

JS I DOM NA PRZYKŁADZIE LISTY TODO

SPIS TREŚCI

Spis treści.....	1
Cel zajęć.....	1
Rozpoczęcie.....	1
Uwaga.....	2
Wymagania.....	2
Strona HTML.....	3
Klasa Todo.....	3
Dodawanie pozycji listy.....	3
Usuwanie pozycji listy.....	4
Edycja pozycji listy.....	4
Odczyt / Zapis LocalStorage.....	5
Wyszukiwanie.....	6
Commit projektu do GIT.....	6
Podsumowanie.....	7

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- przemieszczania się po drzewie DOM;
- dodawania, usuwania, edytowania elementów drzewa DOM.

W praktycznym wymiarze utworzona zostanie dynamiczna lista czynności do zrobienia (lista To Do).

ROZPOCZĘCIE

Rozpoczęcie zajęć. Powtórzenie metod przemieszczania się po drzewie DOM.

Wejściówka?

UWAGA

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do Plik -> Informacje -> Właściwości -> Właściwości zaawansowane -> Niestandardowe i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub Ctrl+A -> F9.

WYMAGANIA

W ramach LAB B przygotowane powinny zostać:

- pojedyncza strona HTML ze skryptem ładowanym z zewnętrznego pliku JS
- lista zadań
- na dole listy pole tekstowe do dodawania nowych zadań, pole typu data/czas do określenia terminu wykonania zadania, przycisk dodawania zadania
- walidacja nowych zadań: co najmniej 3 znaki, nie więcej niż 255 znaków, data musi być pusta albo w przyszłości
- na górze listy pole wyszukiwarki
- po wpisaniu w wyszukiwarkę co najmniej 2 znaków na liście wyświetlają się wyłącznie pozycje zawierające wpisaną w wyszukiwarkę frazę
- wyszukiwana fraza zostaje wyróżniona w każdym wyniku wyszukiwania
- kliknięcie na dowolną pozycję listy zmienia ją w pole edycji; kliknięcie poza pozycję listy zapisuje zmiany
- obok każdej pozycji listy znajduje się przycisku Usuń / Śmiertnik
- wpisy na liście zapisują się do Local Storage
- po odświeżeniu strony lista wypełnia się wpisami z Local Storage

Mockupy:

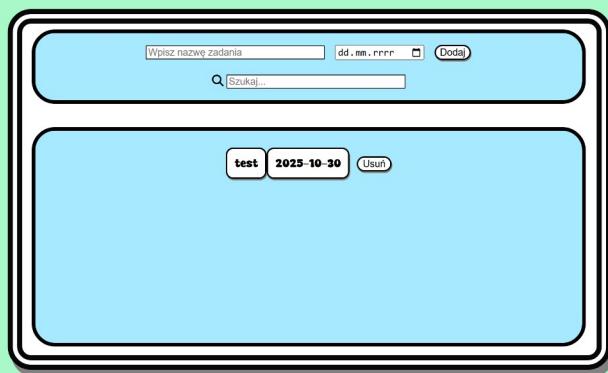
The first mockup shows a simple list of tasks with checkboxes, dates, and delete icons. The second mockup adds a search field and a more complex 'Zapisz' button.

This mockup shows a simplified search interface and a combined 'Zapisz' button.

STRONA HTML

Prace rozpocznij od implementacji HTML z danymi wpisanymi „na sztywno”. Upewnij się, że wstawione zostały wszystkie wymagane elementy – pole wyszukiwarki, lista, pole dodawania, przycisk usuwania. To laboratorium koncentruje się na JS, więc może być ładne, ale nie musi. **Nie trać za dużo czasu na CSS** – to jest laboratorium z JS.

