SHUHAIR SMITH

JAVASCRIPT IWA19

FINAL CAPSTONE PROJECT

WHAT IS REQUIRED

> The user wants to view a list of book previews, by title and author,

> The user wants an image associated with all book previews,

> The user wants to have the option of reading a summary of the book,

> The user wants to have the option of seeing the date that a book was published

> The user wants to find books based on specific text phrases

> The user wants to filter books by author

> The user wants to filter books by genre

> The User want to toggle between dark and light modes

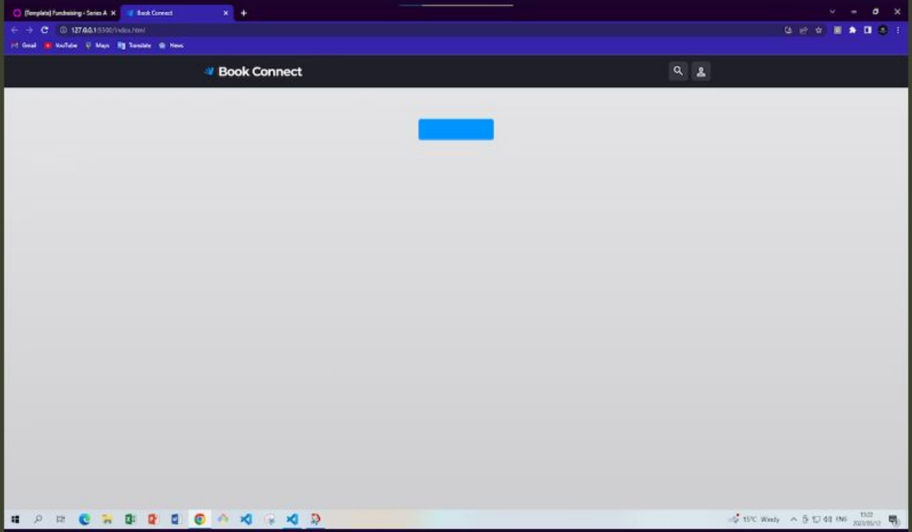
the project (before)

.The title only showed the name and no picture

. The search and the picture next to it

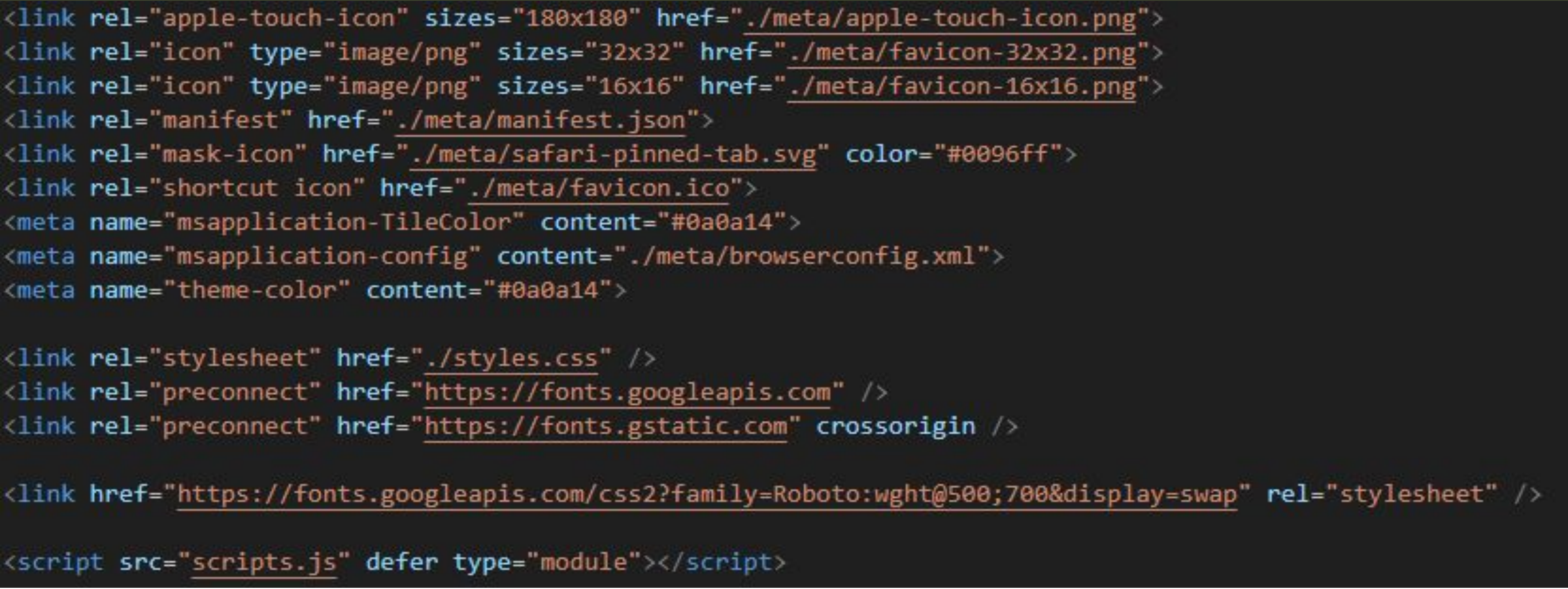
.It had the book connect and the little picture next to it

. And the blue empty button



INDEX HTML

- I ADDED the links, to include the files,pictures that are linked into the html documents
- I have also removed data.js link and used the export and import method instate.



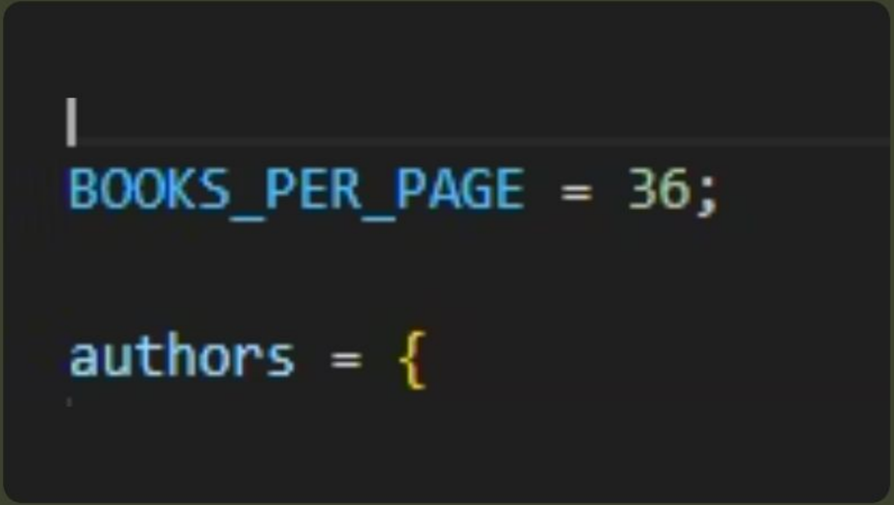
```
<link rel="apple-touch-icon" sizes="180x180" href="./meta/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="./meta/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="./meta/favicon-16x16.png">
<link rel="manifest" href="./meta/manifest.json">
<link rel="mask-icon" href="./meta/safari-pinned-tab.svg" color="#0096ff">
<link rel="shortcut icon" href="./meta/favicon.ico">
<meta name="msapplication-TileColor" content="#0a0a14">
<meta name="msapplication-config" content="./meta/browserconfig.xml">
<meta name="theme-color" content="#0a0a14">

<link rel="stylesheet" href="./styles.css" />
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@500;700&display=swap" rel="stylesheet" />

<script src="scripts.js" defer type="module"></script>
```

DATA.JS CHANGES (BEFORE & AFTER)

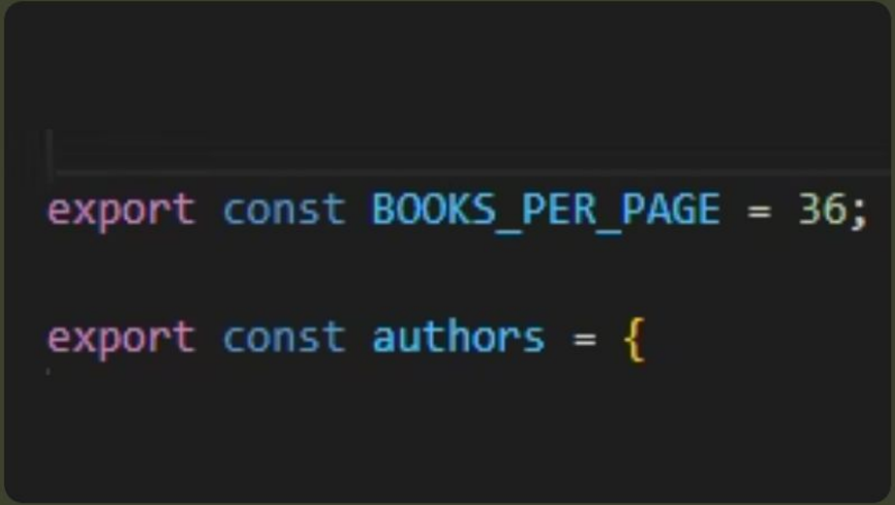
>Exported from data.js ,imported in script.js so that I can be able to use the elements in the script.js.

>I Created variables and definedthem

BEFORE CHANGES	AFTER CHANGES



```
BOOKS_PER_PAGE = 36;

authors = {
```



```
export const BOOKS_PER_PAGE = 36;

export const authors = {
```

This code is for the search respond function.

>created variables and defined them.

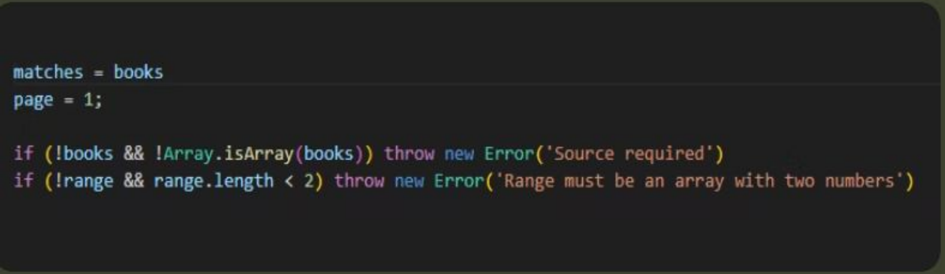
>I’ve replaced && with ||(double vertical bars) to check if the variable books is false or if it's not an array.

>if 'books' variable exists and is an array, to throw an error if not.

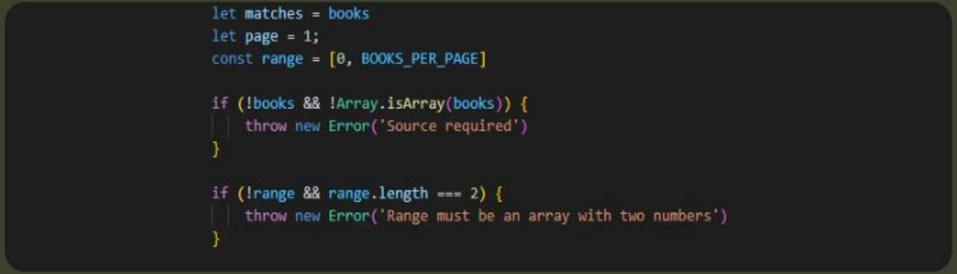
> If ‘book’ & 'range' variable exists and has at least 2 elements, to throw an error if not.

BEFORE CHANGES

AFTER CHANGES



```
matches = books
page = 1;

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')
```



```
let matches = books
let page = 1;
const range = [0, BOOKS_PER_PAGE]

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')
}
```

Theme (dark and light mode)

> created cons variables and defined two objects, day and night, each with two properties: dark and light

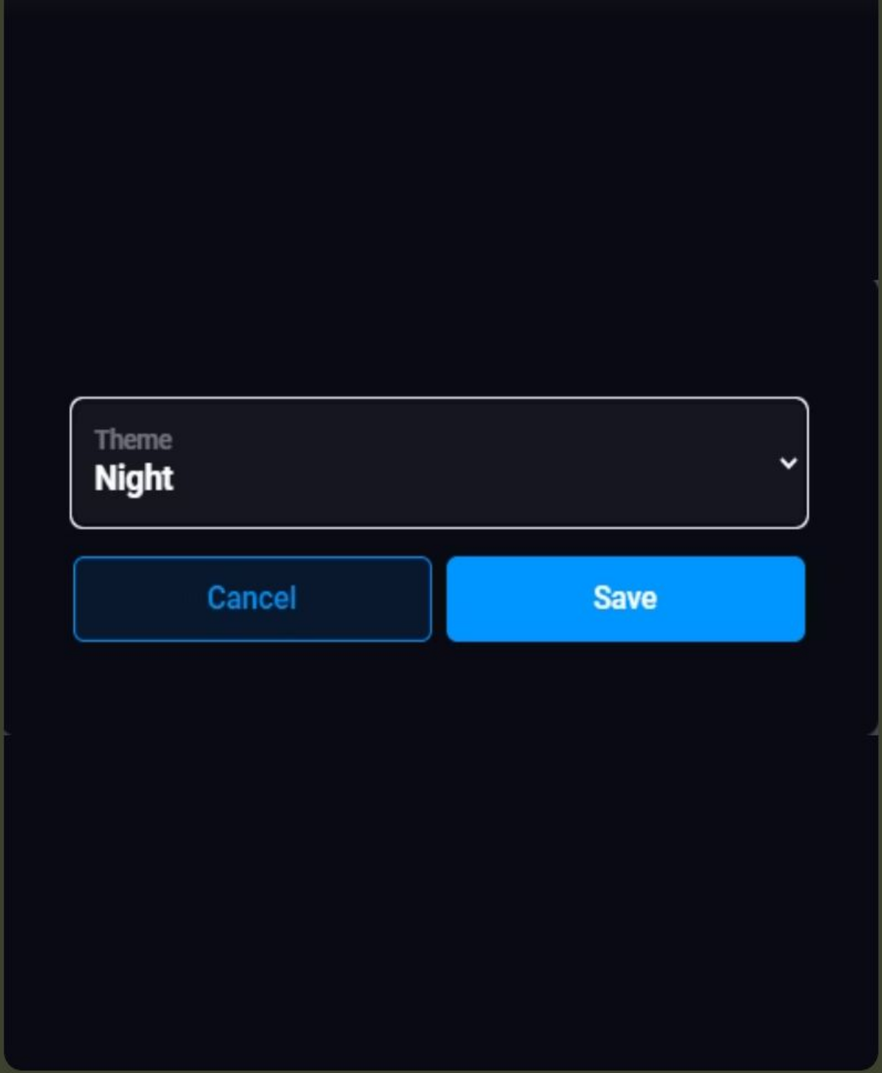
> I have selected elements from the DOM using the query selector

> I have used the event listener to prevent the default behavior of the save button (i.e., submitting a form)

