

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	4
Usuwanie pozycji listy	5
Edycja pozycji listy	6
Odczyt / Zapis LocalStorage	7
Wyszukiwanie.....	10
Commit projektu do GIT.....	11
Podsumowanie.....	11

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

Mockupy:

Mockup of the application interface showing a list of tasks. The interface includes a search bar at the top. Below it, a list of tasks is displayed, each with a checkbox, a name, a date, and a delete button. The tasks are: chocolate (2000-01-01), macaroon, chupa chups, candy canes (2000-01-05), and bon bons. At the bottom, there is a text input field labeled 'do zrobienia...', a date picker set to '2000-01-01', and a 'Zapisz' button.

Mockup of the application interface showing a task being edited. The interface is similar to the previous one, but the 'chupa chups' task is selected, and its details are shown in a form below the list. The form includes a text input field with 'chupa chups', a date picker, and a 'Zapisz' button. The date picker is set to '2000-01-01'.

Mockup of the application interface showing search results. The search bar contains the text 'on'. The list of tasks is filtered to show only those containing the search term. The results are: macaroon and bon bons. The date picker at the bottom is set to '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.

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

The screenshot shows a web application interface for a task list. It consists of a search bar at the top, a table with three tasks, and a form at the bottom for adding new tasks.

Szukaj...			
Zadanie 1	2024-10-31	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń

do zrobienia... dd.mm.rrrr

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 😊

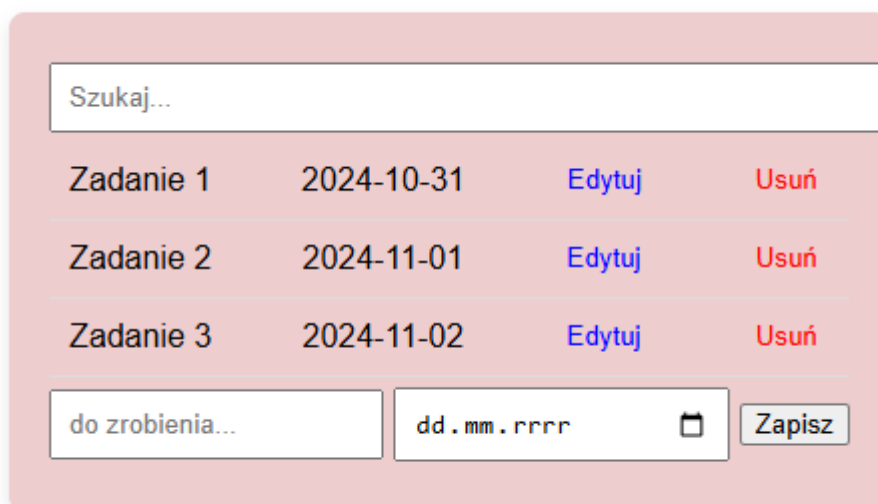
Najlepszym sposobem rozwiązania tego laboratorium jest utworzenie klasy Todo (albo po prostu obiektu z kilkoma metodami). Bez względu na przyjętą strategię, należy w tym nowoutworzonym bycie utworzyć tablicę `tasks` oraz metodę `draw()`, która wyczyści `div` z obecną wizualizacją zadań do zrobienia i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

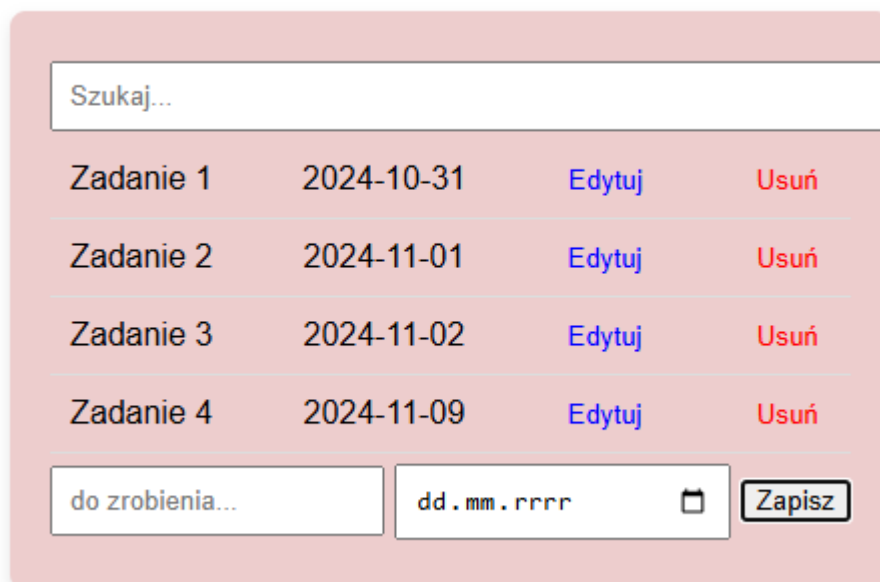


The screenshot shows a task management interface. At the top is a search bar labeled "Szukaj...". Below it is a table with three rows of tasks. Each row contains the task name, a date, and two action buttons: "Edytuj" (Edit) and "Usuń" (Delete). At the bottom of the interface are three input fields: a text field labeled "do zrobienia...", a date field labeled "dd.mm.rrrr" with a calendar icon, and a "Zapisz" (Save) button.

Szukaj...			
Zadanie 1	2024-10-31	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń


do zrobienia... dd.mm.rrrr  Zapisz

Wstaw zrzut ekranu listy po dodaniu nowego zadania:



The screenshot shows the same task management interface as before, but with an additional task added to the list. The table now has four rows. The bottom input fields remain the same.