Wstaw zrzut ekranu przedstawiający stronę HTML z polem wyszukiwarki, listą, polem dodawania, przyciskami usuwania:



Punkty:

0

1

KLASA TODO

Pierwszym instynktem może być chęć dodania zachowań bezpośrednio do elementów listy w drzewie DOM. Chociaż na krótką metę wydaje się być to najprostsze rozwiązanie, za chwilę okaże się krótkowzroczne i trudne do implementacji przy kolejnych punktach 😊

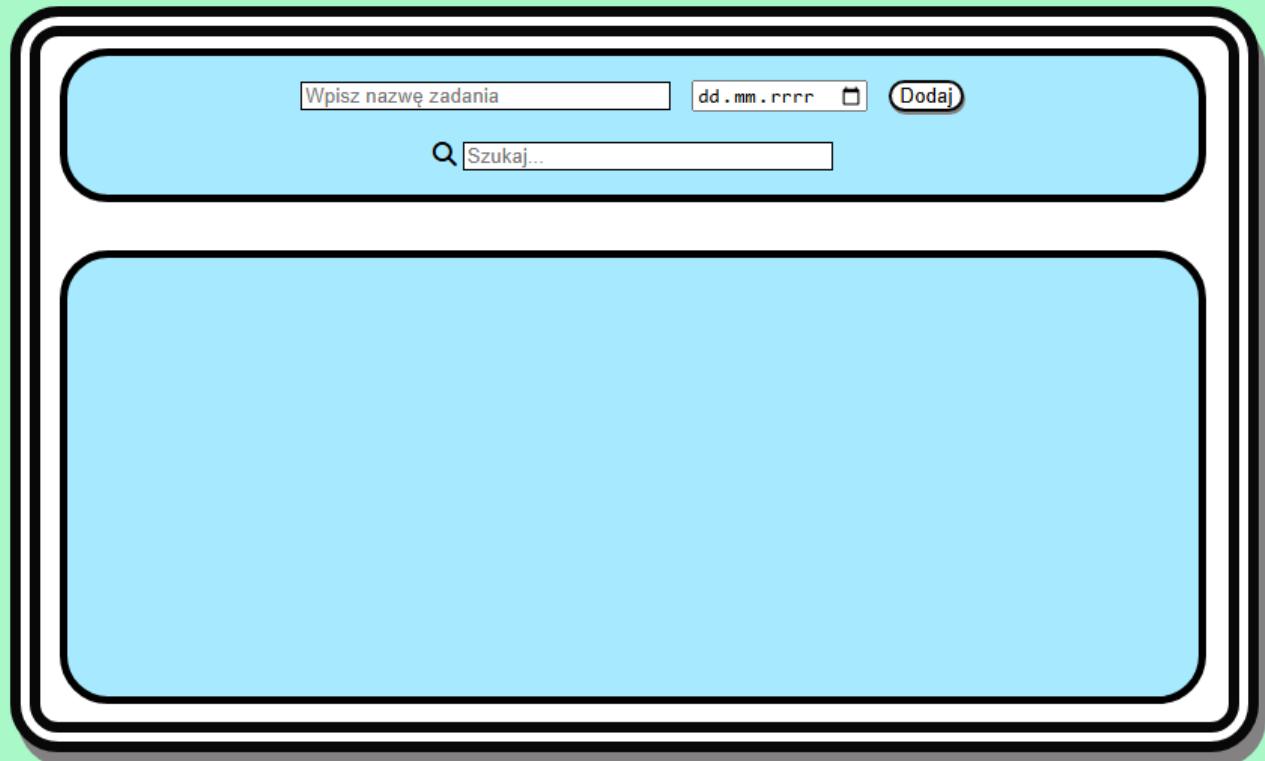
Najlepszym sposobem rozwiązania tego laboratorium jest utworzenie klasy Todo (albo po prostu obiektu z kilkoma metodami). Bez względu na przyjętą strategię, należy w tym nowoutworzonym bycie utworzyć tablicę `tasks` oraz metodę `draw()`, która wyczyści `div` z obecną wizualizacją zadań do zrobienia i wygeneruje ją na nowo na podstawie tablicy `tasks`.

