

Herman Botshelo Ngwenya

IWA_19

Final Capstone Project Presentation

USER REQUIREMENTS

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.

CRITICAL ERRORS IN THE CODE :

There were no imports or exports within the JS files

Variables , functions and objects were not properly declared.

There were a lot of syntax errors, misuse of symbols, methods and operators, and as well as undefined if statements

For and If statements were not properly used

There were no querySelector methods used to return elements and access necessary HTML Collector items.

Overall, the initial code was complicated and unreadable which made it difficult to refactor the entire code and the intended result to make the Web App fully functional was not achievable.

CHANGES AND IMPLEMENTATION TO THE CODE.

I implemented imports to import data from data.js ,which is arrays consisting of all the data and properties needed for the book connect app .

I then created variables and properly declared all relevant variables as well as the essential functions and objects within the code.

I used the querySelector method to return elements that match css selectors and access the HTML classes and divs.

I created fragments for the preview of books, authors and genres.

I used functions to enable display of corresponding overlays when the search and setting buttons are clicked and I added event listeners within those functions.

I changed the syntax and format of the for statements to create loops and I fixed the syntax within the if statements

Finally I changed the whole syntax of the code, I added spaces in between the code and I added comments in order to improve the readability and maintainability of my code.

MEETING THE USER STORIES

The preview list of books displaying the image, title and author is displayed.

A summarized description of the book is displayed when each book is clicked on.

The search form is functional and books can be found by specific text phrases.

Books can be filtered by the different genres and authors.

The theme setting is enabled, allowing the user to toggle between dark and light modes.

JS scripts.js

```
45     const element = document.createElement('div');
46     element.classList.add('preview');
```

SOU... ✓ ↺ ...

JS scripts.js M X index.html # styles.css JS data.js

🔄 □ ...

Message (Ctrl+Ente...)

JS scripts.js > ...

```

44 | const createPreview = ({ author, id, image, title }) => { // created a createPreview function
45 |   const element = document.createElement('button');
46 |   element.classList = 'preview';
47 |   element.setAttribute('data-preview', id);
48 |   element.innerHTML = /* html */ `
49 |     <div>
50 |       <image class='preview__image' src = "${image}" alt="book pic"></image>
51 |     </div>
52 |     <div class='preview__info'>
53 |       <dt class = "preview__title">${title}</dt>
54 |       <dt class = "preview__author"> by ${authors[author]}</dt>
55 |     `;
56 |   return element;
57 | };
58 |
59 | const createPreviewsFragment = (books, startIndex, endIndex) => { // created a createPreviewFragment function
60 |   const fragment = document.createDocumentFragment();//
61 |
62 |   const extracted = books.slice(startIndex, endIndex)//
63 |
64 |   for (let i = 0; i < extracted.length; i++) {
65 |     const { author, id, image, title } = extracted[i];
66 |     const preview = createPreview({ author, id, image, title});
67 |
68 |     fragment.appendChild(preview);
69 |   }
70 |   return fragment;
71 | };
72 |
73 | dataListsItems.appendChild(createPreviewsFragment(matches, 0, BOOKS_PER_PAGE));
74 | dataListsItems.style.display = "grid";
75 |
76 | const genresFragment = document.createDocumentFragment();
77 | const genresOptionAll = document.createElement('option');
78 | genresOptionAll.value = 'any';
79 | genresOptionAll.innerText = 'All Genres';
80 | genresFragment.appendChild(genresOptionAll);
81 | for (const [id, name] of Object.entries(genres)) {
82 |   const option = document.createElement('option');
83 |   const value = id;
84 |   const text = name;
85 |   option.value = value;
86 |   option.innerText = text;
87 |   genresFragment.appendChild(option);
88 | }

```

```

1 | <div>
2 |   <div class="book__cover">
3 |     <img alt="book cover" class="book__cover__img" />
4 |   </div>
5 |   <div class="book__info">
6 |     <div class="book__info__title">
7 |       <h3>${title}</h3>
8 |     </div>
9 |     <div class="book__info__author">
10 |      by ${author}
11 |    </div>
12 |  </div>
13 | </div>
14 |
15 | <div class="book__preview">
16 |   <div class="book__preview__img">
17 |     <img alt="book preview" class="book__preview__img" />
18 |   </div>
19 |   <div class="book__preview__info">
20 |     <div class="book__preview__info__title">
21 |       <h4>${title}</h4>
22 |     </div>
23 |     <div class="book__preview__info__author">
24 |      by ${author}
25 |    </div>
26 |   </div>
27 | </div>
28 |
29 | <div class="book__genres">
30 |   <div class="book__genres__all">
31 |     <button>All Genres</button>
32 |   </div>
33 |   <div class="book__genres__list">
34 |     <div class="book__genres__list__item">
35 |       <button>Fiction</button>
36 |     </div>
37 |     <div class="book__genres__list__item">
38 |       <button>Non-Fiction</button>
39 |     </div>
40 |     <div class="book__genres__list__item">
41 |       <button>Mystery</button>
42 |     </div>
43 |     <div class="book__genres__list__item">
44 |       <button>Science Fiction</button>
45 |     </div>
46 |     <div class="book__genres__list__item">
47 |       <button>Fantasy</button>
48 |     </div>
49 |     <div class="book__genres__list__item">
50 |       <button>Horror</button>
51 |     </div>
52 |     <div class="book__genres__list__item">
53 |       <button>Romance</button>
54 |     </div>
55 |     <div class="book__genres__list__item">
56 |       <button>Drama</button>
57 |     </div>
58 |     <div class="book__genres__list__item">
59 |       <button>Comedy</button>
60 |     </div>
61 |     <div class="book__genres__list__item">
62 |       <button>Action</button>
63 |     </div>
64 |     <div class="book__genres__list__item">
65 |       <button>Thriller</button>
66 |     </div>
67 |     <div class="book__genres__list__item">
68 |       <button>Adventure</button>
69 |     </div>
70 |     <div class="book__genres__list__item">
71 |       <button>Mystery</button>
72 |     </div>
73 |     <div class="book__genres__list__item">
74 |       <button>Science Fiction</button>
75 |     </div>
76 |     <div class="book__genres__list__item">
77 |       <button>Fantasy</button>
78 |     </div>
79 |     <div class="book__genres__list__item">
80 |       <button>Horror</button>
81 |     </div>
82 |     <div class="book__genres__list__item">
83 |       <button>Romance</button>
84 |     </div>
85 |     <div class="book__genres__list__item">
86 |       <button>Drama</button>
87 |     </div>
88 |     <div class="book__genres__list__item">
89 |       <button>Comedy</button>
90 |     </div>
91 |     <div class="book__genres__list__item">
92 |       <button>Action</button>
93 |     </div>
94 |     <div class="book__genres__list__item">
95 |       <button>Thriller</button>
96 |     </div>
97 |     <div class="book__genres__list__item">
98 |       <button>Adventure</button>
99 |     </div>
100 |   </div>
101 | </div>

```