

## 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 przycisk Usun / Śmietnik
- wpisy na liście zapisują się do Local Storage
- po odświeżeniu strony lista wypełnia się wpisami z Local Storage

Mockupy:

Search bar:

<input type="checkbox"/> chocolate	2000-01-01	
<input type="checkbox"/> macaroon		
<input type="checkbox"/> chupa chups		
<input type="checkbox"/> candy canes	2000-01-05	
<input type="checkbox"/> bon bons		

do zrobienia...  2000-01-01

Search bar:

<input type="checkbox"/> chocolate	2000-01-01	
<input type="checkbox"/> macaroon		
<input type="checkbox"/> chupa chups		
<input type="checkbox"/> candy canes	2000-01-05	
<input type="checkbox"/> bon bons		

do zrobienia...  2000-01-01

Search bar:

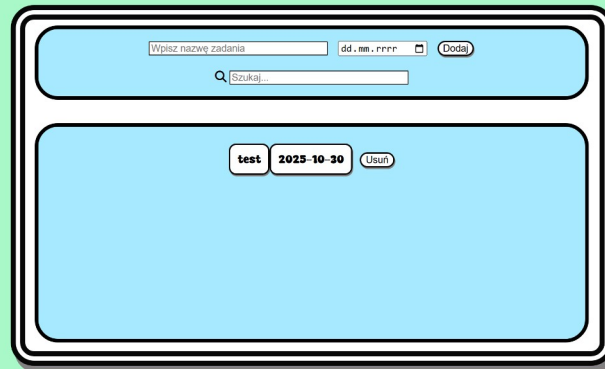
<input type="checkbox"/> macaroon		
<input type="checkbox"/> bon bons		

do zrobienia...  2000-01-01

## 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 😊

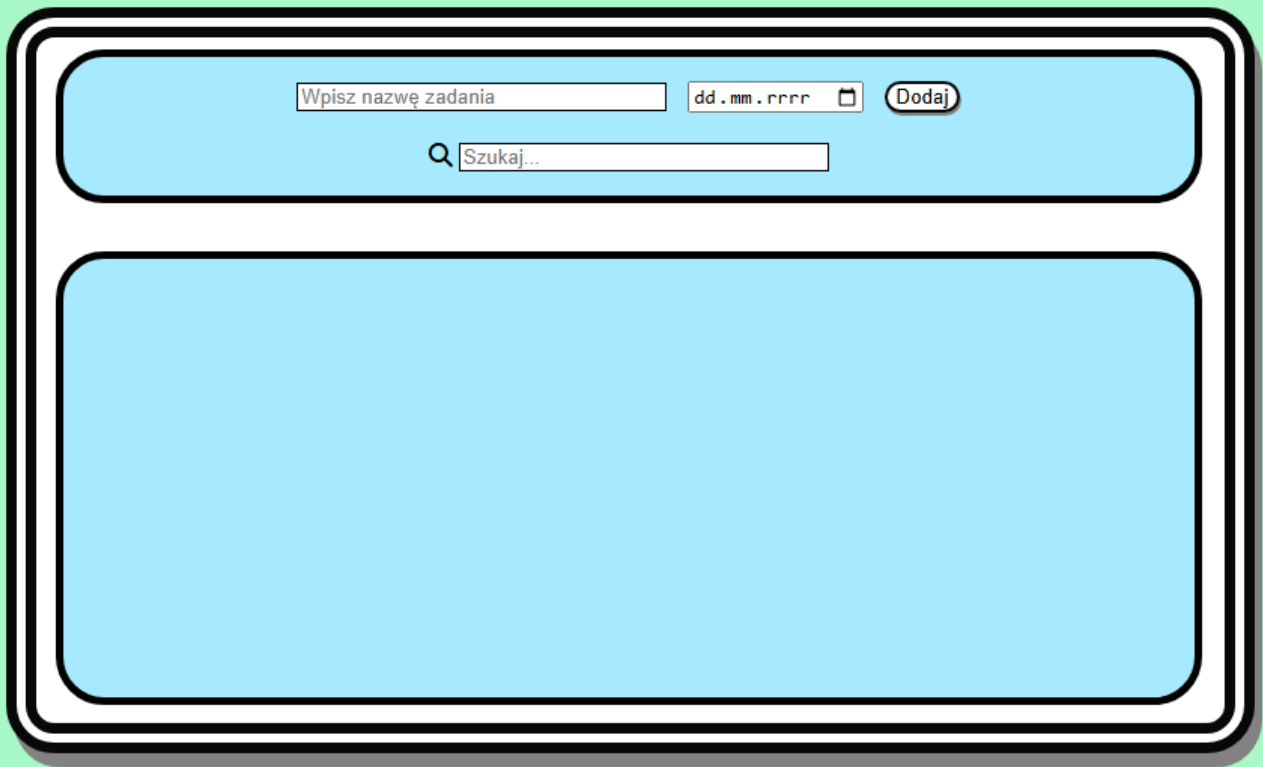
Najlepszym sposobem rozwiązania tego laboratorium jest utworzenie klasy Todo (albo po prostu obiektu z kilkoma metodami). Bez względu na przyjętą strategię, należy w tym nowoutworzonym bycie utworzyć tablicę `tasks` oraz metodę `draw()`, która wyczyści `div` z obecną wizualizacją zadań do zrobienia i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

