

# REFILWE NQATYELWA

**IWA\_19 Capstone**

# ABOUT IWA\_19 CAPSTONE

**I was required 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**

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

```
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 X | ...
```

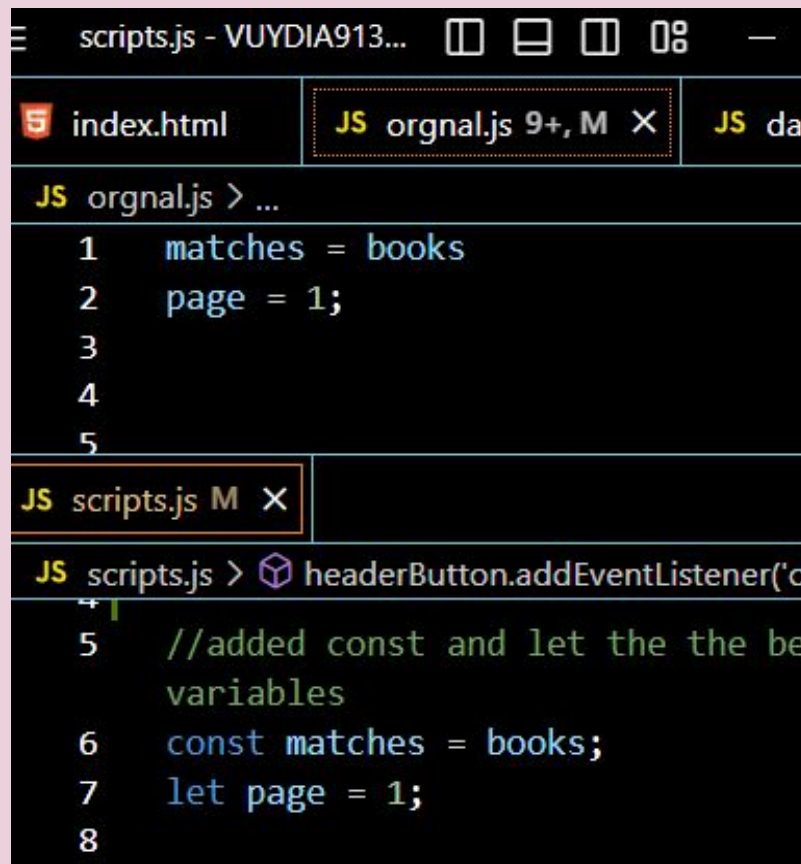
```
JS orgnal.js > ...
1 matches = books
2 page = 1;
3
4
```


## 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.

## 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 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
```

# 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.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 orignal.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     fragment.appendChild(preview)
36   }
37
38   data-list-items.appendChild
      (fragment)
39
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.



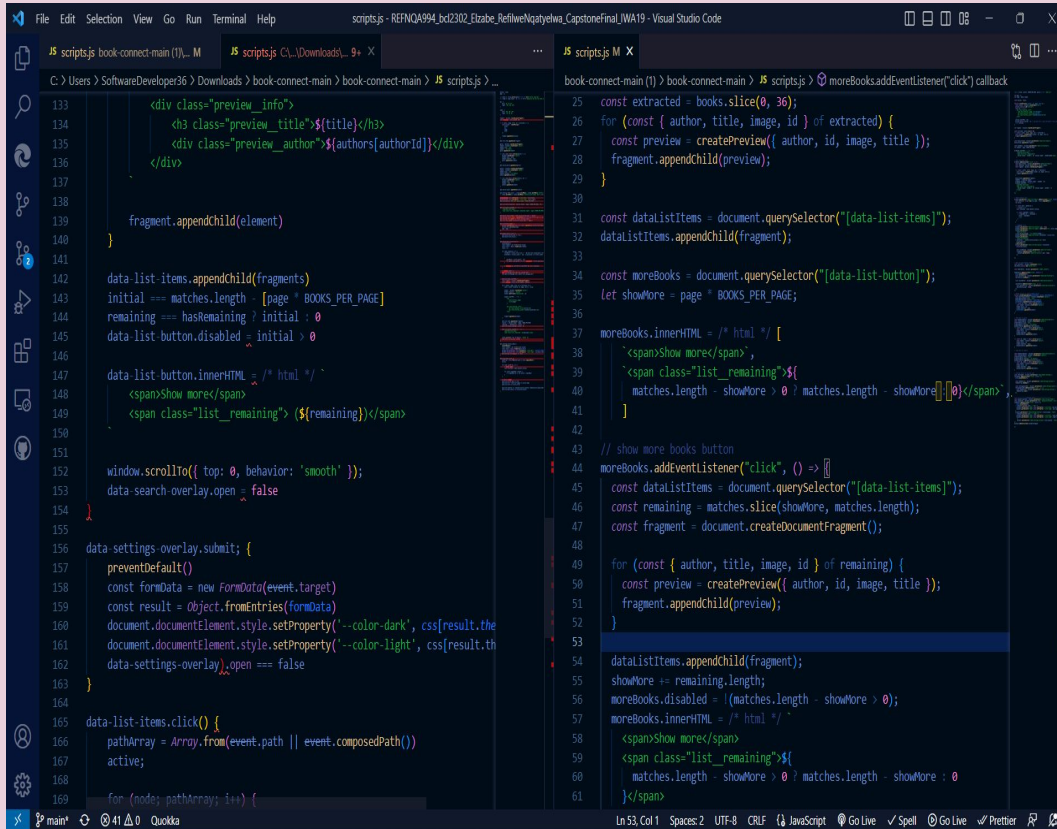
# AUTHORS

- 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 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'.

```
JS original.js > ...
66
67 authors = document.
   createDocumentFragment()
68 element = document.createElement
   ('option')
69 element.value = 'any'
70 element.innerText = 'All Authors'
71 authors.appendChild(element)
72
73
74 for ([id, name]; Object.entries
   (authors); id++) {
75   document.createElement
   ('option')
76   element.value = value
77   element = text
78   authors.appendChild(element)
79 }
80
81 data-search-authors.appendChild
   (authors)
82

JS scripts.js > ...
252
253 // Drop down for authors
254 const dataSearchAuthors = document.
   querySelector("[data-search-authors]");
255 const allAuthorsOption = document.
   createElement("option");
256 allAuthorsOption.value = "any";
257 allAuthorsOption.innerText = "All
   Authors";
258 dataSearchAuthors.appendChild
   (allAuthorsOption);
259 > for (const [id, names] of Object.
   entries(authors)) { ...
   }
264
265 });
266
267
268
269
270
271
272
```

# SHOW MORE BOOKS



```
book-connect-main.js
133 <div class="preview_info">
134   <h3 class="preview_title">${title}</h3>
135   <div class="preview_author">${authors[authorId]}</div>
136 </div>
137
138
139 fragment.appendChild(element)
140
141
142
143 data-list-items.appendChild(fragment)
144 initial == matches.length - [page * BOOKS_PER_PAGE]
145 remaining == hasRemaining ? initial : 0
146 data-list-button.disabled = initial > 0
147
148 data-list-button.innerHTML = /* html */ `
149   <span>show more</span>
150   <span class="list_remaining"> ${remaining}</span>
151 `
152
153 window.scrollTo({ top: 0, behavior: 'smooth' });
154 data-search-overlay.open = false
155
156
157 data-settings-overlay.submit; {
158   preventDefault()
159   const formData = new FormData(event.target)
160   const result = Object.fromEntries(formData)
161   document.documentElement.style.setProperty('--color-dark', css[result.the])
162   document.documentElement.style.setProperty('--color-light', css[result.th])
163   data-settings-overlay.open == false
164 }
165
166 data-list-items.click() {
167   pathArray = Array.from(event.path || event.composedPath())
168   active;
169   for (nodes; pathArray; i++) {
```