>set the dark and light colors for the body element

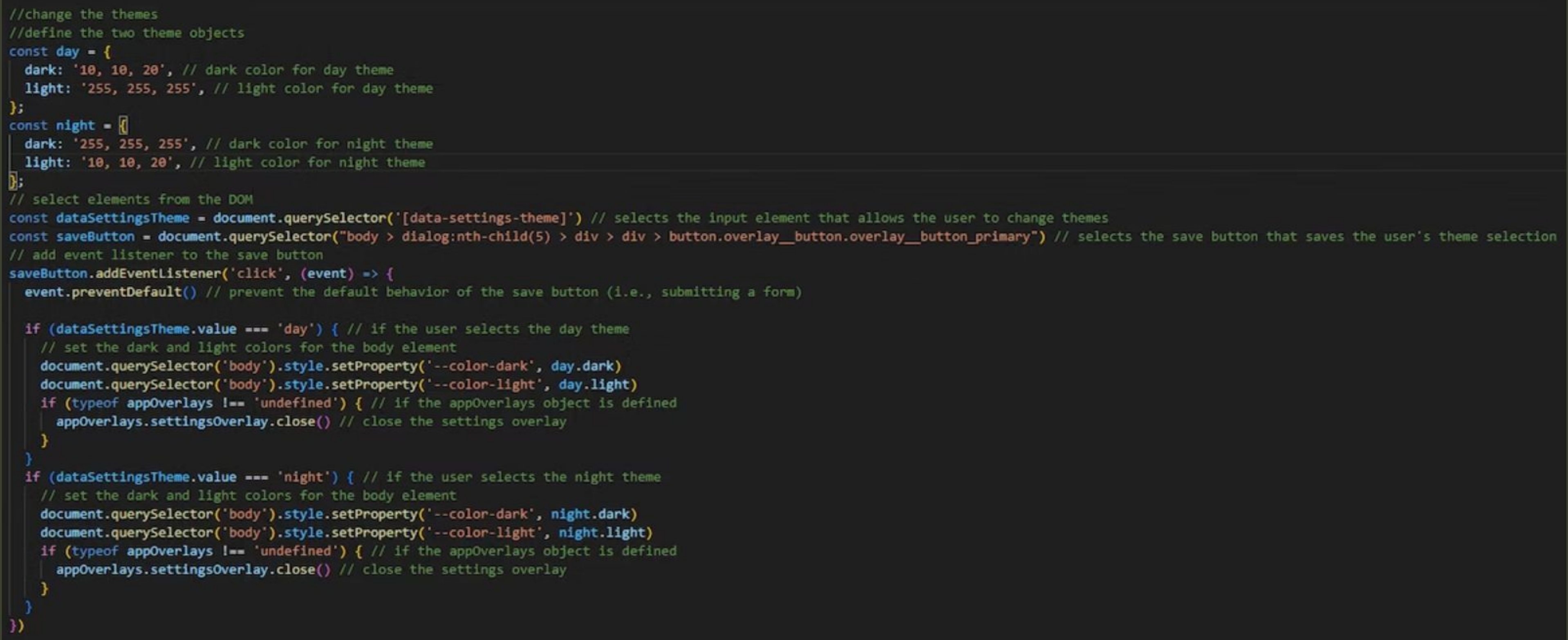
>If the the user selects the day theme

>if the user select the night theme



Theme
Night

Script for themes



```
//change the themes
//define the two theme objects
const day = {
  dark: '10, 10, 20', // dark color for day theme
  light: '255, 255, 255', // light color for day theme
};
const night = {
  dark: '255, 255, 255', // dark color for night theme
  light: '10, 10, 20', // light color for night theme
};
// select elements from the DOM
const dataSettingsTheme = document.querySelector('[data-settings-theme]') // selects the input element that allows the user to change themes
const saveButton = document.querySelector("body > dialog:nth-child(5) > div > div > button.overlay__button.overlay__button_primary") // selects the save button that saves the user's theme selection
// add event listener to the save button
saveButton.addEventListener('click', (event) => {
  event.preventDefault() // prevent the default behavior of the save button (i.e., submitting a form)

  if (dataSettingsTheme.value === 'day') { // if the user selects the day theme
    // set the dark and light colors for the body element
    document.querySelector('body').style.setProperty('--color-dark', day.dark)
    document.querySelector('body').style.setProperty('--color-light', day.light)
    if (typeof appOverlays !== 'undefined') { // if the appOverlays object is defined
      appOverlays.settingsOverlay.close() // close the settings overlay
    }
  }
  if (dataSettingsTheme.value === 'night') { // if the user selects the night theme
    // set the dark and light colors for the body element
    document.querySelector('body').style.setProperty('--color-dark', night.dark)
    document.querySelector('body').style.setProperty('--color-light', night.light)
    if (typeof appOverlays !== 'undefined') { // if the appOverlays object is defined
      appOverlays.settingsOverlay.close() // close the settings overlay
    }
  }
})
```

layout for HTML 36 first books preview

>Create a constant variable called "fragment” and assigned it a value

>Set the starting and ending indices (set of books) for a slice of the "books" array Book selection

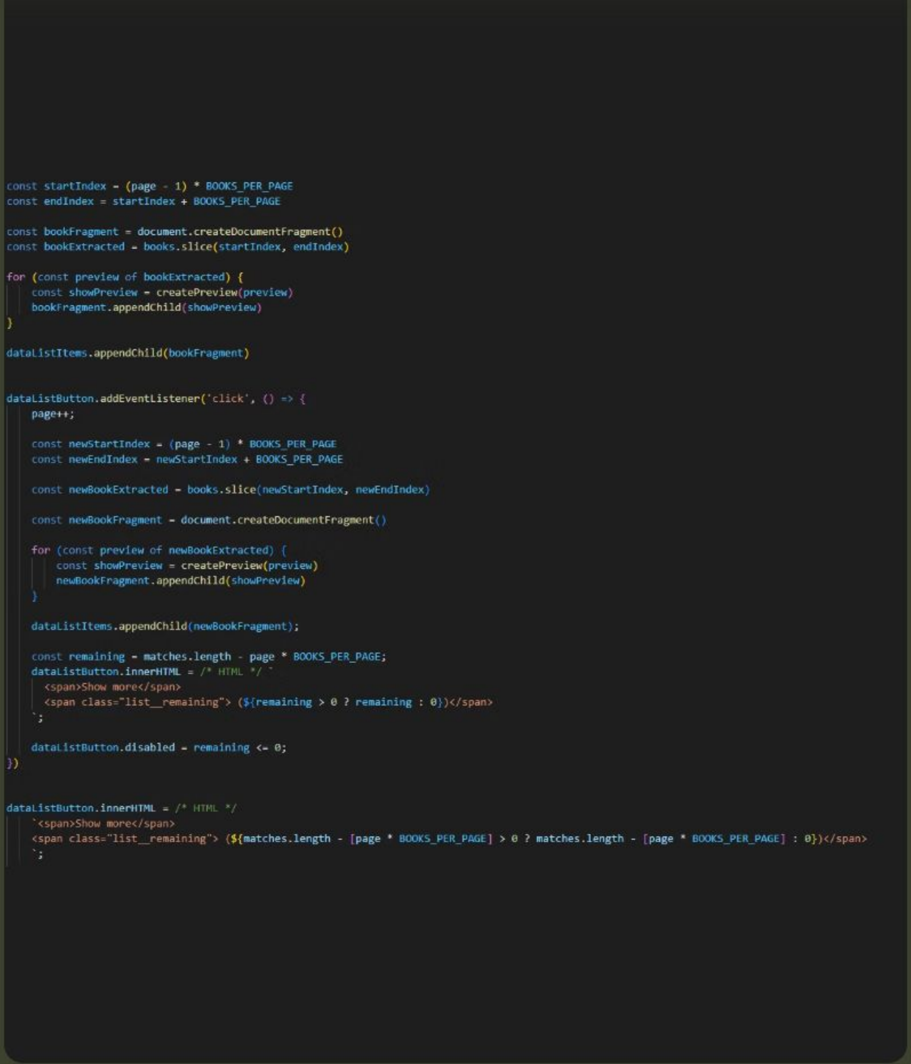
>Loop through each book in the "extracted" array using a"for" loop

>For each book, created a new "dl" element called "preview"and assign it various data attributes based on the book's properties

>Set the "inner HTML" of the "preview" element to a string of HTML code using Template literal(using back ticks)

>Add preview element to the document fragment .This will display all the book previews.

>Extract the slice of books using the "slice" method and assign it to the "extracted" variable.



```javascript
const startIndex = (page - 1) * BOOKS_PER_PAGE
const endIndex = startIndex + BOOKS_PER_PAGE

const bookFragment = document.createDocumentFragment()
const bookExtracted = books.slice(startIndex, endIndex)

for (const preview of bookExtracted) {
    const showPreview = createPreview(preview)
    bookFragment.appendChild(showPreview)
}

dataListItems.appendChild(bookFragment)


dataListButton.addEventListener('click', () => {
    page++;

    const newStartIndex = (page - 1) * BOOKS_PER_PAGE
    const newEndIndex = newStartIndex + BOOKS_PER_PAGE

    const newBookExtracted = books.slice(newStartIndex, newEndIndex)

    const newBookFragment = document.createDocumentFragment()

    for (const preview of newBookExtracted) {
        const showPreview = createPreview(preview)
        newBookFragment.appendChild(showPreview)
    }

    dataListItems.appendChild(newBookFragment);

    const remaining = matches.length - page * BOOKS_PER_PAGE;
    dataListButton.innerHTML = /* HTML */ `
      <span>Show more</span>
      <span class="list__remaining"> (${remaining > 0 ? remaining : 0})</span>
    `;

    dataListButton.disabled = remaining <= 0;
})


dataListButton.innerHTML = /* HTML */
    `<span>Show more</span>
    <span class="list__remaining"> (${matches.length - [page * BOOKS_PER_PAGE] > 0 ? matches.length - [page * BOOKS_PER_PAGE] : 0})</span>
    `;
```

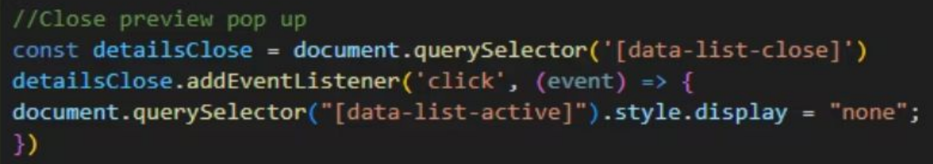
Display book details on preview pop up

>I’ve used the querySelector to select HTML element's settings button HTML element with the data attribute "data-header-settings".