Wpisz nazwę zadania

dd.mm.rrrr

Dodaj

Q Szukaj...

**zadanie testowe** **2025-10-30** Usuń

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

## USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:

The screenshot shows a web application interface for task management. At the top, there is a light blue header bar with a search input field labeled 'Wpisz nazwę zadania', a date input field labeled 'dd.mm.rrrr' with a calendar icon, and a 'Dodaj' button. Below the header, there is a search bar with a magnifying glass icon and the text 'Szukaj...'. The main content area is a large light blue rectangle. Inside this area, there is a task entry with the text 'zadanie2' and '2025-10-30', followed by a 'Usuń' button.

Wstaw zrzut ekranu listy po usunięciu zadania:

Wpisz nazwę zadania

dd.mm.rrrr

Dodaj

Szukaj...

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

## EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:



The screenshot shows a task management interface with a light blue background. At the top, there is a header bar with a text input field labeled 'Wpisz nazwę zadania', a date input field labeled 'dd.mm.rrrr' with a calendar icon, and a 'Dodaj' button. Below the header is a search bar with a magnifying glass icon and the placeholder text 'Szukaj...'. The main area of the interface is a large light blue rectangle. At the top of this rectangle, there are two buttons: 'zadanie3' and '2025-10-30', followed by a 'Usuń' button.

Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:

październik 2025 ▾

↑

↓

pon	wto	śro	czw	pią	sob	nie
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

[Wyczyść](#) [Dzisiaj](#)