W celu sprawdzenia poprawności działania, najlepiej dostać się do tablicy `tasks` i edytować jej zawartość, po czym ręcznie wywołać metodę `draw()`. Jeśli zawartość listy wyrenderuje się na nowo poprawnie – możemy iść dalej!

Zaimplementuj dodawanie, usuwanie, edycję pozycji listy – wszystko modyfikujące tablicę `tasks` i wywołujące na koniec metodę `draw()`.

DODAWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed dodaniem nowego zadania:



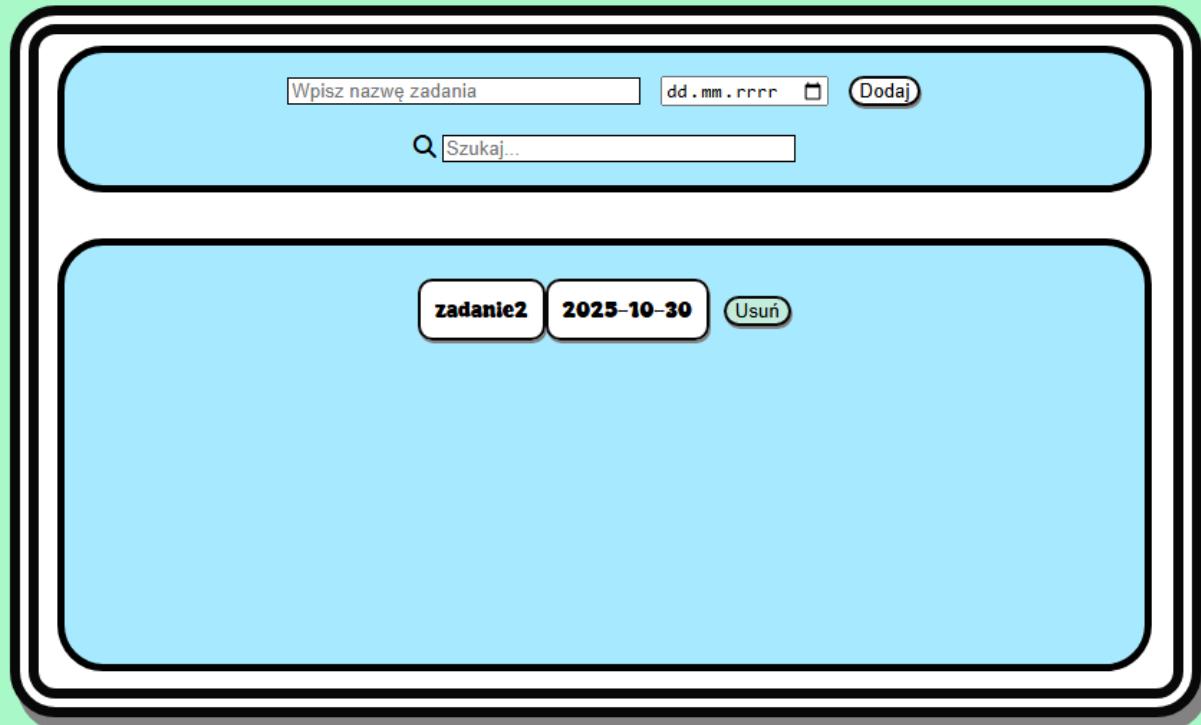
Wstaw zrzut ekranu listy po dodaniu nowego zadania:

zadanie testowe **2025-10-30**

Punkty:	0	1
---------	---	---

USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:



Wstaw zrzut ekranu listy po usunięciu zadania:

A large, rounded rectangular input form with a black double-line border and a gray shadow at the bottom. Inside the form, there are two horizontal input fields. The first field contains the placeholder "Wpisz nazwę zadania". To its right is a date input field with the placeholder "dd.mm.rrrr" and a small calendar icon. Next to the date field is a "Dodaj" button with a plus sign. Below these fields is a search bar with a magnifying glass icon and the placeholder "Szukaj...". The entire form is set against a light green background.

