Documentation - Library Management Page

Introduction :

The Library Management Frontend project is a web application that provides users with the ability to login or sign up, view available books, filter the content, and see the summary of featured books. This documentation provides an overview of the project's features, functionality, and usage.

Table of Contents:

1. Tools and Frameworks:

The tools, frameworks that are being used for the development of this task are:
(i) HTML
(ii) CSS
(iii) JavaScript
(iv) Bootstrap
(v) Fire Base

The IDE that is being used for the task is Visual Studio Code.

2. How to use it:

To use this front-end application:

1. Clone this project from the git repository.
2. The index.html is the entry point for this project.
3. By opening the index.html, you will be able to access the whole project.
4. The project uses Firebase for user authentication, Login and SignUp.
5. Replace the FireBase configurations with your “API Key”
6. Now you are set to go for the full usage.

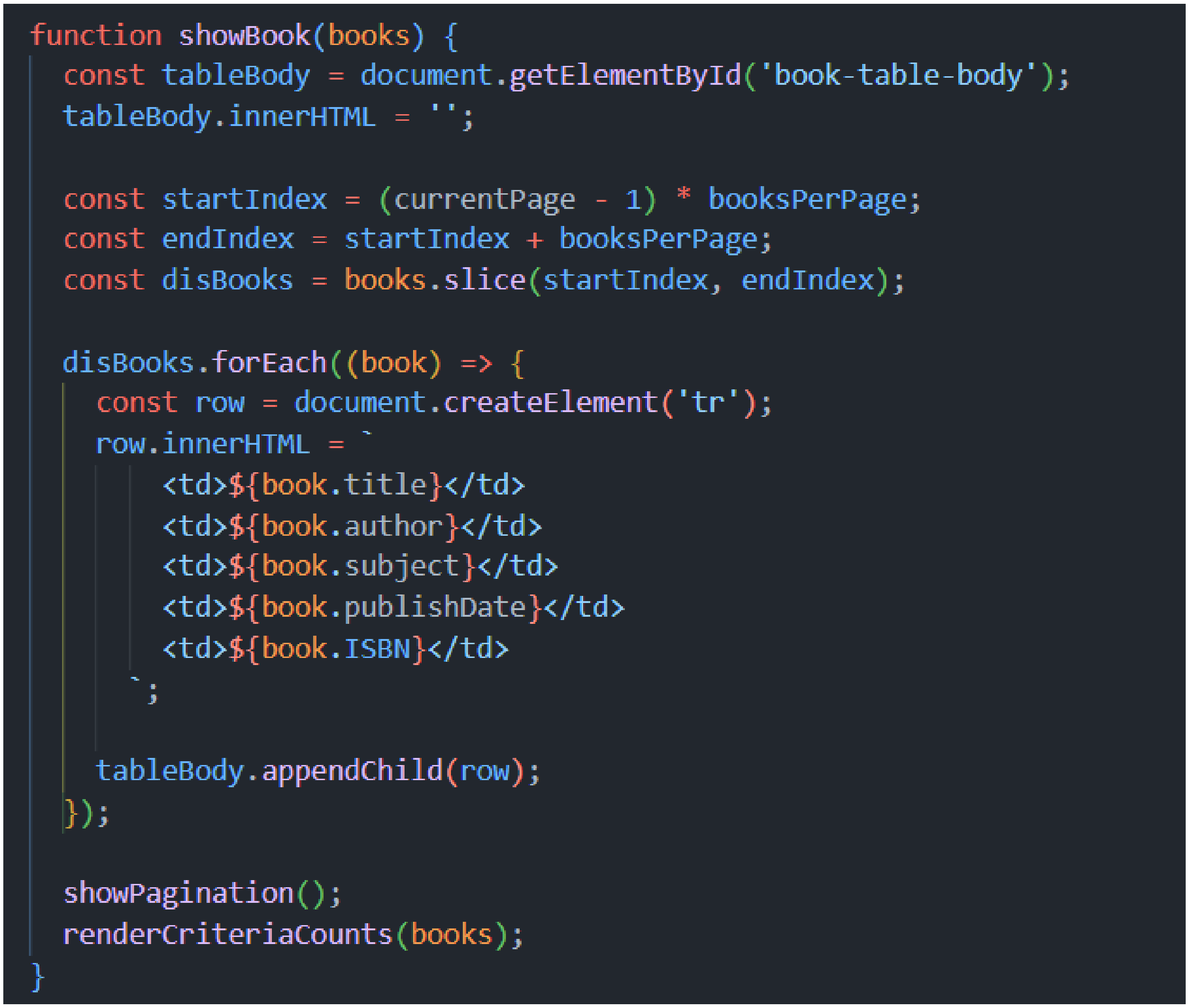
3. Functionalities :