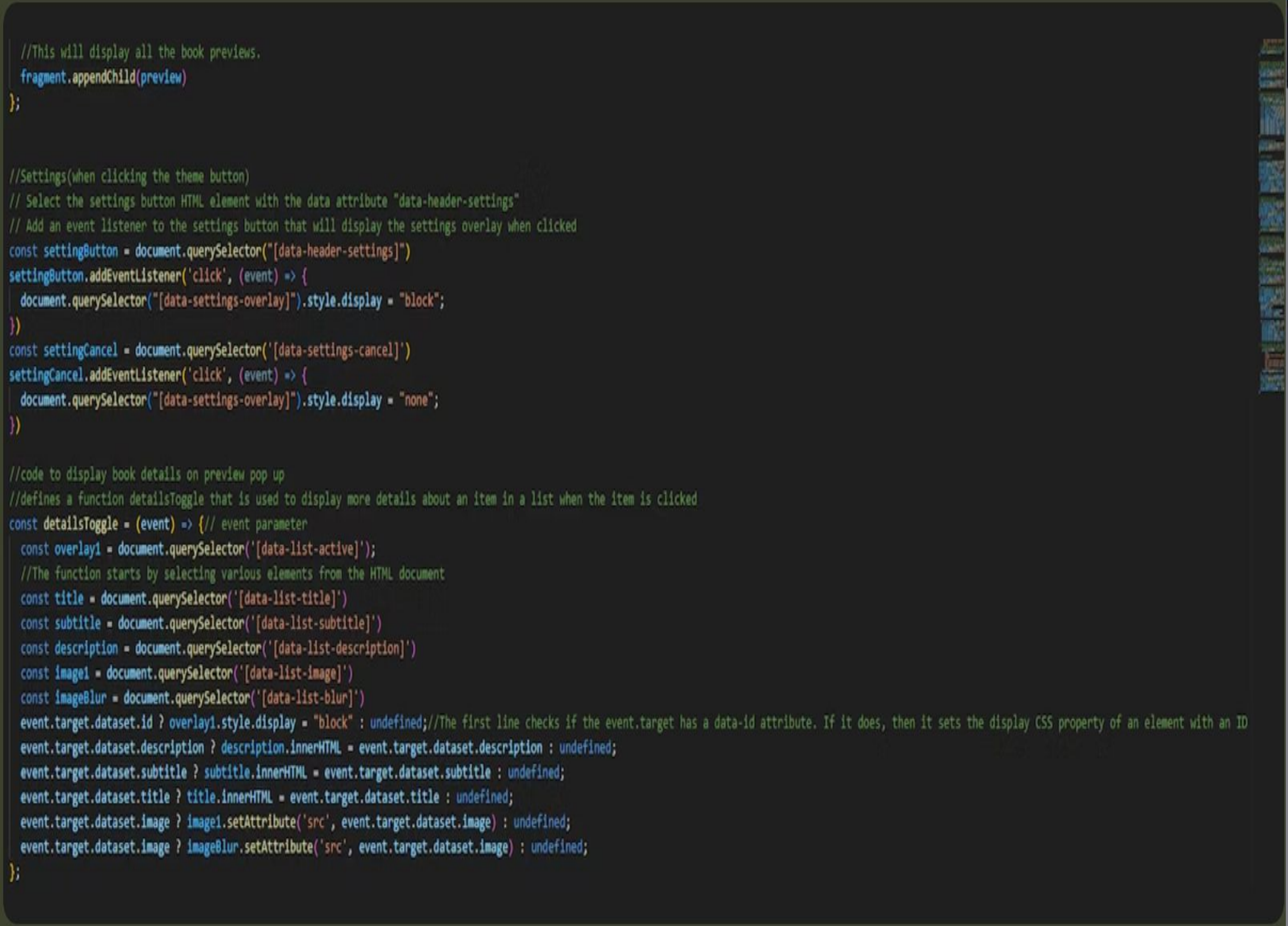
>Added an event listener to the settings button that will display the settings overlay when clicked.

Close preview pop up

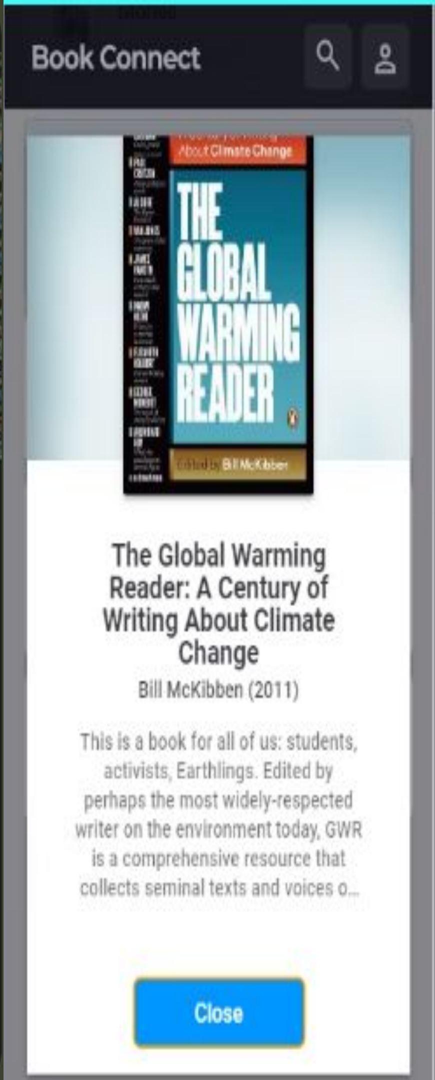
> I’ve use the query selector and the event listener to create the close option.



```
//Close preview pop up
const detailsClose = document.querySelector('[data-list-close]')
detailsClose.addEventListener('click', (event) => {
document.querySelector("[data-list-active]").style.display = "none";
})
```



```
  //This will display all the book previews.
  fragment.appendChild(preview)
};


//Settings(when clicking the theme button)
// Select the settings button HTML element with the data attribute "data-header-settings"
// Add an event listener to the settings button that will display the settings overlay when clicked
const settingButton = document.querySelector("[data-header-settings]")
settingButton.addEventListener('click', (event) => {
  document.querySelector("[data-settings-overlay]").style.display = "block";
})
const settingCancel = document.querySelector('[data-settings-cancel]')
settingCancel.addEventListener('click', (event) => {
  document.querySelector("[data-settings-overlay]").style.display = "none";
})

//code to display book details on preview pop up
//defines a function detailsToggle that is used to display more details about an item in a list when the item is clicked
const detailsToggle = (event) => {// event parameter
  const overlay1 = document.querySelector('[data-list-active]');
  //The function starts by selecting various elements from the HTML document
  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]')
  event.target.dataset.id ? overlay1.style.display = "block" : undefined;//The first line checks if the event.target has a data-id attribute. If it does, then it sets the display CSS property of an element with an ID
  event.target.dataset.description ? description.innerHTML = event.target.dataset.description : undefined;
  event.target.dataset.subtitle ? subtitle.innerHTML = event.target.dataset.subtitle : undefined;
  event.target.dataset.title ? title.innerHTML = event.target.dataset.title : undefined;
  event.target.dataset.image ? image1.setAttribute('src', event.target.dataset.image) : undefined;
  event.target.dataset.image ? imageBlur.setAttribute('src', event.target.dataset.image) : undefined;
};
```



The Global Warming Reader: A Century of Writing About Climate Change

Bill McKibben (2011)

This is a book for all of us: students, activists, Earthlings. Edited by perhaps the most widely-respected writer on the environment today, GWR is a comprehensive resource that collects seminal texts and voices o...

SEARCH OPTION (FUNCTION)

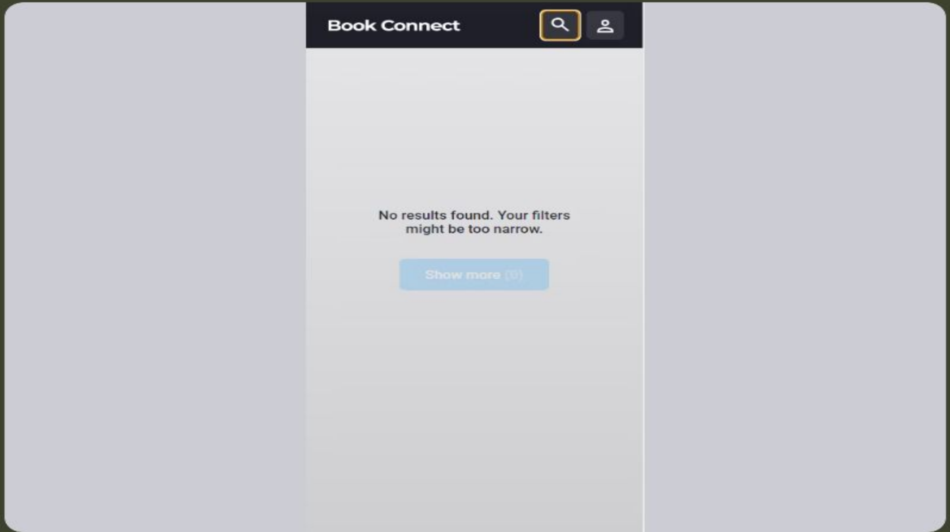
I have created variables and assigned them elements from html, assigned a click event listener to the element with data-list-item attribute.

And when the element is clicked, it triggers the ‘detailsToggle” function.

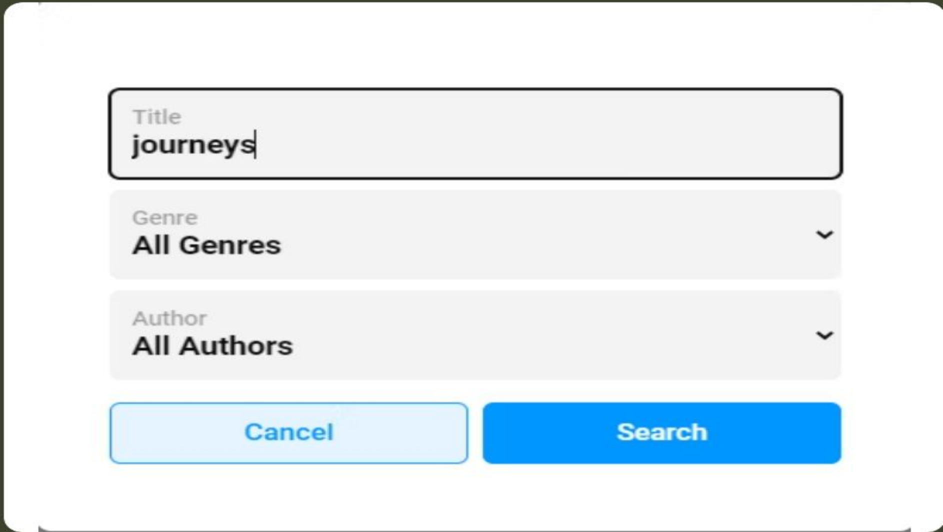
The <u>for loop</u> creates an option element for each author in the

"authors" object, setting the value of each element to the author's ID and the text content to their name.

These option elements are also appended to the "data-search-authors" select element.

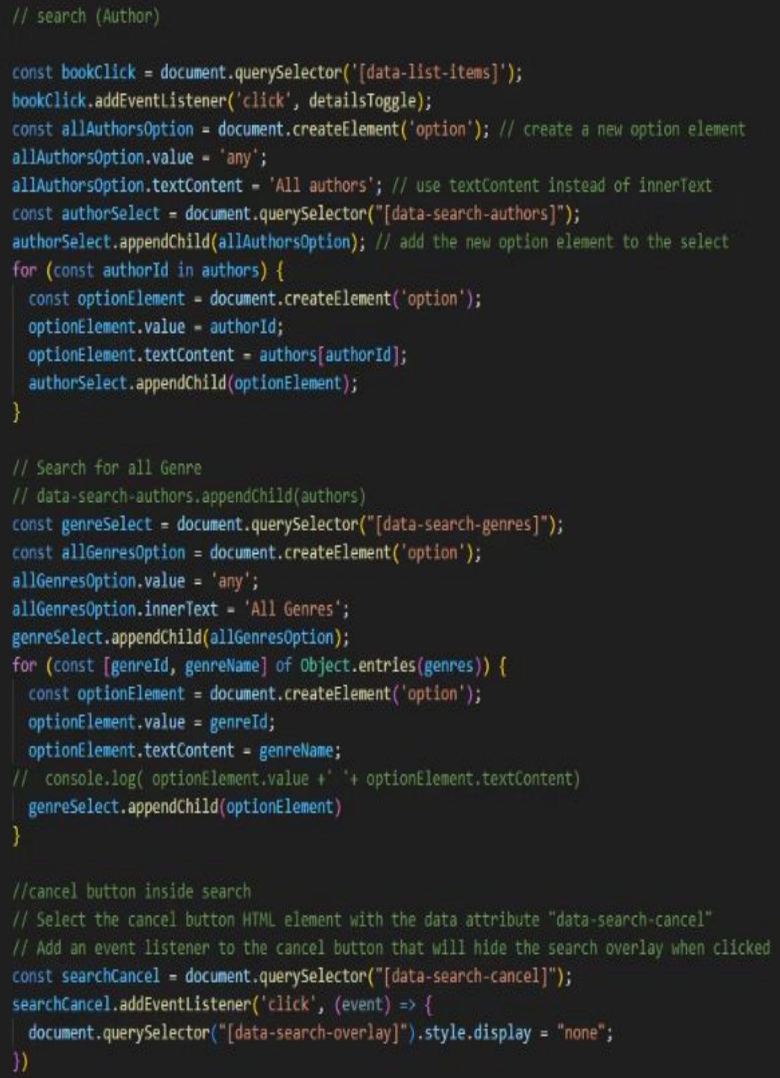


No results found. Your filters might be too narrow.


Title
journeys

Genre
All Genres

Author
All Authors



```
// search (Author)

const bookClick = document.querySelector('[data-list-items]');
bookClick.addEventListener('click', detailsToggle);
const allAuthorsOption = document.createElement('option'); // create a new option element
allAuthorsOption.value = 'any';
allAuthorsOption.textContent = 'All authors'; // use textContent instead of innerText
const authorSelect = document.querySelector("[data-search-authors]");
authorSelect.appendChild(allAuthorsOption); // add the new option element to the select
for (const authorId in authors) {
  const optionElement = document.createElement('option');
  optionElement.value = authorId;
  optionElement.textContent = authors[authorId];
  authorSelect.appendChild(optionElement);
}

// Search for all Genre
// data-search-authors.appendChild(authors)
const genreSelect = document.querySelector("[data-search-genres]");
const allGenresOption = document.createElement('option');
allGenresOption.value = 'any';
allGenresOption.innerText = 'All Genres';
genreSelect.appendChild(allGenresOption);
for (const [genreId, genreName] of Object.entries(genres)) {
  const optionElement = document.createElement('option');
  optionElement.value = genreId;
  optionElement.textContent = genreName;
//  console.log( optionElement.value +' '+ optionElement.textContent)
  genreSelect.appendChild(optionElement)
}

//cancel button inside search
// Select the cancel button HTML element with the data attribute "data-search-cancel"
// Add an event listener to the cancel button that will hide the search overlay when clicked
const searchCancel = document.querySelector("[data-search-cancel]");
searchCancel.addEventListener('click', (event) => {
  document.querySelector("[data-search-overlay]").style.display = "none";
})
```

Script for search function

Show more Overlay

>Updates the text of the “showMore” button many more items will displayed

>I have created variables, assigned them by selecting elements from the html and CSS then appended a fragment to it.

> added an event listener when the element is clicked

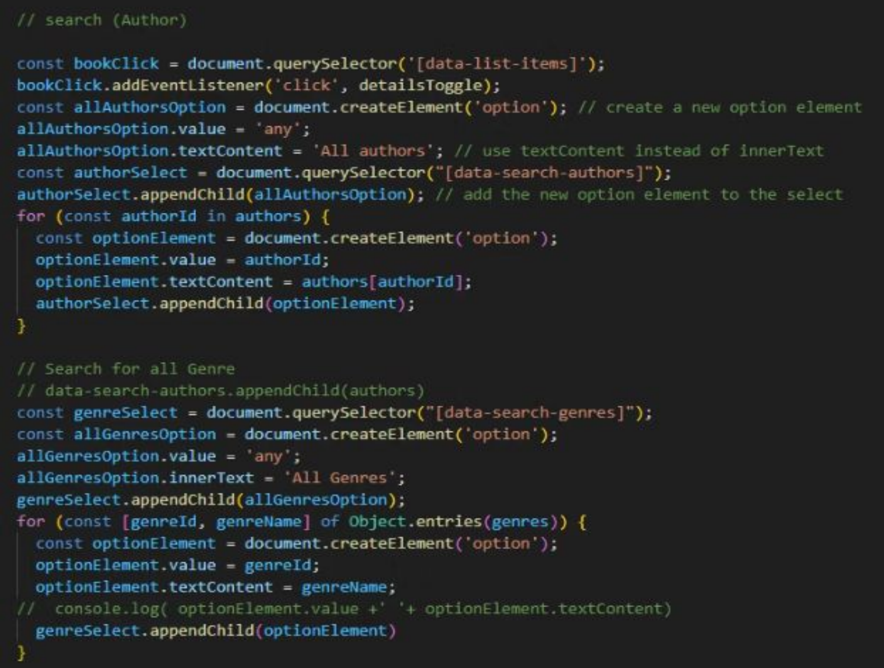
>It then updates the startIndex and endIndex variables to select the next 36 items in an array called books. I also have set the button to countdown when its clicked

>On the showMoreButton i calculated the number of items to show by subtracting the current endIndex from the length of an array and taking the minimum value.