Punkty:

0

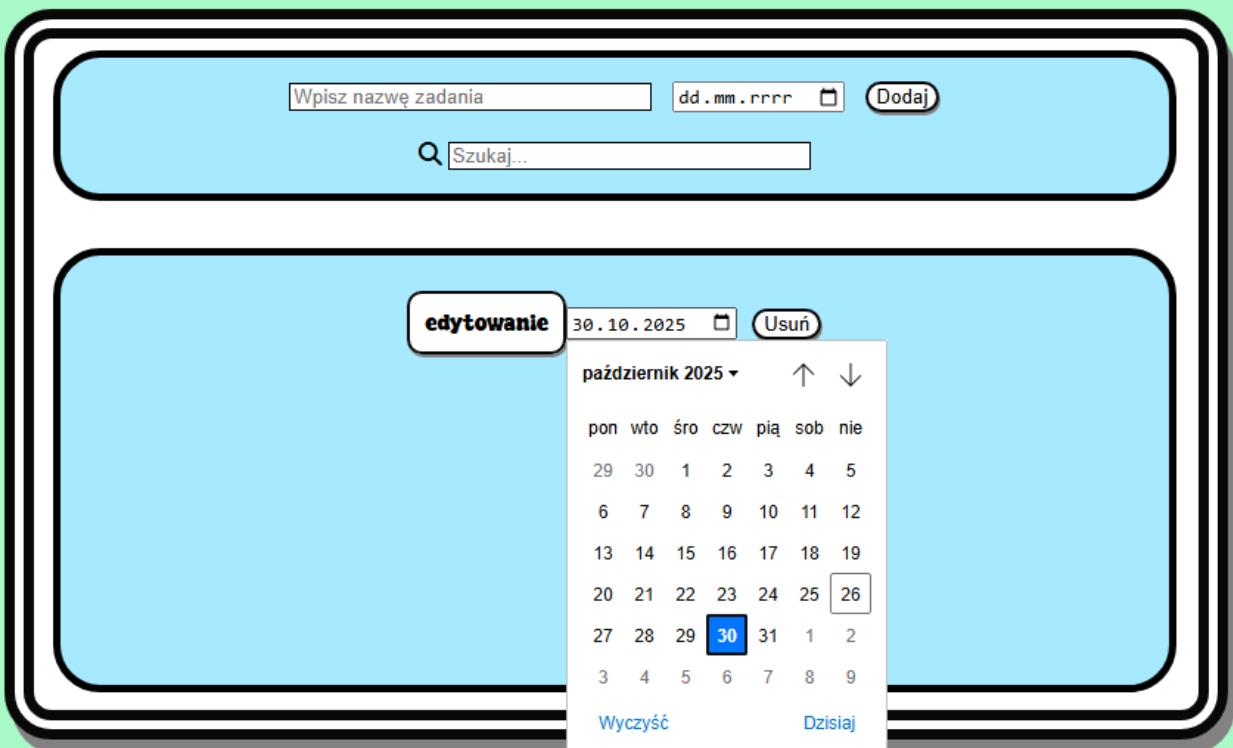
1

EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:



Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:



Wstaw zrzut ekranu listy po edycji zadania i daty. Upewnij się, że dane się zapisały i zadanie jest zmienione:

The screenshot shows a user interface for managing tasks. At the top, there is a search bar with a placeholder 'Szukaj...' and a date input field 'dd.mm.yyyy' with a calendar icon. Below these are two buttons: 'Dodaj' (Add) and a magnifying glass icon. The main area displays a list item with a blue background. This item has three buttons at the top: 'edytowanie' (Edit), '2025-10-27' (date), and 'Usuń' (Delete). The rest of the list item is blank.

Punkty:

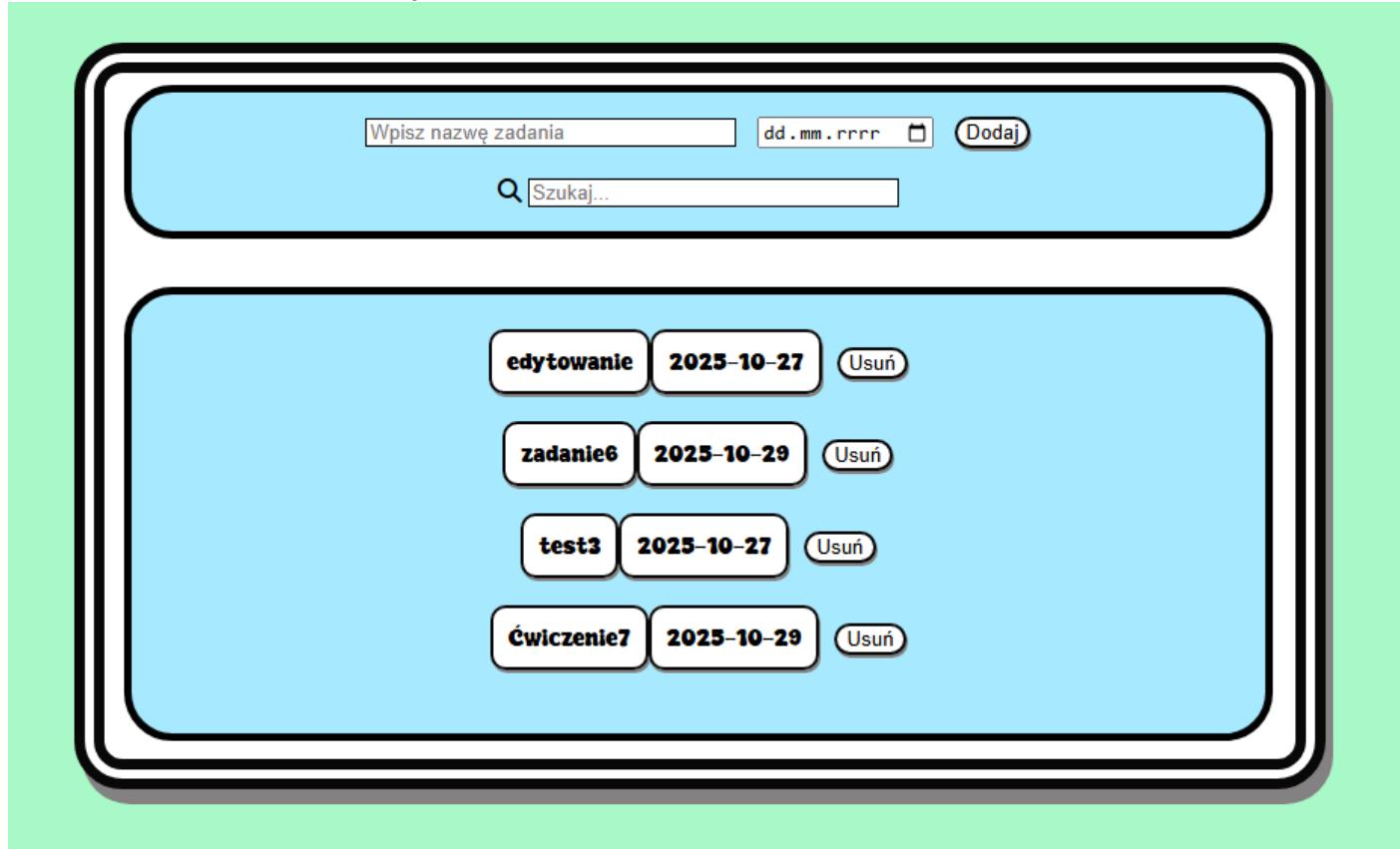
0

1

ODCZYT / ZAPIS LOCALSTORAGE

Zastosowanie klasy Todo w realizacji tego laboratorium pozwala w bardzo łatwy sposób odczytywać i zapisywać stan listy do pamięci przeglądarki. Wystarczy serializacja / deserializacja za pomocą `JSON.parse()` i `JSON.stringify()`.

Wstaw zrzuty ekranu przedstawiające wygląd listy i zawartość local storage gdy na liście są pewne zadania:



The screenshot shows the "Application" tab of the Chrome DevTools developer console. On the left, a sidebar lists storage types: Manifest, Services, Storage, Sessions, Extended Storage, IndexDB, Cookies, Private Browsing, Intererfaces, Shares, Caches, and Local Storage. The "Local Storage" section is expanded, showing a table with one item: "zadania" (Key) with a value of "edytowanie|2025-10-27;zadanie6|2025-10-29;test3|2025-10-27;Ćwiczenie7|2025-10-29" (Value). The table has columns "Key" and "Value".

Wstaw zrzuty ekranu przedstawiające wygląd listy i zawartość local storage po dodaniu nowej pozycji listy. Upewnij się, że widoczne w local storage są dane dotyczące nowego zadania:

edytowanie | 2025-10-27 | Usuń

zadanie6 | 2025-10-29 | Usuń

test3 | 2025-10-27 | Usuń

Ćwiczenie7 | 2025-10-29 | Usuń

nowe zadanie | 2025-11-01 | Usuń

Elements Console Sources Network Performance Memory Application Privacy and security > 7 ⚙️

Application

- 📄 Manif...
- ⚙️ Servic...
- 📅 Stora...

Storage

- ▶ 📂 Local ...
 - 📄 file://
- ▶ 📂 Sessi...
- 📄 Exten...
- 📄 Index...
- ▶ 🍳 Cooki...
- 📄 Privat...
- 📄 Intere...
- ▶ 📂 Share...
- 📄 Cach...
- 📅 Stora...

file://

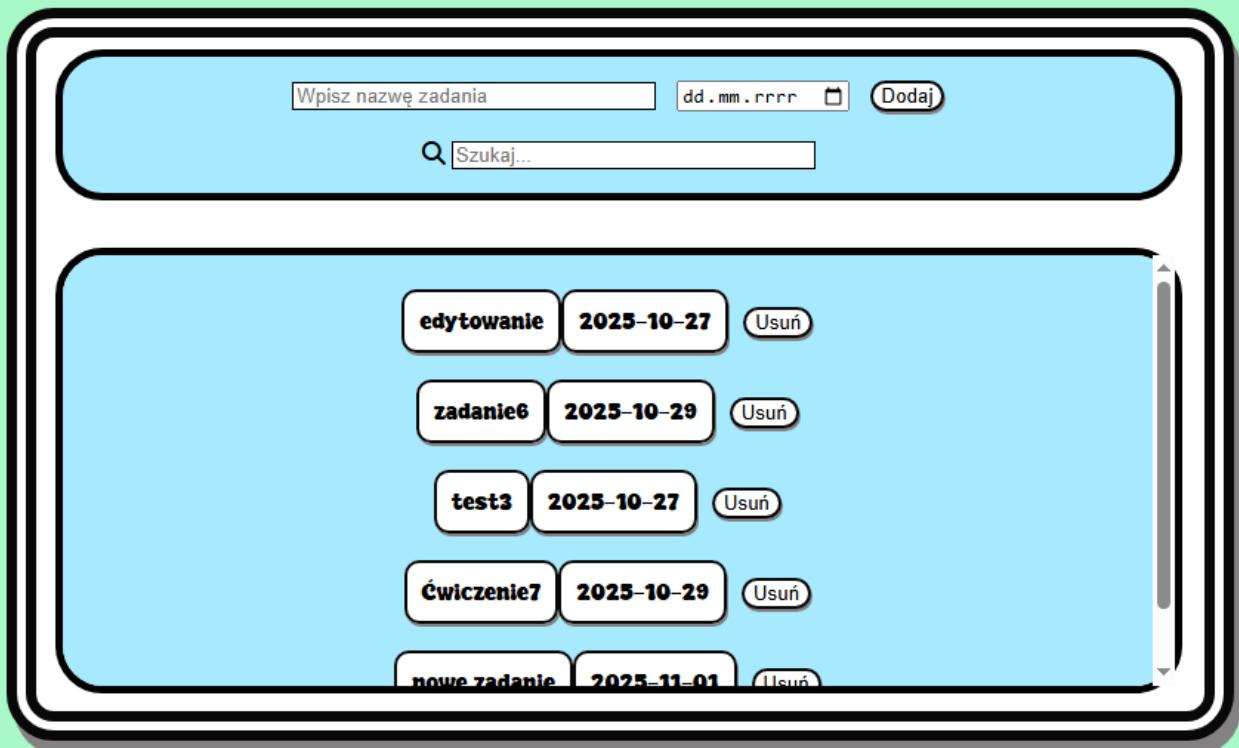
Origin	file://
zadania	edytowanie 2025-10-27;zadanie6 2025-10-29;test3 2025-10-27;Ćwiczenie7 2025-10-29;nowe zadanie 2025-11-01