The basic functionalities that are available in this project are:

1. Ability for the user to view the list.
2. Ability to filter out the books based on criteria such as “Book name”, “Author”, “Subject”, “Publishing Date”.
3. The user will be able to see the count of books that are available based on filtering criteria.
4. There is Sign Up and Login functionality without which the user won’t be able to access the book list. Logout functionality can be used for logging out from the session.

(i) Viewing the book list :

The function showBook() in the script1.js which is in the “src” folder performs the operation:



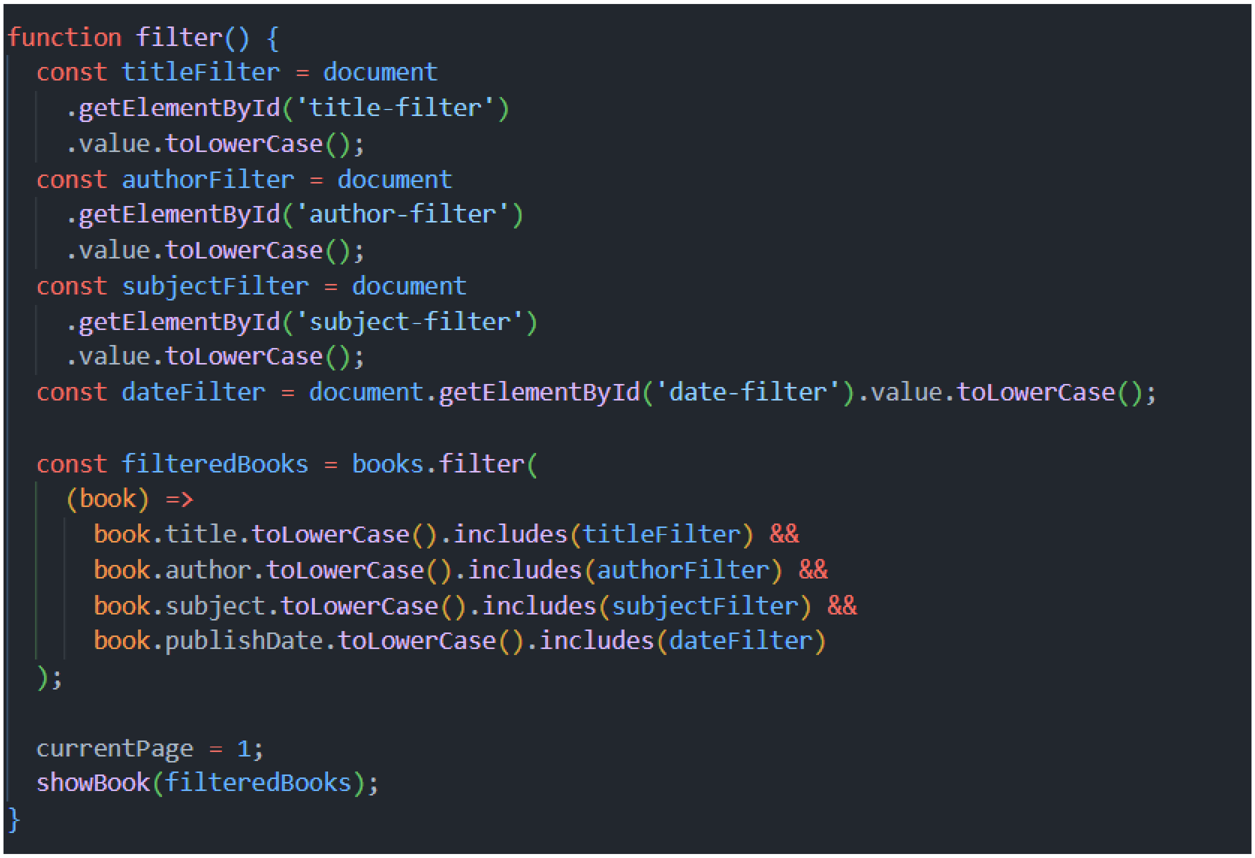
```
function showBook(books) {
  const tableBody = document.getElementById('book-table-body');
  tableBody.innerHTML = '';

  const startIndex = (currentPage - 1) * booksPerPage;
  const endIndex = startIndex + booksPerPage;
  const disBooks = books.slice(startIndex, endIndex);

  disBooks.forEach((book) => {
    const row = document.createElement('tr');
    row.innerHTML = `
        <td>${book.title}</td>
        <td>${book.author}</td>
        <td>${book.subject}</td>
        <td>${book.publishDate}</td>
        <td>${book.ISBN}</td>
      `;

    tableBody.appendChild(row);
  });

  showPagination();
  renderCriteriaCounts(books);
}
```