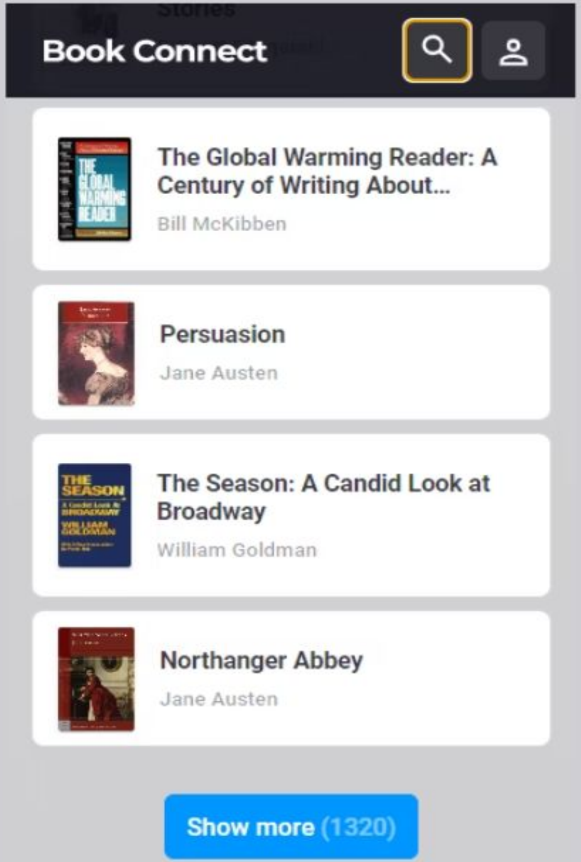
>when the "Show More" button is clicked, the code creates a fragment and appends it to the previously selected HTML element with a data attribute "data-list-items".

>i used the for loop to extract the values of author, image,title, id, description, and published from each object in extracted.

>It then creates a preview element for each selected book



```
// search (Author)

const bookClick = document.querySelector('[data-list-items]');
bookClick.addEventListener('click', detailsToggle);
const allAuthorsOption = document.createElement('option'); // create a new option element
allAuthorsOption.value = 'any';
allAuthorsOption.textContent = 'All authors'; // use textContent instead of innerText
const authorSelect = document.querySelector("[data-search-authors]");
authorSelect.appendChild(allAuthorsOption); // add the new option element to the select
for (const authorId in authors) {
  const optionElement = document.createElement('option');
  optionElement.value = authorId;
  optionElement.textContent = authors[authorId];
  authorSelect.appendChild(optionElement);
}

// Search for all Genre
// data-search-authors.appendChild(authors)
const genreSelect = document.querySelector("[data-search-genres]");
const allGenresOption = document.createElement('option');
allGenresOption.value = 'any';
allGenresOption.innerText = 'All Genres';
genreSelect.appendChild(allGenresOption);
for (const [genreId, genreName] of Object.entries(genres)) {
  const optionElement = document.createElement('option');
  optionElement.value = genreId;
  optionElement.textContent = genreName;
//  console.log( optionElement.value +' '+ optionElement.textContent)
  genreSelect.appendChild(optionElement)
}
```



