BOM

DEADLINE: 04/11/2020

FOLDER STRUCTURE

```
FL14_HW16/ [*]
├── task [**]
├── homework/ [*]
        ├── index.html [*]
        ├── styles.css [*]
        ├── app.js [*]
        ├── books.js [*]
        └── .eslintrc.js
```

[*] - required
[**] - restricted (no need to submit)
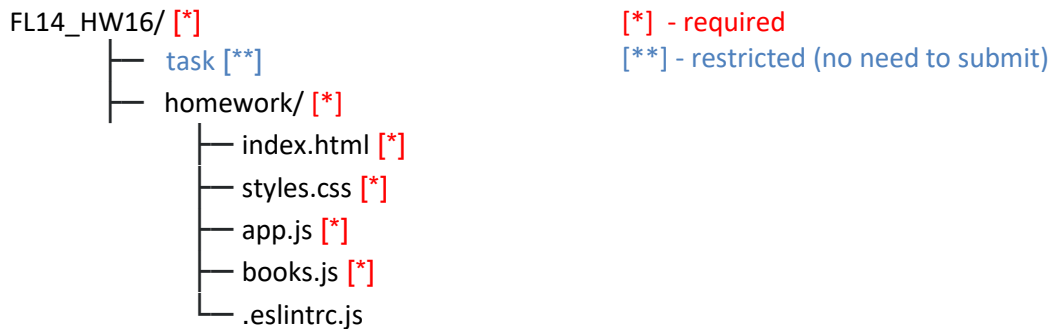
TASK

You need to create a web page that contains an editable set of books.
   NOTE: All styles are up to you (don't waste much time on design).

Your page should be divided into two sections. The first one always
represents a list of books. The second section is a dynamic
one and can represent either **book preview** or **edit book** or **add book** section depending
on the URL.

The list of books is just a set of book names and next to each item there should be
an "*edit*" button. Also, there should be an "*add*" button at the end of the list.
Each book (list item) is a clickable element.

When you click on the book the URL changes to:
/index.html?id={uid}#preview (uid represents the unique id of the book selected)
and the dynamic section renders a book preview. Book preview section displays the
following information: book name, author, image and plot.

When you click on edit button the URL changes to:
/index.html?id={uid}#edit (uid represents the unique id of the book selected)
and the dynamic section renders book editing form. There should be 4 input fields (all required)
filled with the book information (book name, author, image url, plot) and two buttons (cancel
and save). When the cancel button is clicked a modal window appears with "Discard changes?"

message. If the user confirms then discard all the changes and go back to the previous page (the URL must be changed accordingly), on cancelling - stay on the current page (Edit form page).

When save button is clicked – save all the changes, redirect to the book preview (the URL must be changed accordingly - /index.html?id={uid}#preview) and after 200ms show alert modal window ("*Book successfully updated*").

When you click on add button the URL changes to:
/index.html#add
and the dynamic section renders a book adding form. This form is the same as the book editing form but with unpopulated input fields (all the requirements are the same). After adding the book, it should be saved and added to the end of the book list.

When you open the page for the very first time you should see three books on your page (they should be defined in books.js file).

After refreshing the page all the saved changes must be kept (use local storage to save data). Also, after manually changing the URL the web page must be rendered accordingly. For example, if you manually enter /index.html?id=2#preview, the dynamic section must render the preview for the book with id === 2. If manually entered URL is invalid (has wrong or extra query params/hash or non-existent book id) then render the home page (/index.html) – just a list of books.

Pay attention:
- browser *back/forward* buttons must work correctly
- you can edit *app.js, book.js* and *styles.css* files but NOT index.html
- all DOM manipulations must be done in *app.js* file

RESTRICTIONS

- Editing index.html is forbidden
- You don't need to use any frameworks or libs for routing, just pure JS
- You don't need to spend much time on implementing design of the application

BEFORE SUBMIT

- Remove all unnecessary files that you might have included by mistake
- Verify that all functionality is implemented according to requirements
- Make sure you code is well-formatted, and validated via validator (w3org Markup Validation Service)

- Add comments if the code is difficult to understand
- Fix warnings/errors in the browser console
- Verify that the name of the folders and files meet the requirements
- Make sure there are no errors/warnings in the browser console
- Run the linter and fix all warnings and errors.

SUBMIT

- The folder should be uploaded to GitLab repository 'FL-14' into master branch

USEFUL LINKS

- https://developer.mozilla.org/en-US/docs/Web/API/History_API
- https://developer.mozilla.org/en-US/docs/Web/API/URL
- https://developer.mozilla.org/en-US/docs/Web/API/Window/popstate_event
- https://javascript.info/settimeout-setinterval
- https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage