

Contact Information:

Email: ayabermen2014@gmail.com

LinkedIn: www.linkedin.com/in/aya-b-566896127

Book List Form

Book List Form

Overview

This is a simple JavaScript form that allows you to manage a list of books. You can add new books, display them, and remove them from the list. The data is stored in the browser's local storage, so it persists even after you refresh the page.

Key Features

- **Add a Book:** Enter details like title, author, and ISBN to add a new book.
- **Display Books:** View all books that have been added.
- **Remove a Book:** Remove a book from the list by clicking a delete button.

How It Works

The application is divided into three main parts:

1. **Book Class:** Represents a book with properties like title, author, and ISBN.
2. **UI Class:** Handles all user interface operations like displaying books, adding books to the UI, clearing input fields, showing alerts, and deleting books.
3. **Store Class:** Manages storing and retrieving book data from the browser's local storage.

Workflow

- **Adding a Book:** When you submit the form, a new book object is created and added to both the UI and local storage.
- **Displaying Books:** When the page loads, the application retrieves books from local storage and displays them in a table.
- **Removing a Book:** When you click the delete button, the book is removed from the UI and local storage.

Functions Explained

Book Class

- **Constructor:** Initializes a book with `title`, `author`, and `isbn`.

Contact Information:

Email: ayabermen2014@gmail.com

LinkedIn: www.linkedin.com/in/aya-b-566896127

Book List Form

UI Class

- `displayBook()`
 - **Purpose:** Retrieves all books from local storage and displays them on the page.
 - **How It Works:** Calls `Store.getBook()` to get the books and `UI.addBook()` to add each book to the UI.
- `addBook(book)`
 - **Purpose:** Adds a book to the HTML table.
 - **How It Works:** Creates a table row (`<tr>`) with book details and appends it to the book list.
- `clearFields()`
 - **Purpose:** Clears the input fields after adding a book.
 - **How It Works:** Sets the values of the input fields to empty strings.
- `showAlert(msg, className)`
 - **Purpose:** Shows an alert message to the user.
 - **How It Works:** Creates a `div` element with the alert message and displays it above the form. Removes it after 3 seconds.
- `deleteEl(el)`
 - **Purpose:** Deletes a book from the UI.
 - **How It Works:** If the clicked element has the class `delete`, it removes the parent row from the table and shows a success message.

Store Class

- `getBook()`
 - **Purpose:** Retrieves the list of books from local storage.
 - **How It Works:** Checks if there are books in local storage. If not, returns an empty array. Otherwise, parses and returns the stored books.
- `addBook(book)`
 - **Purpose:** Adds a book to local storage.
 - **How It Works:** Gets the current list of books, adds the new book, and saves the updated list to local storage.
- `removeBook(isbn)`
 - **Purpose:** Removes a book from local storage.
 - **How It Works:** Finds and removes the book with the specified ISBN from the list and updates local storage.

Contact Information:

Email: ayabermen2014@gmail.com

LinkedIn: www.linkedin.com/in/aya-b-566896127

Book List Form

JS Explanation

This section explains various JavaScript methods and functionalities used in this application.

JavaScript Methods Used

- `forEach()`

- **Purpose:** Executes a provided function once for each array element.
- **How It Works:** Used to iterate over an array of book objects to perform operations such as displaying each book.
- **Example:**

```
books.forEach(book => UI.addBook(book));
```

- `document.querySelector()`

- **Purpose:** Returns the first element that matches a specified CSS selector.
- **How It Works:** Used to select elements from the DOM (Document Object Model) to interact with, such as input fields or the book list.
- **Example:**

```
const list = document.querySelector("#book-list");
```

- `document.getElementById()`

- **Purpose:** Returns the element that has the specified ID.
- **How It Works:** Used to access a specific element in the DOM by its ID attribute.
- **Example:**

```
const form = document.getElementById("book-form");
```

- `appendChild()`

- **Purpose:** Adds a node to the end of the list of children of a specified parent node.
- **How It Works:** Used to add new elements to the DOM. For example, adding a new row to the book list.
- **Example:**

```
list.appendChild(row);
```

- `setTimeout()`

- **Purpose:** Executes a function after a specified number of milliseconds.

Contact Information:

Email: ayabermen2014@gmail.com

LinkedIn: www.linkedin.com/in/aya-b-566896127

Book List Form

- **How It Works:** Used to delay the execution of a function. In this case, it's used to remove an alert message after a few seconds.

- **Example:**

```
setTimeout(() => document.querySelector('.alert').remove(), 3000);
```

- **classList.contains()**

- **Purpose:** Checks if an element has a specific class.
- **How It Works:** Used to determine if an element contains a certain class, often used in conditional statements to perform actions based on the presence of a class.
- **Example:**

```
if (el.classList.contains('delete')) {  
    // Perform action  
}
```

- **parentElement**

- **Purpose:** Returns the parent element of the specified element.
- **How It Works:** Used to traverse up the DOM tree from the current element.
- **Example:**

```
el.parentElement.remove();
```

- **previousElementSibling**

- **Purpose:** Returns the previous sibling element of the specified element.
- **How It Works:** Used to access the element immediately preceding the current element in the DOM.
- **Example:**

```
const isbn = e.target.parentElement.previousElementSibling.textContent;
```

- **JSON.parse()**

- **Purpose:** Converts a JSON string into a JavaScript object.
- **How It Works:** Used to retrieve and parse JSON data stored in local storage.
- **Example:**

```
books = JSON.parse(localStorage.getItem('books'));
```

- **JSON.stringify()**

- **Purpose:** Converts a JavaScript object into a JSON string.

Contact Information:

Email: ayabermen2014@gmail.com

LinkedIn: www.linkedin.com/in/aya-b-566896127

Book List Form

– **How It Works:** Used to store JavaScript objects as strings in local storage.

– **Example:**

```
localStorage.setItem('books', JSON.stringify(books));
```

Usage

1. **Open the HTML file** in a web browser.
2. **Add a book** by filling out the form with the book's title, author, and ISBN, and then submit the form.
3. **View the list** of books displayed in the table.
4. **Remove a book** by clicking the delete button next to the book you want to remove.

Contributing

Feel free to fork this repository and submit pull requests. If you have any questions or suggestions, open an issue or contact me.