes
  searchOverlay.close();
  searchForm.reset();
});

searchForm.addEventListener("submit", (event) => {
  //user selects requested data in searchbar form, results returned on "search" click
  event.preventDefault();
});
```

○

## Search Function

## 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.



# IMPLEMENT USER STORIES

```
//SETTINGS Function - Fetch HTML DOM Elements
const settings = document.querySelector("[data-header-settings]");
const settingsTheme = document.querySelector("[data-settings-theme]");
const settingsOverlay = document.querySelector("[data-settings-overlay]");
const settingsForm = document.querySelector("[data-settings-form]");
const settingsCancel = document.querySelector("[data-settings-cancel]");
```

+

•

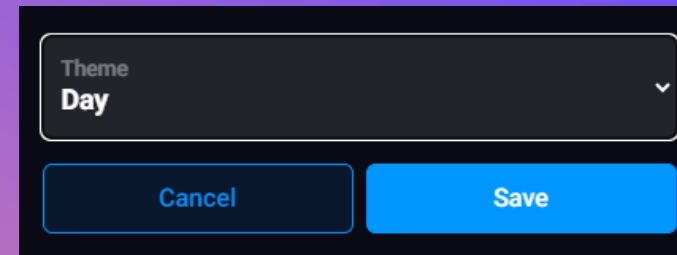
```
settingsForm.addEventListener("submit", (event) => {
  // user selects dark/light
  event.preventDefault();

  const formSubmit = new FormData(event.target);
  const submit = Object.fromEntries(formSubmit);

  if (submit.theme === "night") {
    document.documentElement.style.setProperty(
      "--color-light",
      themes[submit.theme][0]
    );
    document.documentElement.style.setProperty(
      "--color-dark",
      themes[submit.theme][1]
    );
  } else {
    document.documentElement.style.setProperty(
      "--color-light",
      themes[submit.theme][0]
    );
    document.documentElement.style.setProperty(
      "--color-dark",
      themes[submit.theme][1]
    );
  }
  settingsOverlay.close();
});
```

○

## Theme Settings


 A UI mockup of a theme settings overlay. It features a dark background with a white border. At the top, there is a dropdown menu labeled 'Theme' with 'Day' selected. Below the dropdown are two buttons: 'Cancel' and 'Save'. The 'Save' button is highlighted in a bright blue color.

## 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.

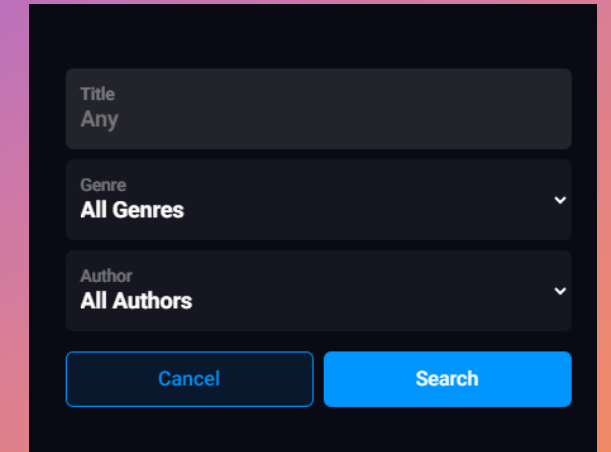
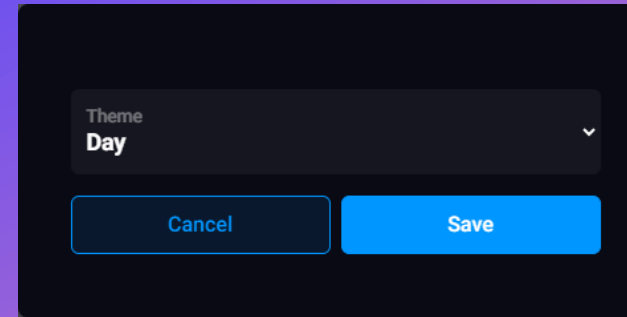
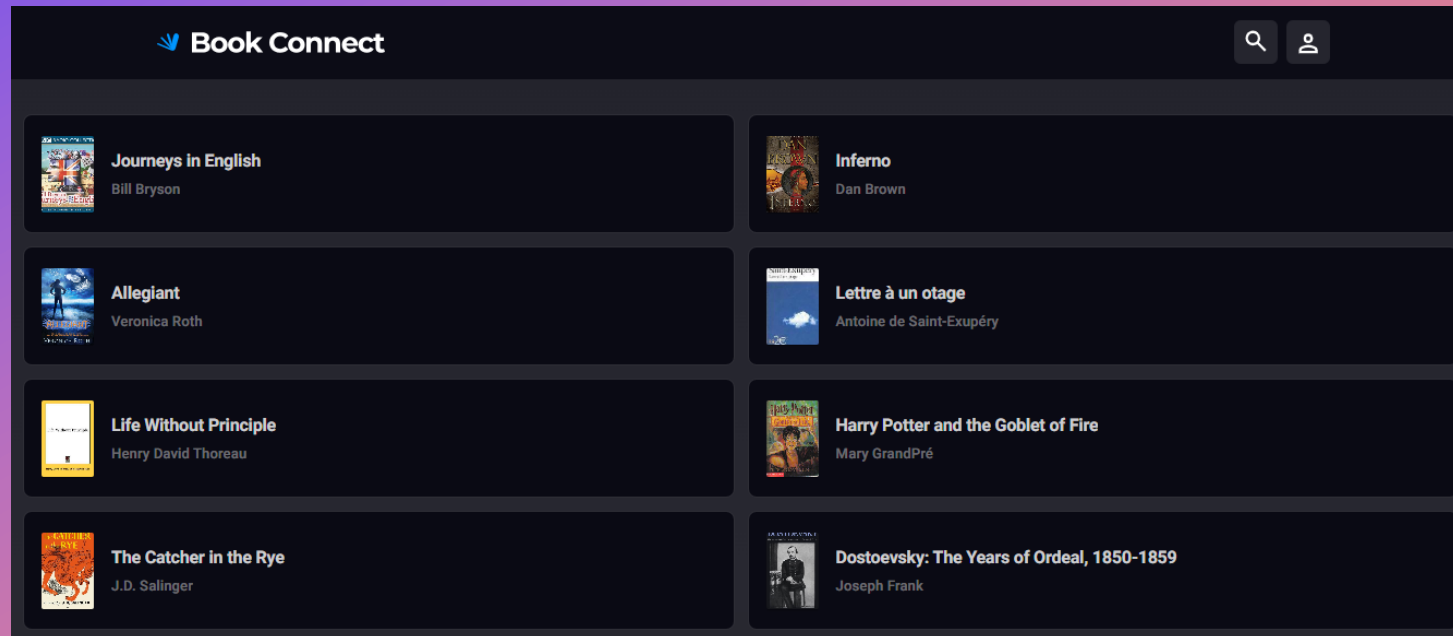
As a user, I want to find books based on specific text phrases so that I don't need to remember the entire title of a book.

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.

# FINAL PRODUCT



+

o

.

# THANK YOU