Book Connect

The Global Warming Reader: A Century of Writing About...
Bill McKibben

Persuasion
Jane Austen

The Season: A Candid Look at Broadway
William Goldman

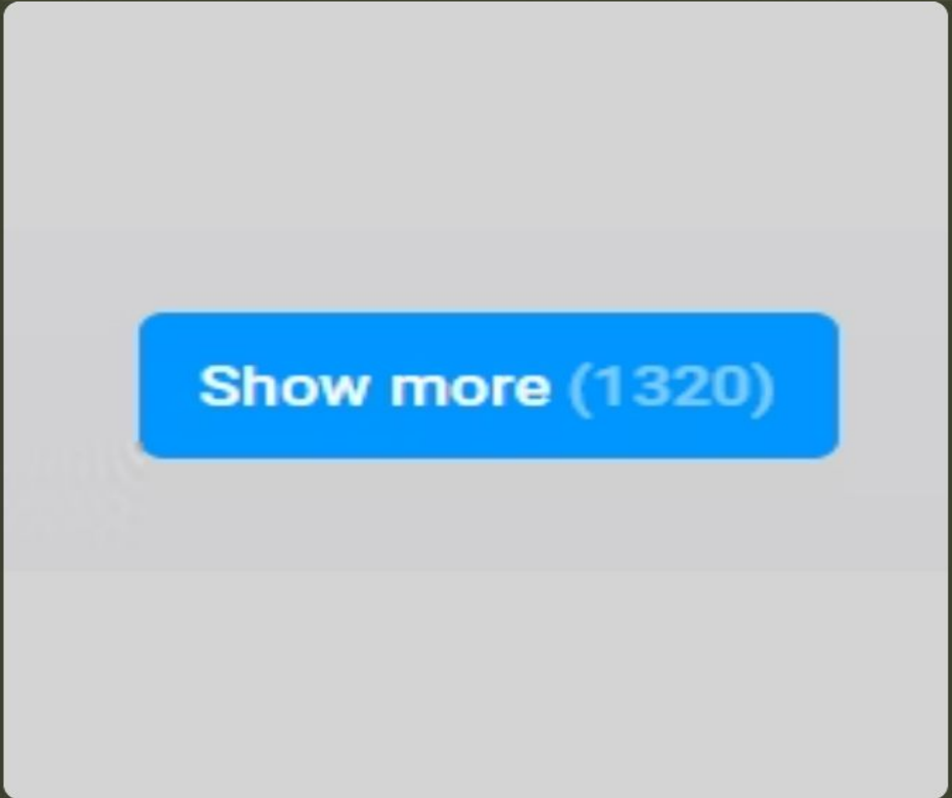
Northanger Abbey
Jane Austen

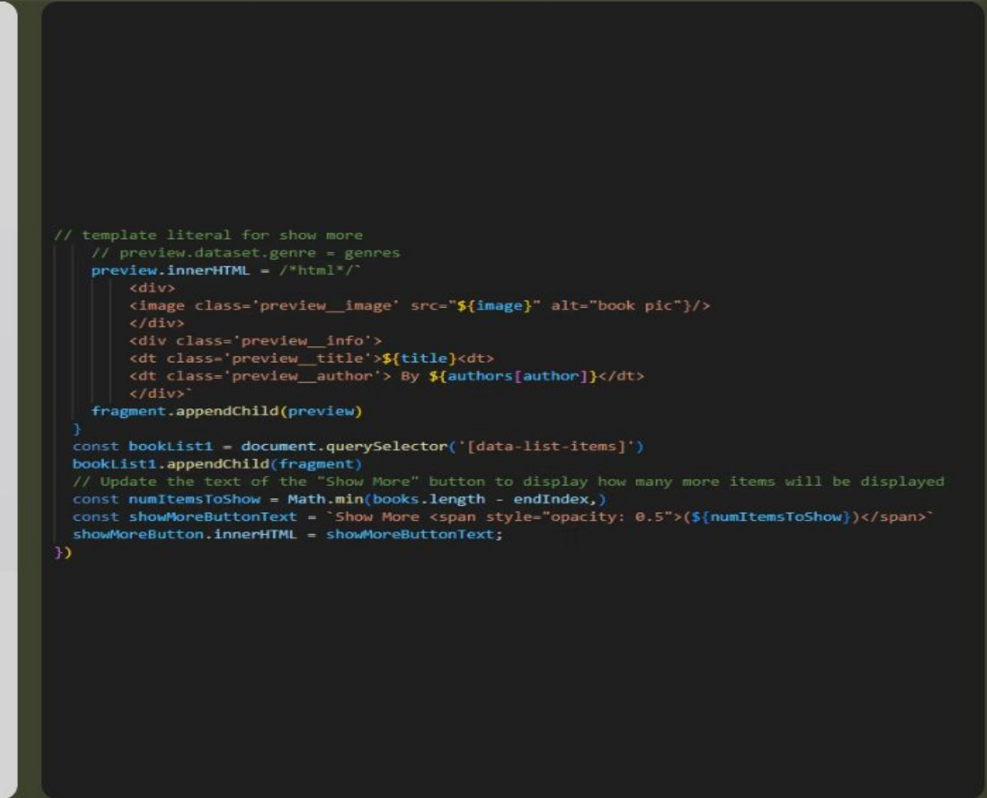
TEMPLATE LITERAL

>I used template literal to generate the HTML content of the dl element

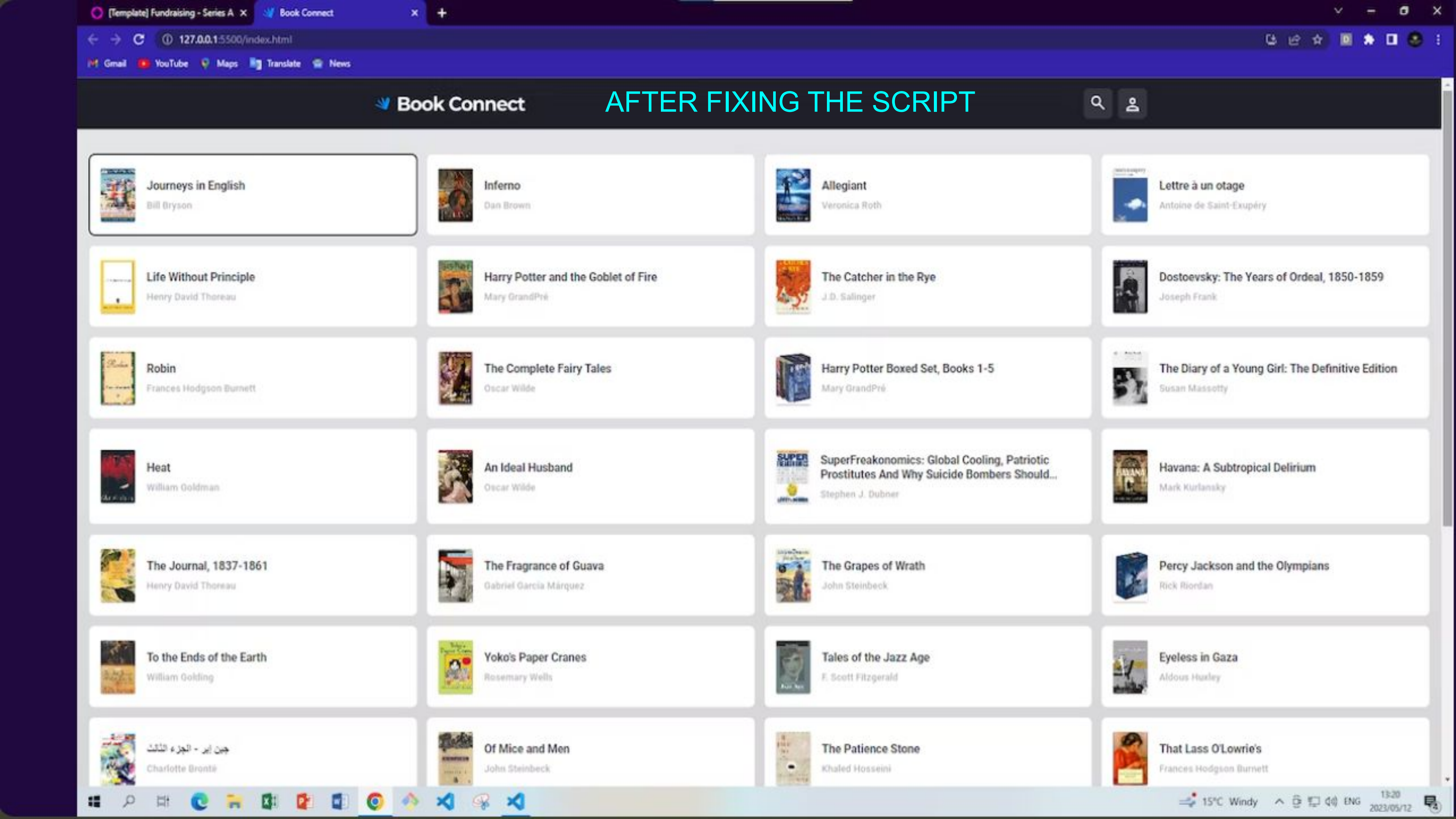
>created the preview element and appended to a fragment

>Update the text of the "Show More" button to display how many more items will be displayed





```
// template literal for show more
    // preview.dataset.genre = genres
    preview.innerHTML = /*html*/`
        <div>
        <image class='preview__image' src="${image}" alt="book pic"}/>
        </div>
        <div class='preview__info'>
        <dt class='preview__title'>${title}<dt>
        <dt class='preview__author'> By ${authors[author]}</dt>
        </div>`
    fragment.appendChild(preview)
  }
  const bookList1 = document.querySelector('[data-list-items]')
  bookList1.appendChild(fragment)
  // Update the text of the "Show More" button to display how many more items will be displayed
  const numItemsToShow = Math.min(books.length - endIndex,)
  const showMoreButtonText = `Show More <span style="opacity: 0.5">(${numItemsToShow})</span>`
  showMoreButton.innerHTML = showMoreButtonText;
})
```



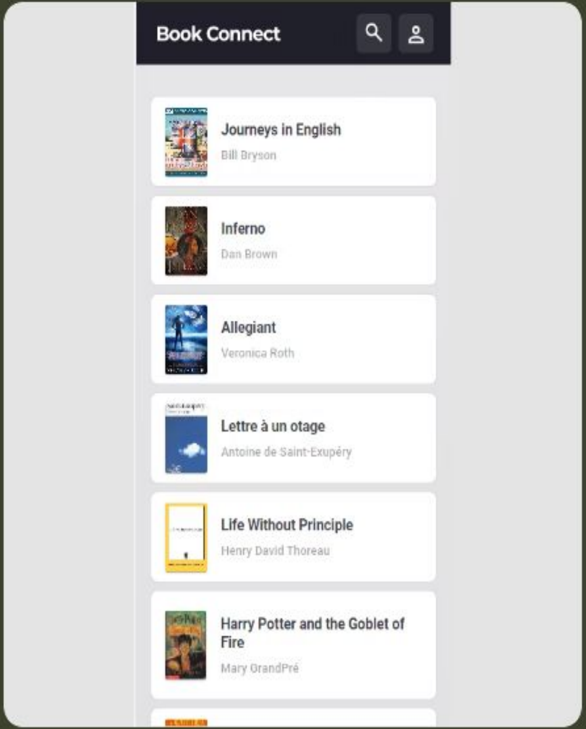
Book Connect

AFTER FIXING THE SCRIPT

Journeys in English Bill Bryson	Inferno Dan Brown	Allegiant Veronica Roth	Lettre à un otage Antoine de Saint-Exupéry
Life Without Principle Henry David Thoreau	Harry Potter and the Goblet of Fire Mary GrandPré	The Catcher in the Rye J.D. Salinger	Dostoevsky: The Years of Ordeal, 1850-1859 Joseph Frank
Robin Frances Hodgson Burnett	The Complete Fairy Tales Oscar Wilde	Harry Potter Boxed Set, Books 1-5 Mary GrandPré	The Diary of a Young Girl: The Definitive Edition Susan Massotty
Heat William Goldman	An Ideal Husband Oscar Wilde	SuperFreakonomics: Global Cooling, Patriotic Prostitutes And Why Suicide Bombers Should... Stephen J. Dubner	Havana: A Subtropical Delirium Mark Kurlansky
The Journal, 1837-1861 Henry David Thoreau	The Fragrance of Guava Gabriel García Márquez	The Grapes of Wrath John Steinbeck	Percy Jackson and the Olympians Rick Riordan
To the Ends of the Earth William Golding	Yoko's Paper Cranes Rosemary Wells	Tales of the Jazz Age F. Scott Fitzgerald	Eyeless in Gaza Aldous Huxley
جين اير - الجزء الثالث Charlotte Brontë	Of Mice and Men John Steinbeck	The Patience Stone Khaled Hosseini	That Lass O'Lowrie's Frances Hodgson Burnett

