BOM

**DEADLINE:** 17/05/2020

## FOLDER STRUCTURE

|  |  |
| --- | --- |
| FL13\_HW12/ [\*]  ├─ 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.

Нам нужно сделать веб страницу, которая будет содержать набор книг, которые можно редактировать

All styles are up to you (don’t waste much time on design). Какой будет стиль – вам решать (не тратье много времени на дизайн)

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** sectiondepending on the URL. Ваша старница должна быть разделена на две части. В прервой части должен быть список книг. Вторая часть – динамическая. Она может представлять раздел либо **обзор** книг либо **редактирование** книг либо **добавить** книгу, раздел зависит от 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. Список книг – это набор наименований книг и рядом с каждым елементом будет кнопка “редактировать”. В конце списка должна быть кнопка “добавить”. Каждая книга (элемент списка) – должна быть кликабельным элементом.

*When you click on the book the URL changes to: Когда вы кликаете по книги, URL меняется на:*

/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. /index.html?id={uid}#preview (uid – это уникальный id выбранной книги) и динамическая секция показывает обзор книги. Обзор книги содержит: название книги, автор, фото и содержание.

*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).

4 поля (обязательны для заполнения): название, автор, адрес фото, содержание, а тажке 2 кнопки (отмена и сохранить). Когда нажата кнопка отменить, появляется модальное окно: Discard changes? Если пользователь подтверждает, то нужно вернуться на предыдущую старницу (УРЛ должен измениться соответсвенно), если нажата отмена – остаться на той же странице (редактировать)

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 300ms 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.

Форма – добавить книгу. Такая же как и форма с редактированием, но при *unpopulated input fields* После добавления она сохраняется в конце списка

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). В начале при открытии, должны быть показаны 3 книги на странице (они заданы в books.js )

After refreshing the page all the saved changes must be kept (use local storage to save data). Использовать local storage to save data после рефрешинга страници для сохранения всех изменений.

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.

Также, после ручного изменнения адреса старници, соответствующая страница должна быть отабражена. То есть если введено /index.html?id=2#preview, то динамическая часть должна быть – обзор книги с 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.

Если вручную введен невалидный адрес (неправильный параметр. Или не существующий ид) то показать старницу (/index.htm)

Pay attention that browser back/forward buttons must work correctly. В барузере должны корректно работать кнопки back/forward

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

* Code should be clean, without comments, readable, and tested
* Make sure your GitHub folder structure meets folder structure from this document (without **task** folder)

## SUBMIT

* The **FL13\_HW12** folder without **task** folder should be uploaded to github repository “**FL-13**” 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>