

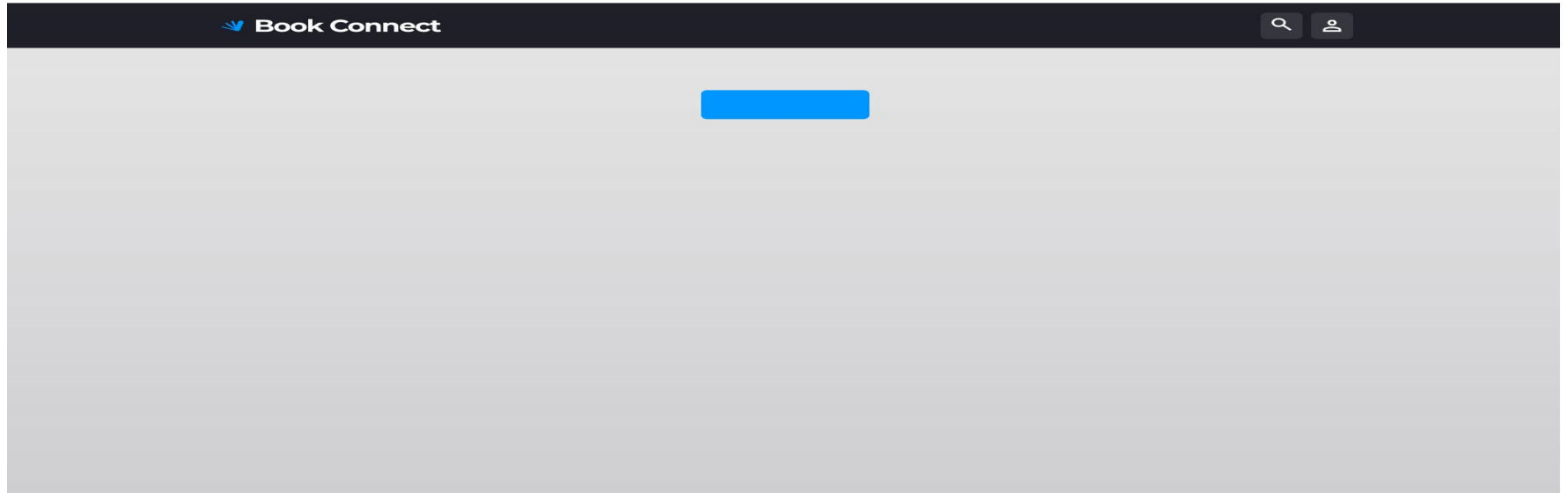


# BOOK CONNECT

By Rorisang Makgaka



We are given a code for Book Connect store, it is not working, current display....



# What code must display.....

 Book Connect



**Journeys in English**

Bill Bryson



**Allegiant**

Veronica Roth



**Life Without Principle**

Henry David Thoreau



**The Catcher in the Rye**

J.D. Salinger

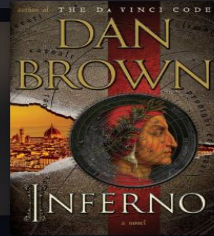


**Robin**

Frances Hodgson Burnett



**Harry Potter and the Sorcerer's Stone**



**Inferno**

Dan Brown (2013)

Harvard professor of symbology Robert Langdon awakens in an Italian hospital, disoriented and with no recollection of the past thirty-six hours, including the origin of the macabre object hidden in his belongings. With a relentless female assassin trailing them through Florence, he and his resourceful doctor, Sienna Brooks, are forced to flee....

Close



Oscar Wilde

**Tablet of Fire**

**Journal of Ordeal, 1850-1859**

**Notes**



**The Picture of Dorian Gray: The Definitive Edition**

Code supplied for this project has the following issues

- Syntax errors
- Logical errors

Code given...

```
scripts.js book-connect-main  JS scripts.js C:\...\Downloads\... 9+ X  <> index.html
> Users > SoftwareDeveloper21 > Downloads > book-connect-main > book-connect-main > JS scripts.js > ...
1  matches = books
2  page = 1;
3
```

Code fixed

- Imported variables for data.js file and declared variables using let

```
1  import { BOOKS_PER_PAGE, authors, genres, books } from "./data.js";
2
3  let matches = books;
4  let page = 1;
5
```

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

```
const day = { //added const  
  dark: '10, 10, 20',  
  light: '255, 255, 255',  
}  
  
const night = { //added const  
  dark: '255, 255, 255',  
  light: '10, 10, 20',  
}
```

# Code given....

```
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)

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

# Fixed code for display..