```
moreBooks.js
25 const extracted = books.slice(0, 36);
26 for (const { author, title, image, id } of extracted) {
27   const preview = createPreview({ author, id, image, title });
28   fragment.appendChild(preview);
29 }
30
31 const dataListItems = document.querySelector("[data-list-items]");
32 dataListItems.appendChild(fragment);
33
34 const moreBooks = document.querySelector("[data-list-button]");
35 let showMore = page * BOOKS_PER_PAGE;
36
37 moreBooks.innerHTML = /* html */ [
38   '<span>show more</span>',
39   '<span class="list_remaining">${
40     matches.length - showMore > 0 ? matches.length - showMore : 0
41   }</span>',
42 ]
43
44 // show more books button
45 moreBooks.addEventListener("click", () => {
46   const dataListItems = document.querySelector("[data-list-items]");
47   const remaining = matches.slice(showMore, matches.length);
48   const fragment = document.createDocumentFragment();
49
50   for (const { author, title, image, id } of remaining) {
51     const preview = createPreview({ author, id, image, title });
52     fragment.appendChild(preview);
53   }
54
55   dataListItems.appendChild(fragment);
56   showMore += remaining.length;
57   moreBooks.disabled = (matches.length - showMore > 0);
58   moreBooks.innerHTML = /* html */ `
59     <span>show more</span>
60     <span class="list_remaining">${
61       matches.length - showMore > 0 ? matches.length - showMore : 0
62     }</span>
63 `
```

- 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 - REFQA94_bcd302_Elabe_RefineVagayewa_CapstoneFinal_JWA19 - Visual Studio Code

JS scripts.js book-connect-main (1)... M JS scripts.js C:\Downloads\... 9+ X JS scripts.js M X

C:\Users\SoftwareDeveloper36\Downloads\book-connect-main\book-connect-main> JS scripts.js 0 filters

71
72 data-list-button.innerHTML = `<html> / [
73   '<span>show more</span>',
74   '<span class="list_remaining"> (${matches.length - [page * BOOKS_PER_PAGE
75 ]
76
77 data-search-cancel.click() { data-search-overlay.open === false }
78 data-settings-cancel.click() { querySelect(data-settings-overlay).open === fa
79 data-settings-form.submit() { actions.settings.submit }
80 data-list.close.click() { data-list-active.open === false }
81
82 data-list-button.click() {
83   document.querySelector([data-list-items]).appendChild(createPreviewFrags
84   actions.list.updateRemaining()
85   page = page + 1
86 }
87
88 data-header-search.click() {
89   data-search-overlay.open === true ;
90   data-search-title.focus();
91 }
92
93 data-search-form.click(filters) {
94   preventDefault()
95   const formData = new FormData(event.target)
96   const filters = object.fromEntries(formData)
97   result = []
98
99   for (book; booksList; i++) {
100     titleMatch = filters.title.trim() == '' && book.title.toLowerCase().in
101     authorMatch = filters.author == 'any' || book.author === filters.autho
102
103     {
104       genreMatch = filters.genre == 'any'
105       for (genre; book.genres; i++) { if singleGenre = filters.genre }
106     }
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

- 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 author.
- 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

```
1 matches = books
2 page = 1;
3
4 if (books && !Array.isArray(books)) throw new Error('Source required')
5 if (range && range.length < 2) throw new Error('Range must be an array')
6
7 day = {
8   dark: '10, 10, 20',
9   light: '255, 255, 255',
10 }
11
12 night = {
13   dark: '255, 255, 255',
14   light: '10, 10, 20',
15 }
16
17 fragment = document.createDocumentFragment()
18 const extracted = books.slice(0, 36)
19
20 for ([author, image, title, id]; extracted; i++) {
21   const preview = createPreview({
22     author,
23     id,
24     image,
25     title
26   })
27   fragment.appendChild(preview)
28 }
29
30 dataListItems.appendChild(fragment)
31
32 genres = document.createDocumentFragment()
33 element = document.createElement('option')
34 element.value = 'any'
35 element = 'All Genres'
36
37 genres.appendChild(element)
```

```
183 settingsCancelBtn.addEventListener('click', () => {
184   const themeOverlay = document.querySelector('[data-settings-theme]')
185   themeOverlay.close();
186 });
187
188 // Enabling the Save Button to change the background
189
190 const css = {
191   day: {
192     dark: '10, 10, 20',
193     light: '255, 255, 255',
194   },
195   night: {
196     dark: '255, 255, 255',
197     light: '10, 10, 20',
198   }
199 }
200
201 const form = document.getElementById('settings');
202 const themeSelect = document.querySelector('[data-settings-theme]')
203
204 form.addEventListener('submit', (event) => {
205   event.preventDefault();
206   const theme = themeSelect.value;
207
208   document.documentElement.style.setProperty('--color-dark', css[theme].dark);
209   document.documentElement.style.setProperty('--color-light', css[theme].light);
210 });
211
212 // Initialize theme based on user's OS theme preference
213 const prefersDarkMode = window.matchMedia('(prefers-color-scheme: dark)');
214 const initialTheme = prefersDarkMode ? 'night' : 'day';
215 document.documentElement.style.setProperty('--color-dark', css[initialTheme].dark);
216 document.documentElement.style.setProperty('--color-light', css[initialTheme].light);
217
218
219
```

# 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.