PICS OF HOW IT FUNCTIONS

SHOWS THE LIST OF BOOKS



Book Connect

Journeys in English
Bill Bryson

Inferno
Dan Brown

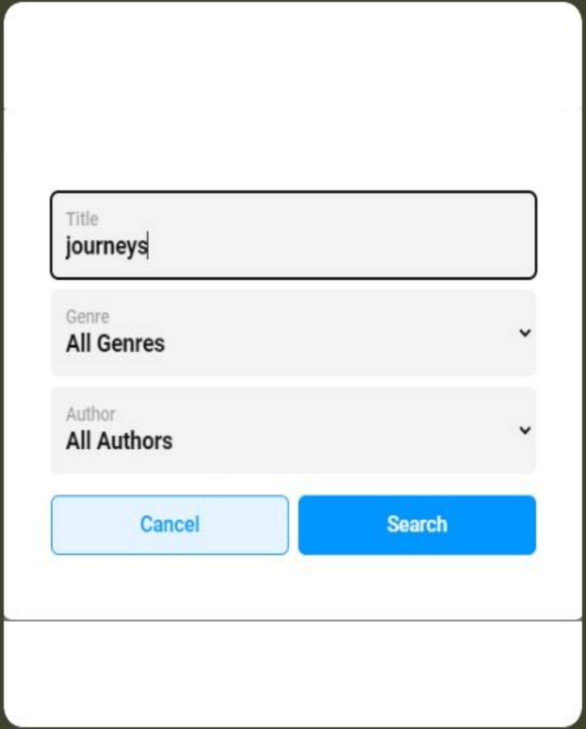
Allegiant
Veronica Roth

Lettre à un otage
Antoine de Saint-Exupéry

Life Without Principle
Henry David Thoreau

Harry Potter and the Goblet of Fire
Mary GrandPré

THE SEARCH MENU

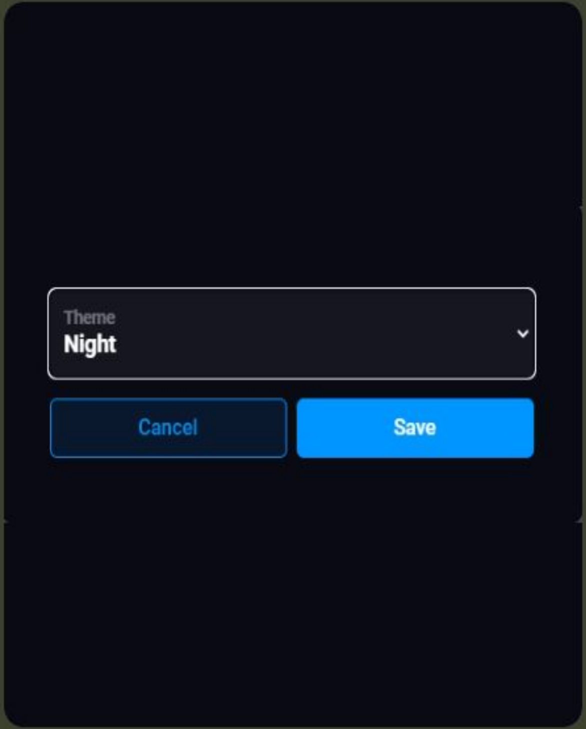


Title: journeys

Genre: All Genres

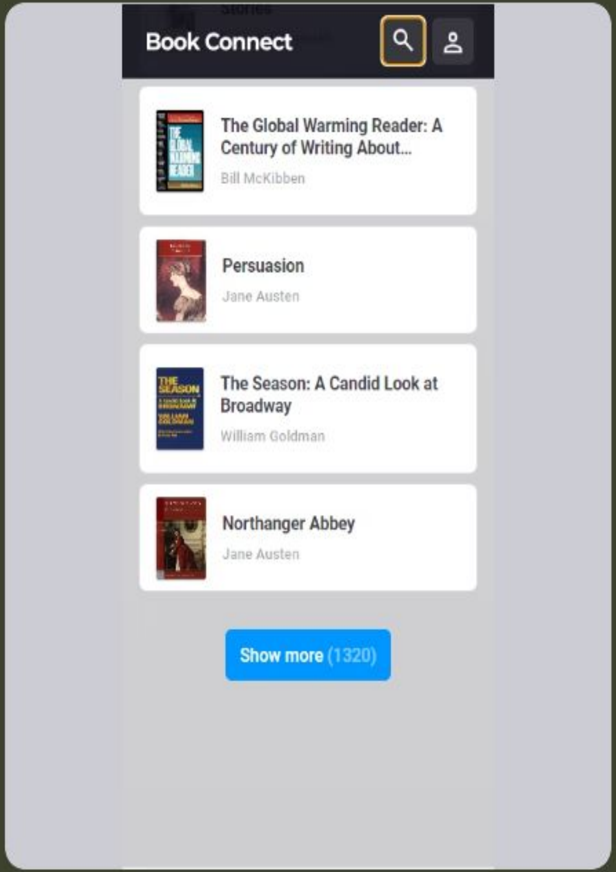
Author: All Authors

THEME



Theme: Night

SHOW MORE FUNCTION



Book Connect

The Global Warming Reader: A Century of Writing About...
Bill McKibben

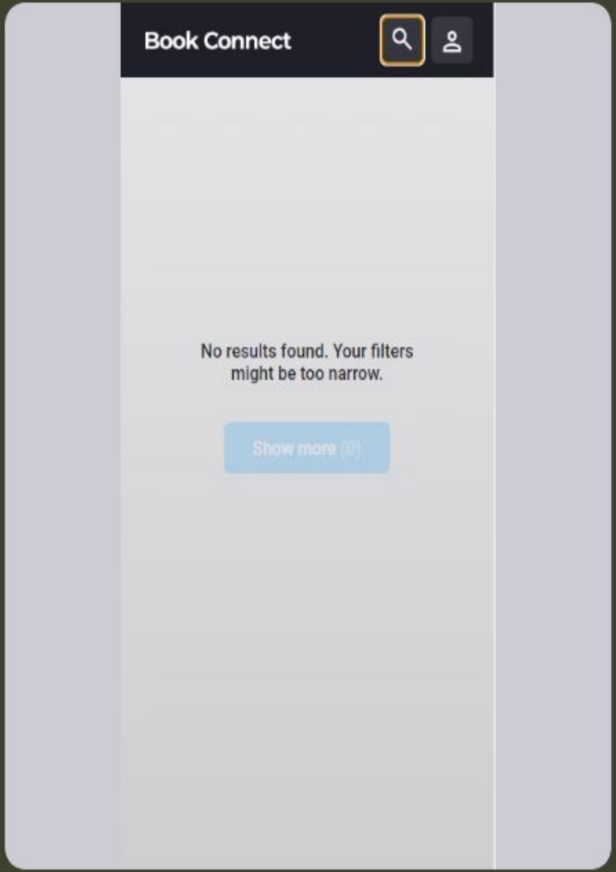
Persuasion
Jane Austen

The Season: A Candid Look at Broadway
William Goldman

Northanger Abbey
Jane Austen

Show more (1320)

WHEN THERE IS NO RESULT

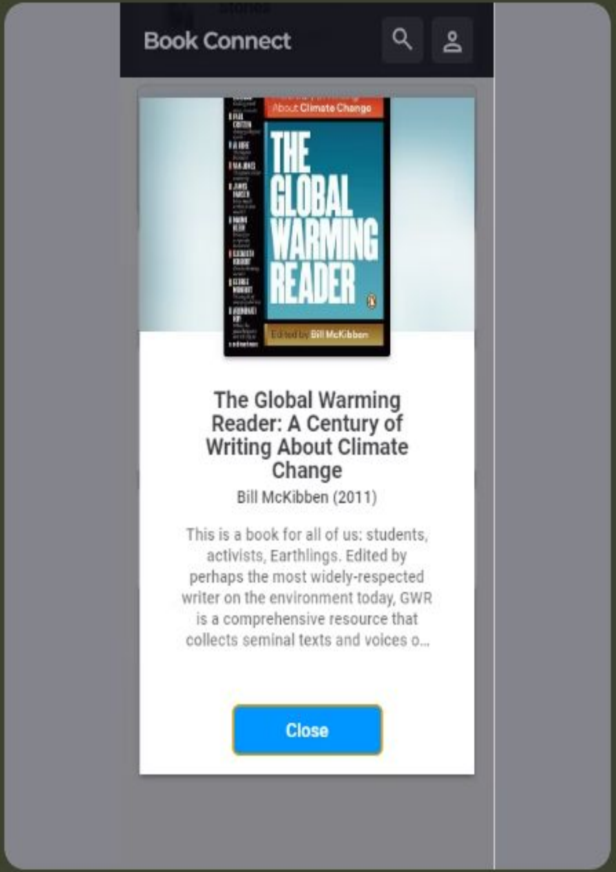


Book Connect

No results found. Your filters might be too narrow.

Show more (0)

BOOK PREVIEW



Book Connect

The Global Warming Reader: A Century of Writing About Climate Change

Bill McKibben (2011)

This is a book for all of us: students, activists, Earthlings. Edited by perhaps the most widely-respected writer on the environment today, GWR is a comprehensive resource that collects seminal texts and voices o...

Close

Thank you!

SHUHAIR SMITH

suhairsmith17@gmail.com