```
let startIndex = 0
let endIndex = 36
const extracted = books.slice(startIndex, endIndex) //source.slice(range[0], range[1]) // commented out source and replaced it w
console.log(books[0])
const add = document.querySelector('[data-list-items]') // created a variable and took the empty div from HTML line 93

const fragment = document.createDocumentFragment()
for (const {author, image, title, id, description, published} of extracted) { // added s on author, turned it into a for loop
  //const { author: authorId, id, image, title } = props

  let element = document.createElement('button')
  element.classList = 'preview'
  element.dataset.id = id
  element.dataset.title = title
  element.dataset.description = description
  element.dataset.image = image
  element.dataset.subtitle = (`${authors[author]} (${(new Date(published)).getFullYear()})`)
  element.setAttribute('data-preview', id)
```



```
    element.innerHTML = /* html */ `
        <div></div>

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

    fragment.appendChild(element)
}

add.appendChild(fragment) // appended the fragment into add
document.querySelector('[data-list-items]').appendChild(fragment);
```

# Show more button

```
showMoreButton.addEventListener('click', () => {  
  startIndex += 36  
  endIndex += 36  
  const startIndex1 = startIndex  
  const endIndex1 = endIndex  
  const extracted = books.slice(startIndex1, endIndex1)  
  const fragment = document.createDocumentFragment()
```

```

for ( const {author, image, title, id, description, published} of extracted) { // added s on author, turned
  //const { author: authorId, id, image, title } = props

  let element = document.createElement('button')
  element.classList = 'preview'
  element.dataset.id = id
  element.dataset.title = title
  element.dataset.description = description
  element.dataset.image = image
  element.dataset.subtitle = (`${authors[author]} (${(new Date(published)).getFullYear()})`)
  element.setAttribute('data-preview', id)

  element.innerHTML = /* html */ `
    <div></div>

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

  fragment.appendChild(element)
}
add.appendChild(fragment) // appended the fragment into add
document.querySelector('[data-list-items]').appendChild(fragment);
})

```

## Preview button, fixed code..

```
const details = (event) =>{  
  const overlay1 = document.querySelector('[data-list-active]')  
  const title = document.querySelector('[data-list-title]')  
  const subtitle = document.querySelector('[data-list-subtitle]')  
  const description = document.querySelector('[data-list-description]')  
  const image1 = document.querySelector('[data-list-image]')  
  const imageblur = document.querySelector('[data-list-blur]')
```

## Search button code

```
6
7  const genresFragment = document.createDocumentFragment();
8  let element = document.createElement('option');
9  element.value = 'any';
10 element.innerText = 'All Genres';
11 genresFragment.appendChild(element);
12
13 for (let [id, name] of Object.entries(genres)) {
14   const element = document.createElement('option');
15   const value = id;
16   const text = name;
17   element.value = value;
18   element.innerText = text;
19   genresFragment.appendChild(element);
20 }
```

```
1 document.querySelector('[data-search-genres]').appendChild(genresFragment);  
2  
3 const authorsFragment = document.createDocumentFragment();  
4 element = document.createElement('option');  
5 element.value = 'any';  
6 element.innerText = 'All Authors';  
7 authorsFragment.appendChild(element);  
8 document.querySelector('[data-search-authors]').appendChild(authorsFragment);
```

# Show more button

```
5 //show more
6 const dataListButton = document.querySelector('[data-list-button]');
7 dataListButton.innerHTML = /* html */ [
8   '<span>Show more</span>',
9   '<span class="list__remaining">${matches.length - page * BOOKS_PER_PAGE > 0 ? matches.length - page * BOOKS_PER_PAGE : 0}</span>',
10 ].join('');
11
12 dataListButton.disabled = !(matches.length - page * BOOKS_PER_PAGE > 0);
13
14 //data-search-cancel.click() { data-search-overlay.open === false }
15 const dataSearchCancel = document.querySelector('[data-search-cancel]');
16 dataSearchCancel.addEventListener('click', function() {
17   const overlay = document.querySelector('[data-search-overlay]');
18   overlay.show();
19   if (overlay.open === false) {
20     // do something here
21     overlay.close();
22   }
23 });
24
```

```
1 document.querySelector('[data-search-genres]').appendChild(genresFragment);  
2  
3 const authorsFragment = document.createDocumentFragment();  
4 element = document.createElement('option');  
5 element.value = 'any';  
6 element.innerText = 'All Authors';  
7 authorsFragment.appendChild(element);  
8 document.querySelector('[data-search-authors]').appendChild(authorsFragment);
```