Punkty: 0 1

WYSZUKIWANIE

Na koniec zostało filtrowanie wyników. Proponowanym podejściem do tego tematu jest umieszczenie w klasie Todo właściwości `term` – frazy wyszukiwanej przez użytkownika. Następnie można utworzyć metodę `getFilteredTasks`, albo getter `filteredTasks`, która zwracać będzie te elementy tablicy `tasks`, które odpowiadają zapytaniu. Można użyć funkcji wyższego rzędu `filter()`.

Wstaw zrzut ekranu listy, gdy pole wyszukiwania jest puste:



Wstaw zrzut ekranu listy, gdy w polu wyszukiwania wpisano wystarczająco dużo znaków, by zadziałało filtrowanie. Upewnij się, że chociaż 2 wyniki będą wciąż widoczne:

The screenshot shows a user interface for managing tasks. At the top, there is a search bar with the placeholder "Wpisz nazwę zadania" and a date input field "dd.mm.yyyy" followed by a calendar icon and a "Dodaj" button. Below the search bar is a search results area containing two items:

- A task card for "zadanie6" due on "2025-10-29" with a "Usuń" (Delete) button.
- A task card for "nowe zadanie" due on "2025-11-01" with a "Usuń" (Delete) button.

Punkty:

0

1

Wstaw zrzut ekranu przedstawiający podświetlenie szukanej frazy w wynikach wyszukiwania, przykładowo dla frazy ko i zadania Ala ma kota otrzymujemy: Ala ma kota:

The screenshot shows a user interface for managing tasks. At the top, there is a search bar with a placeholder "Wpisz nazwę zadania" and a date input field "dd.mm.rrrr" with a calendar icon. A "Dodaj" button is also present. Below the search bar is a search field containing the text "ko". The main area displays a single task card with the title "Ala ma kota", the due date "2025-10-28", and a "Usuń" (Delete) button.

Punkty:

0

1

COMMIT PROJEKTU DO GIT

Zacommittuj i pushnij swoje rozwiązanie do repozytorium GIT.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie lab-b na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-b w swoim repozytorium: https://github.com/gp55571/AI1_Guzowski_Piotr/tree/lab-b
...link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-b...>

PODSUMOWANIE

W kilku słowach/zdaniach napisz swoje przemyślenia odnośnie tego laboratorium. Nie używaj LLM.

...podsumowanie...

