

Book connect

- I 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.

- ## The user has the following requirements

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

- **Errors found in code**

- The title variable is not defined, but it's used in the following line: `preview.dataset.title = title[i].title`. It should be `books[i].title` instead.
- In the for loop where genres and authors are being created, there are some errors:
- The loop syntax is incorrect. It should be `for (const [id, name] of Object.entries(genres))` instead of `for ([id, name]; Object.entries(genres); i++)`.
- The `element.innerText` should be assigned the name variable, but instead, it's assigned to `text`.
- The `element` variable is not defined in the first iteration of the loop.
- The `css` variable is not defined anywhere, but it's being used in this line: `const document = document.getElementById.style.setProperty('--color-dark', css[v].dark);`. It should be defined or replaced with `day` and `night` variables.
- The `preventDefault()` method should be called on the event object in the `data-search-form.click(filters)` function.
- In the same function, there are some syntax errors in the for loop that populates the result array:
- The loop should be `for (const book of booksList)` instead of `for (book; booksList; i++)`.
- The `singleGenre` variable is not defined.
- The `genreMatch` variable should be initialized to `false`.
- The if statement that checks for `genreMatch` should use `===` instead of `=`.
- The `data-list-button.disabled` property is being assigned a boolean value, but it should be assigned a string value: `"disabled"` or `""`.
- In the last lines of the code, there are some variables (`initial`, `remaining`, `hasRemaining`) that are not defined anywhere.

This code imports data from a separate file called `data.js`, creates a book list on the page, and allows users to search for and select books. The imported data consists of three arrays, `authors`, `genres`, and `books`. The `books` array contains objects with properties such as `id`, `title`, `author`, `image`, `published`, and `description`. The code also includes a DOM manipulation to create and update various elements on the page, such as search and settings overlays, a book list, and a "Show More" button.

The code begins by importing the `authors`, `genres`, and `books` arrays from another file using ES6 syntax. It then creates a `matches` constant and sets its value to `books`. Next, the code checks to ensure that `books` is not null and is an array, and that `page` is a valid range. If either of these conditions is not met, the code throws an error.

The code then defines two objects, `day` and `night`, which are used to set the color theme of the page. A fragment is created using `document.createDocumentFragment()`, and two variables, `startIndex` and `endIndex`, are initialized to 0 and 36, respectively. The code extracts a subset of the `books` array from `startIndex` to `endIndex` using `Array.slice()` and creates a new preview element for each book using `document.createElement()`. The preview element is then given various dataset attributes and HTML content using template literals. The preview element is then appended to the fragment using `Node.appendChild()`. Finally, the fragment is added to the book list on the page.

The code then defines event listeners for the search and settings buttons on the page. When these buttons are clicked, the corresponding overlays are displayed. The code also creates two select elements for `authors` and `genres` and adds options to them based on the `authors` and `genres` arrays.

The `detailsToggle` function is defined to show the details of a book when a user clicks on it. When a book is clicked, the `detailsToggle` function is called, which updates the `title`, `subtitle`, `description`, and `image` elements on the page with the corresponding dataset attributes of the clicked book. If no book is clicked, the `detailsToggle` function does nothing.

Finally, the code defines an event listener for the "Show More" button, which updates the `startIndex` and `endIndex` variables and extracts a new subset of the `books` array to be displayed on the page. The "Show More" button also updates its text to display the number of remaining items to be displayed.

- The code imports data from a separate file and then creates a list of books with preview images and some metadata. It also includes functionality for a search button, a settings button, and a "show more" button to load additional items. When a user clicks on a book's preview image, it displays a larger image with additional metadata in an overlay.
- The first few lines of code import the necessary data from a file. It then sets up a few constants and objects to define the look and feel of the app, such as the color theme for day and night.
- Next, the code creates a document fragment to store elements to add to the list of books. It then loops through a subset of the books array and creates a preview element for each one. These preview elements include an image, title, author, and other metadata, which are stored in custom data attributes.
- The code then adds the preview elements to the document fragment and appends the fragment to the document to display the list of books.
- The code also sets up event listeners for the search and settings buttons to display and hide overlays when clicked. It also sets up event listeners for selecting authors and genres for search purposes.
- When a user clicks on a book's preview image, the code displays an overlay with a larger image and additional metadata. The code also sets up a "Show More" button to load additional items when clicked.
- The last part of the code sets up an event listener for the "Show More" button to load additional items and append them to the list of books.