```

62
63 data-settings-theme.value === window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' : 'day'
64 v = window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches ? 'night' | 'day'
65
66 documentElement.style.setProperty('--color-dark', css[v].dark);
67 documentElement.style.setProperty('--color-light', css[v].light);
68 data-list-button = "Show more (books.length - BOOKS_PER_PAGE)"
69

```

```

155 //Dark mode and light mode
156 const settingOverlay = document.querySelector('[data-header-settings]')
157 settingOverlay.addEventListener('click', (event)=>{
158     document.querySelector('[data-settings-overlay]').style.display = 'block'
159 })
160 const dataSettingsTheme = document.querySelector('[data-settings-theme]')
161 const saveButton = document.querySelector("body > dialog:nth-child(5) > div > div > button.overlay__button.overlay__button_primary")
162 Complexity is 3 Everything is cool!
163 saveButton.addEventListener('click', (event) =>{
164     event.preventDefault()
165     if (dataSettingsTheme.value === 'day') {
166         document.querySelector('body').style.setProperty('--color-dark', day.dark)
167         document.querySelector('body').style.setProperty('--color-light', day.light)
168         document.querySelector('[data-settings-overlay]').style.display = 'none'
169     }
170     if (dataSettingsTheme.value === 'night') {
171         document.querySelector('body').style.setProperty('--color-dark', night.dark)
172         document.querySelector('body').style.setProperty('--color-light', night.light)
173         document.querySelector('[data-settings-overlay]').style.display = 'none'
174     }
175 });

```

# Overlay code

```
const overlay = document.querySelector('#data-settings-overlay');
if (overlay && overlay.open) {
  overlay.close();
}
const form = document.querySelector('#data-settings-form');
//form.addEventListener('submit', submitFormData);


//data-list-close.click() { data-list-active.open === false }
const dataListClose = document.getElementById('data-list-close');
const dataListActive = document.getElementById('data-list-active');
```

# Code for searching books

```
function dataSearchFormClick(event) { // add function keyword and function name
  event.preventDefault(); // add event argument and fix preventDefault method call
  const formData = new FormData(event.target);
  const filters = Object.fromEntries(formData);
  const booksList = []; // declare booksList array
  let result = [];

  // fixed for loop
  for (let i=0; i<booksList.length; i++) {
    const book = booksList[i]; // declare book variable
    const titleMatch = filters.title.trim() === '' || book.title.toLowerCase().includes(filters.title.toLowerCase()); // use comparison operator
    const authorMatch = filters.author === 'any' || book.author === filters.author;
    let genreMatch = true; // declare genreMatch and set it to true
    if (filters.genre !== 'any') { // if filters.genre is not 'any', check genreMatch
      genreMatch = false; // set genreMatch to false by default
      for (let j=0; j<book.genres.length; j++) { // use correct for loop syntax and fix book.genres property
        if (book.genres[j] === filters.genre) { // if singleGenre is equal to filters.genre, set genreMatch to true
          genreMatch = true;
          break; // exit loop early
        }
      }
    }
    if (titleMatch && authorMatch && genreMatch) { // check if all three conditions are true
      result.push(book);
    }
  }
  return result; // return result array
}
```

Complexity is 12 You must be kidding

```
function dataSearchFormClick(event) { // add function keyword and function name   
  event.preventDefault(); // add event argument and fix preventDefault method call  
  const formData = new FormData(event.target);  
  const filters = Object.fromEntries(formData);  
  const booksList = []; // declare booksList array  
  let result = [];
```

```
// fixed for loop
for (let i=0; i<booksList.length; i++) {
  const book = booksList[i]; // declare book variable
  const titleMatch = filters.title.trim() === '' || book.title.toLowerCase().includes(filters.title.toLowerCase()); // use
  const authorMatch = filters.author === 'any' || book.author === filters.author;
  let genreMatch = true; // declare genreMatch and set it to true
  if (filters.genre !== 'any') { // if filters.genre is not 'any', check genreMatch
    genreMatch = false; // set genreMatch to false by default
    for (let j=0; j<book.genres.length; j++) { // use correct for loop syntax and fix book.genres property
      if (book.genres[j] === filters.genre) { // if singleGenre is equal to filters.genre, set genreMatch to true
        genreMatch = true;
        break; // exit loop early
      }
    }
  }
  if (titleMatch && authorMatch && genreMatch) { // check if all three conditions are true
    result.push(book);
  }
}
return result; // return result array
}
```

Thank you!!!!!!!

**QUESTIONS?**