(ii) Filtering Functionalities:

The user can filter out book based on some criteria:

1. Book name
2. Author
3. Subject
4. Publish Date

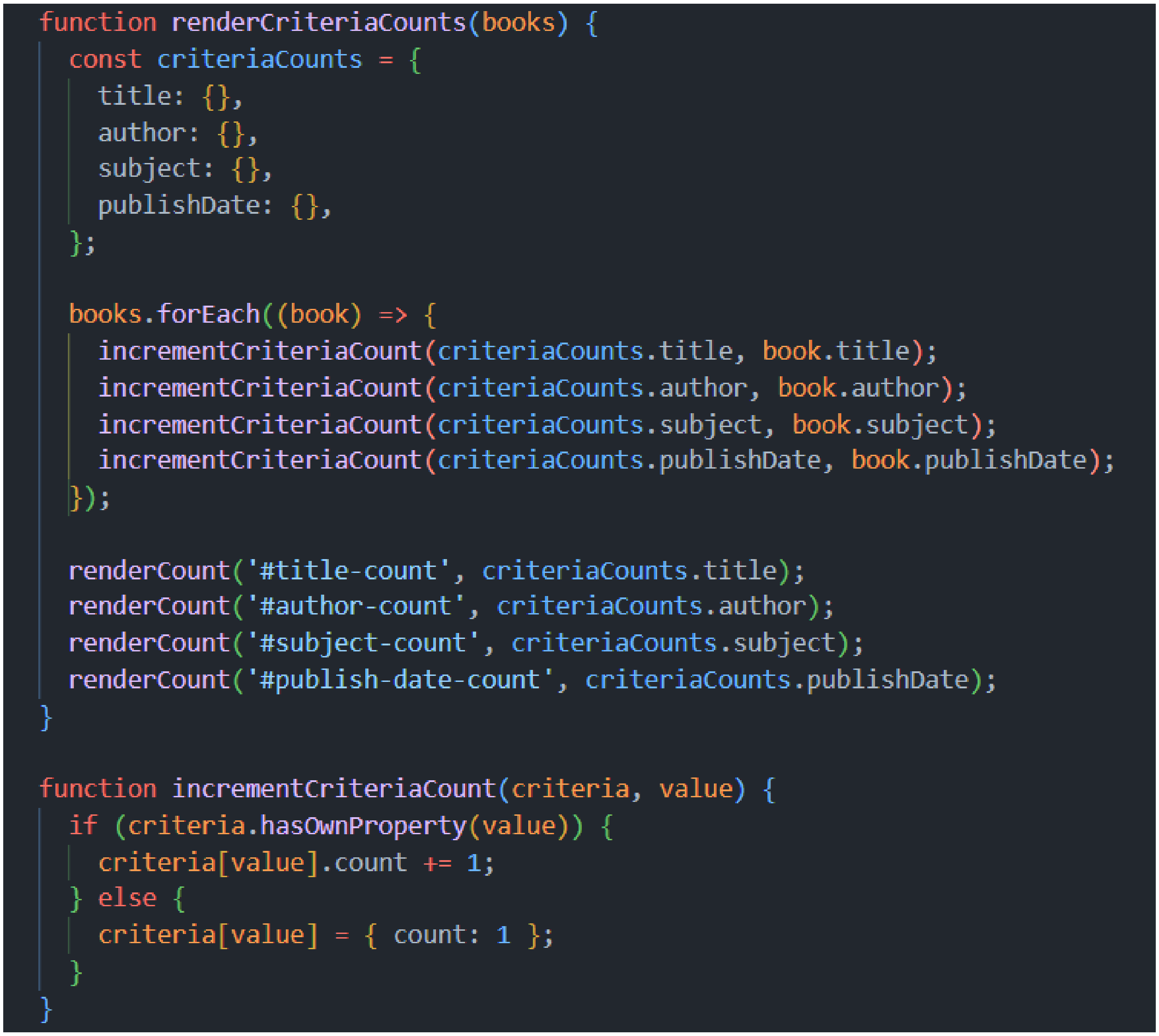
The following functionalities perform the filtering operation:



```
function filter() {
  const titleFilter = document
    .getElementById('title-filter')
    .value.toLowerCase();
  const authorFilter = document
    .getElementById('author-filter')
    .value.toLowerCase();
  const subjectFilter = document
    .getElementById('subject-filter')
    .value.toLowerCase();
  const dateFilter = document.getElementById('date-filter').value.toLowerCase();

  const filteredBooks = books.filter(
    (book) =>
      book.title.toLowerCase().includes(titleFilter) &&
      book.author.toLowerCase().includes(authorFilter) &&
      book.subject.toLowerCase().includes(subjectFilter) &&
      book.publishDate.toLowerCase().includes(dateFilter)
  );

  currentPage = 1;
  showBook(filteredBooks);
}
```

(iii) Count of Books:

The user can see the count of books that are available based on the criteria that is being chosen.



```
function renderCriteriaCounts(books) {
  const criteriaCounts = {
    title: {},
    author: {},
    subject: {},
    publishDate: {},
  };

  books.forEach((book) => {
    incrementCriteriaCount(criteriaCounts.title, book.title);
    incrementCriteriaCount(criteriaCounts.author, book.author);
    incrementCriteriaCount(criteriaCounts.subject, book.subject);
    incrementCriteriaCount(criteriaCounts.publishDate, book.publishDate);
  });

  renderCount('#title-count', criteriaCounts.title);
  renderCount('#author-count', criteriaCounts.author);
  renderCount('#subject-count', criteriaCounts.subject);
  renderCount('#publish-date-count', criteriaCounts.publishDate);
}

function incrementCriteriaCount(criteria, value) {
  if (criteria.hasOwnProperty(value)) {
    criteria[value].count += 1;
  } else {
    criteria[value] = { count: 1 };
  }
}
```

(iv) Sign Up and Login:

The user will be able to access to book list only by signing up or by logging in the page.

Firebase is used for this functionality, The API key must be added to the firebase configurations for accessing it.

4. Summary:

This project provides a very simple interface for the user for using and managing the library, This provides the overall design which can be further extended into a major project by adding major backend functionalities.

This documentation provided an overview of the Library Management Frontend project, describing its features, functionality, and installation steps. Users can now navigate the application, log in or sign up, view available books, filter content, and explore the summary of featured books.