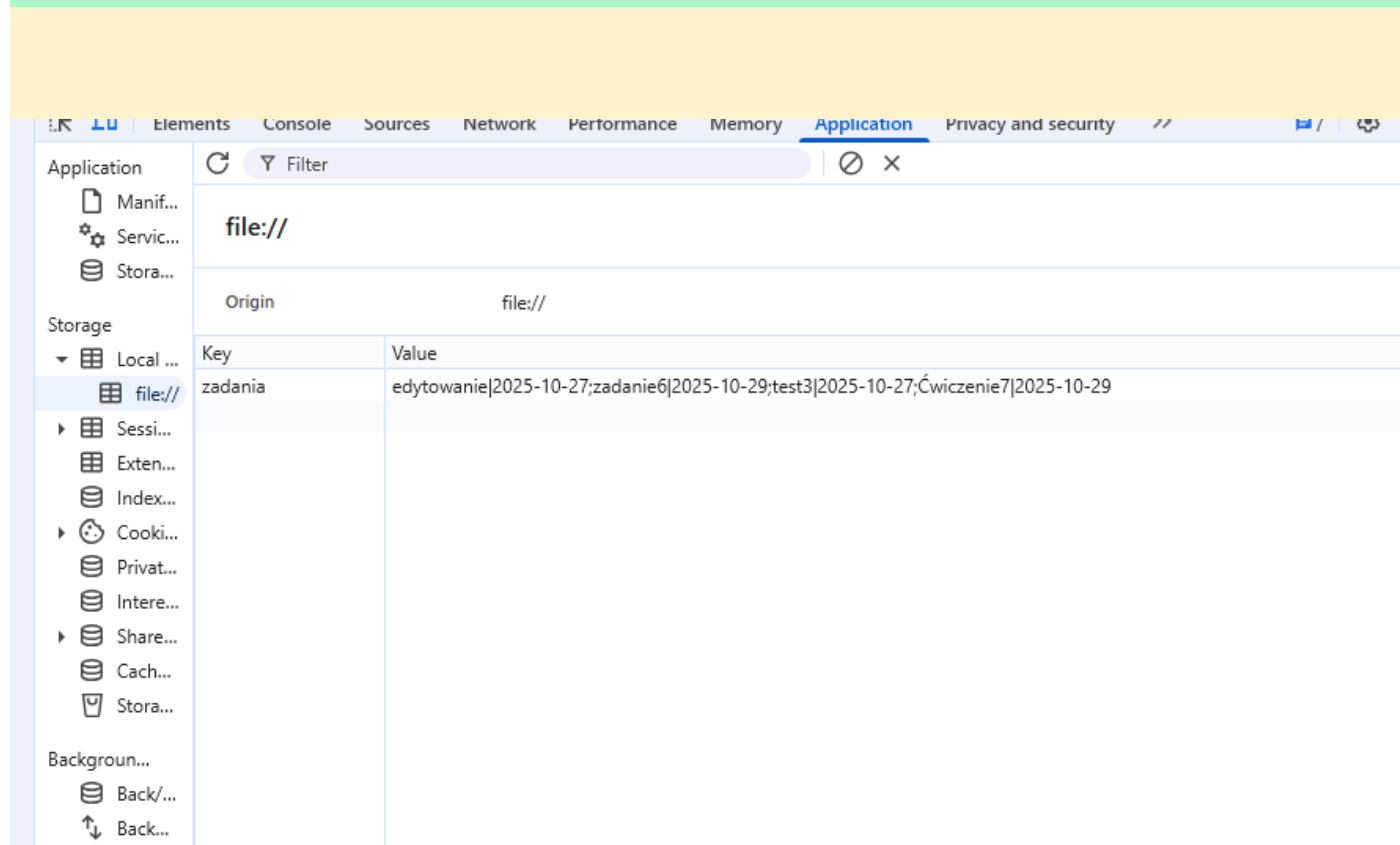
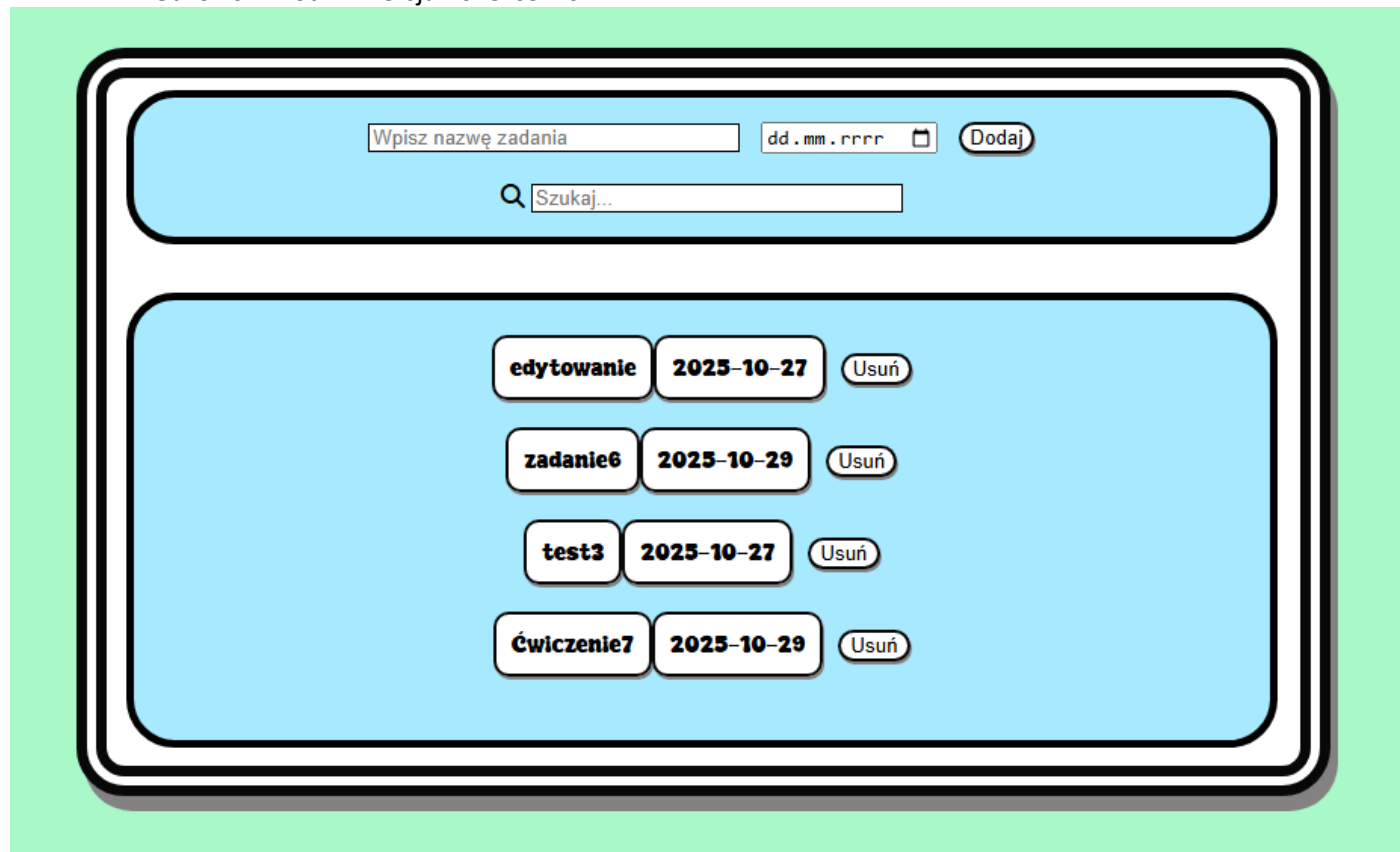
Wstaw zrzut ekranu listy po edycji zadania i daty. Upewnij się, że dane się zapisały i zadanie jest zmienione:

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

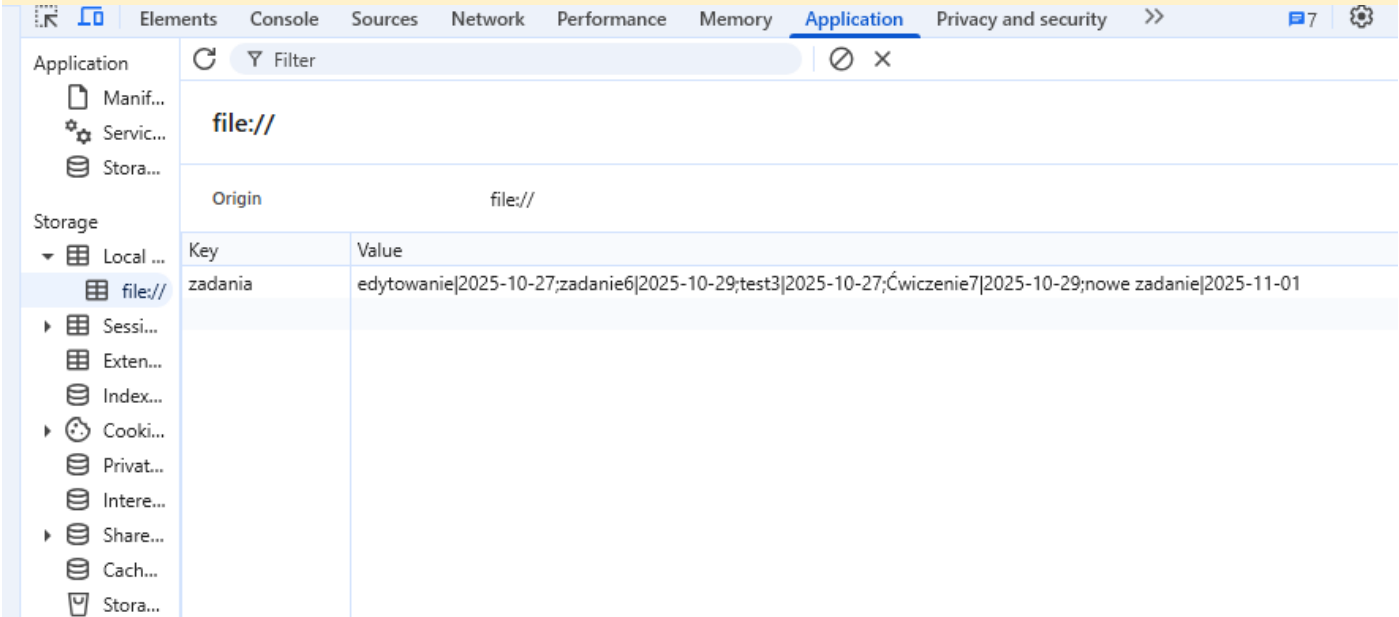
## ODCZYT / ZAPIS LOCALSTORAGE

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:



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:

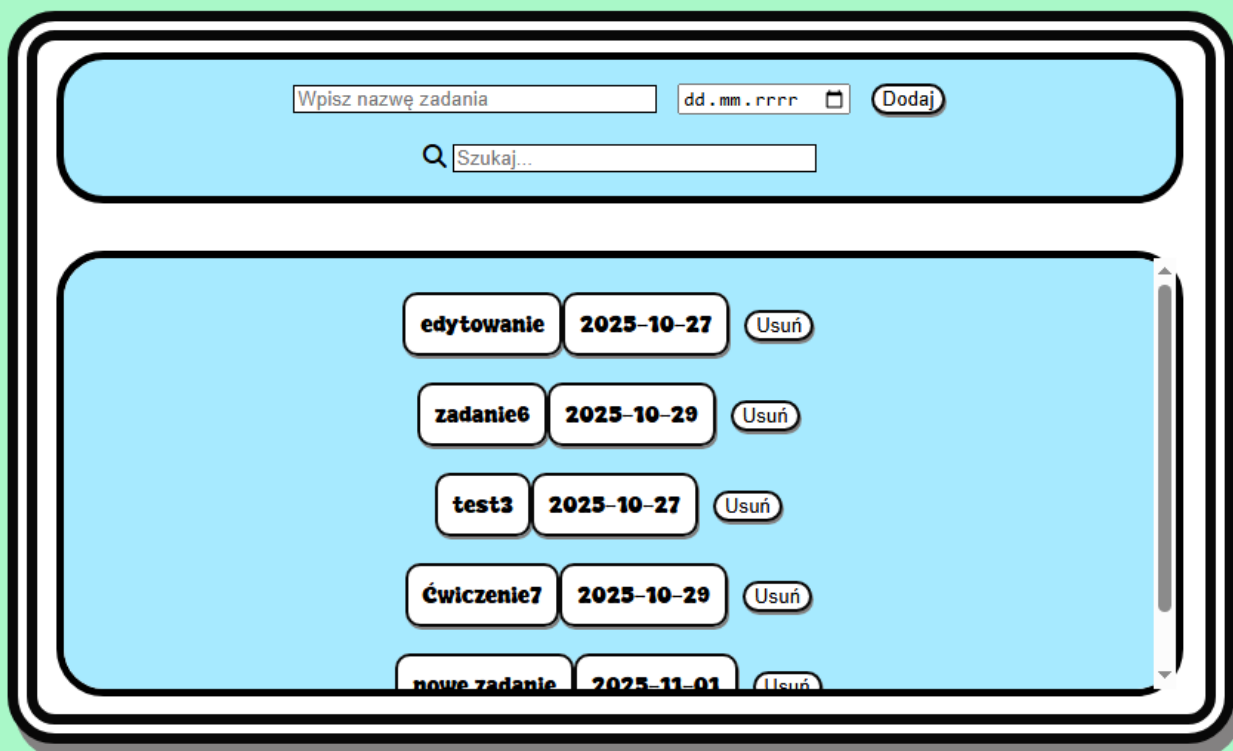


Punkty:	0	1
---------	---	---

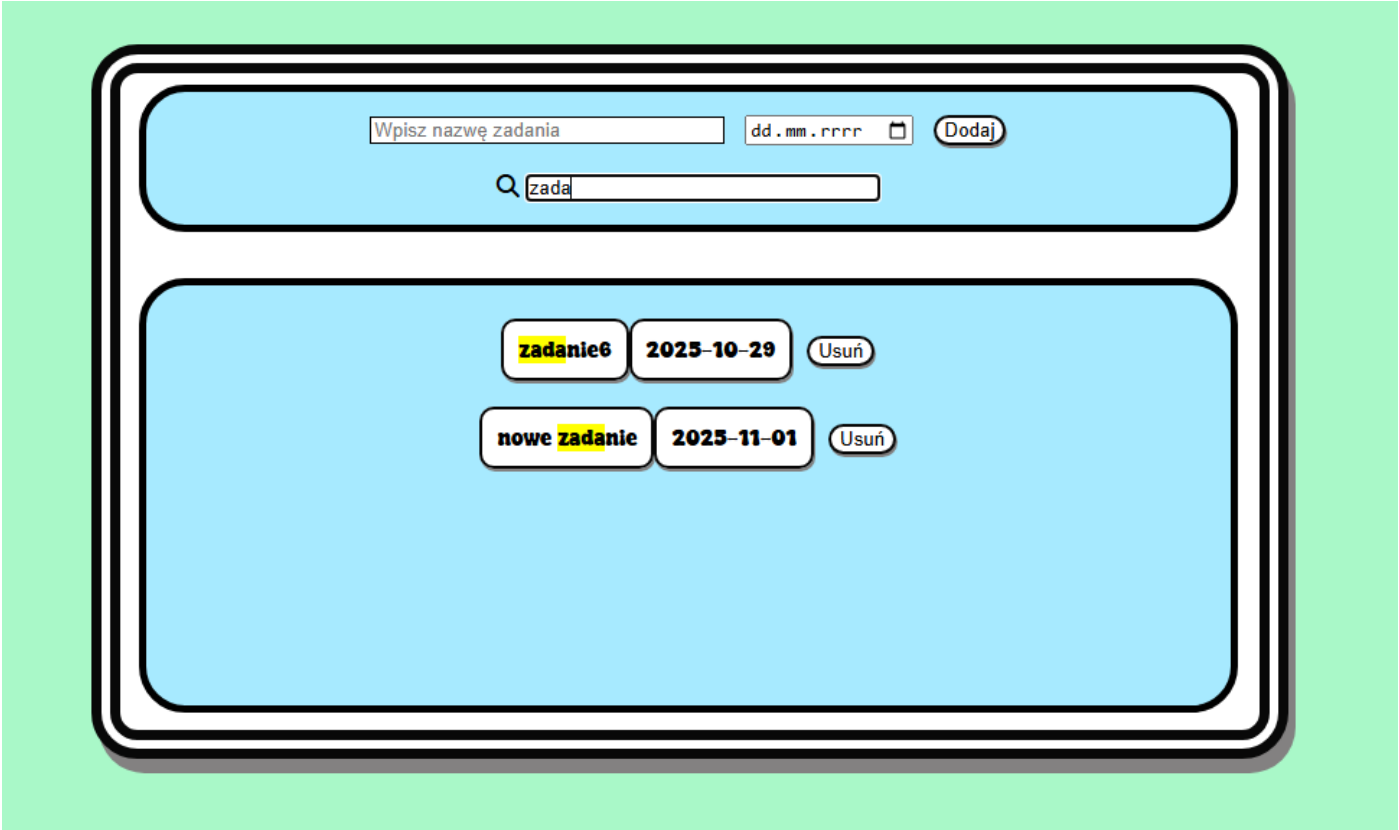
## WYSZUKIWANIE

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:



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**:



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

### COMMIT PROJEKTU DO GIT

Zacommituj 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](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...