Szukaj...			
Zadanie 1	2024-10-31	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zadanie 4	2024-11-09	Edytuj	Usuń

do zrobienia... dd.mm.rrrr  Zapisz

Punkty:

0

1

USUWANIE POZYCJI LISTY

Wstaw zrzut ekranu listy przed usunięciem wybranego zadania:

Zadanie 1	2024-10-31	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zadanie 4	2024-11-09	Edytuj	Usuń

Wstaw zrzut ekranu listy po usunięciu zadania:

Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zadanie 4	2024-11-09	Edytuj	Usuń

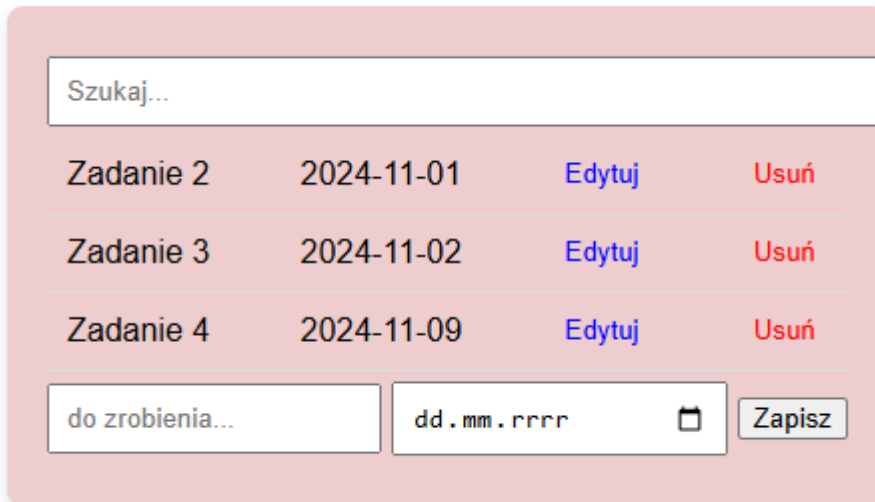
Punkty:

0

1


EDYCJA POZYCJI LISTY

Wstaw zrzut ekranu listy przed edycją wybranego zadania:

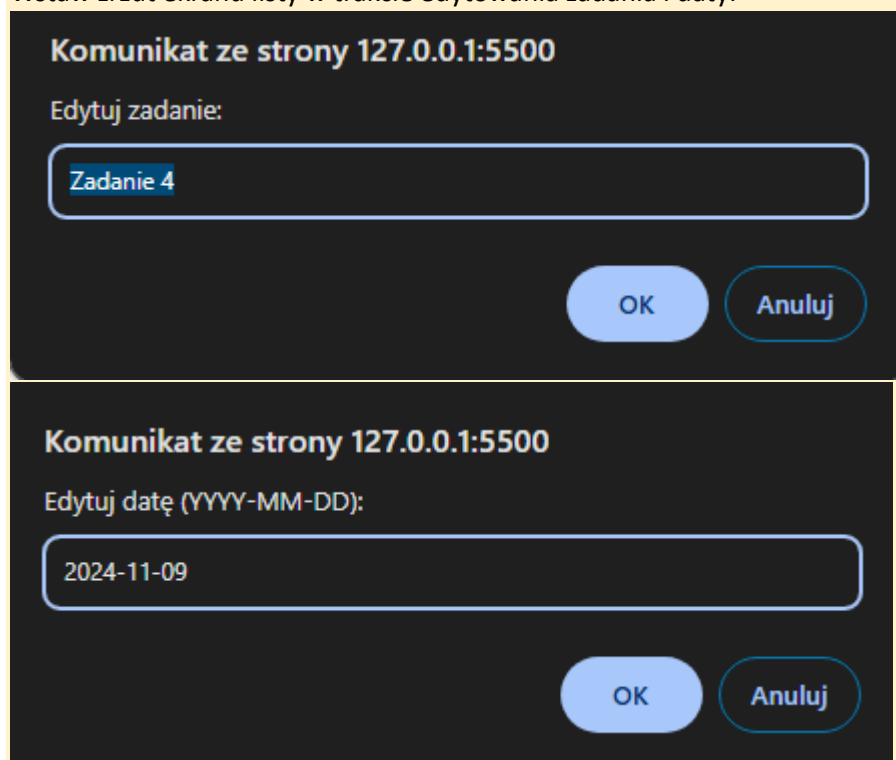


The screenshot shows a task list interface with a search bar at the top labeled "Szukaj...". Below it is a table with three rows of tasks. Each row contains the task name, a date, and two action buttons: "Edytuj" (Edit) and "Usuń" (Delete). At the bottom of the list, there are two input fields: one for a task description labeled "do zrobienia..." and another for a date labeled "dd.mm.rrrr" with a calendar icon. To the right of these fields is a "Zapisz" (Save) button.

Szukaj...			
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zadanie 4	2024-11-09	Edytuj	Usuń

do zrobienia... dd.mm.rrrr  Zapisz

Wstaw zrzut ekranu listy w trakcie edytowania zadania i daty:



The first screenshot shows a dialog box titled "Komunikat ze strony 127.0.0.1:5500". It contains the text "Edytuj zadanie:" followed by a text input field containing "Zadanie 4". At the bottom are two buttons: "OK" and "Anuluj".

The second screenshot shows a similar dialog box titled "Komunikat ze strony 127.0.0.1:5500". It contains the text "Edytuj datę (YYYY-MM-DD):" followed by a text input field containing "2024-11-09". At the bottom are two buttons: "OK" and "Anuluj".

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

Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zmiana kół	2024-11-11	Edytuj	Usuń

Punkty:

0

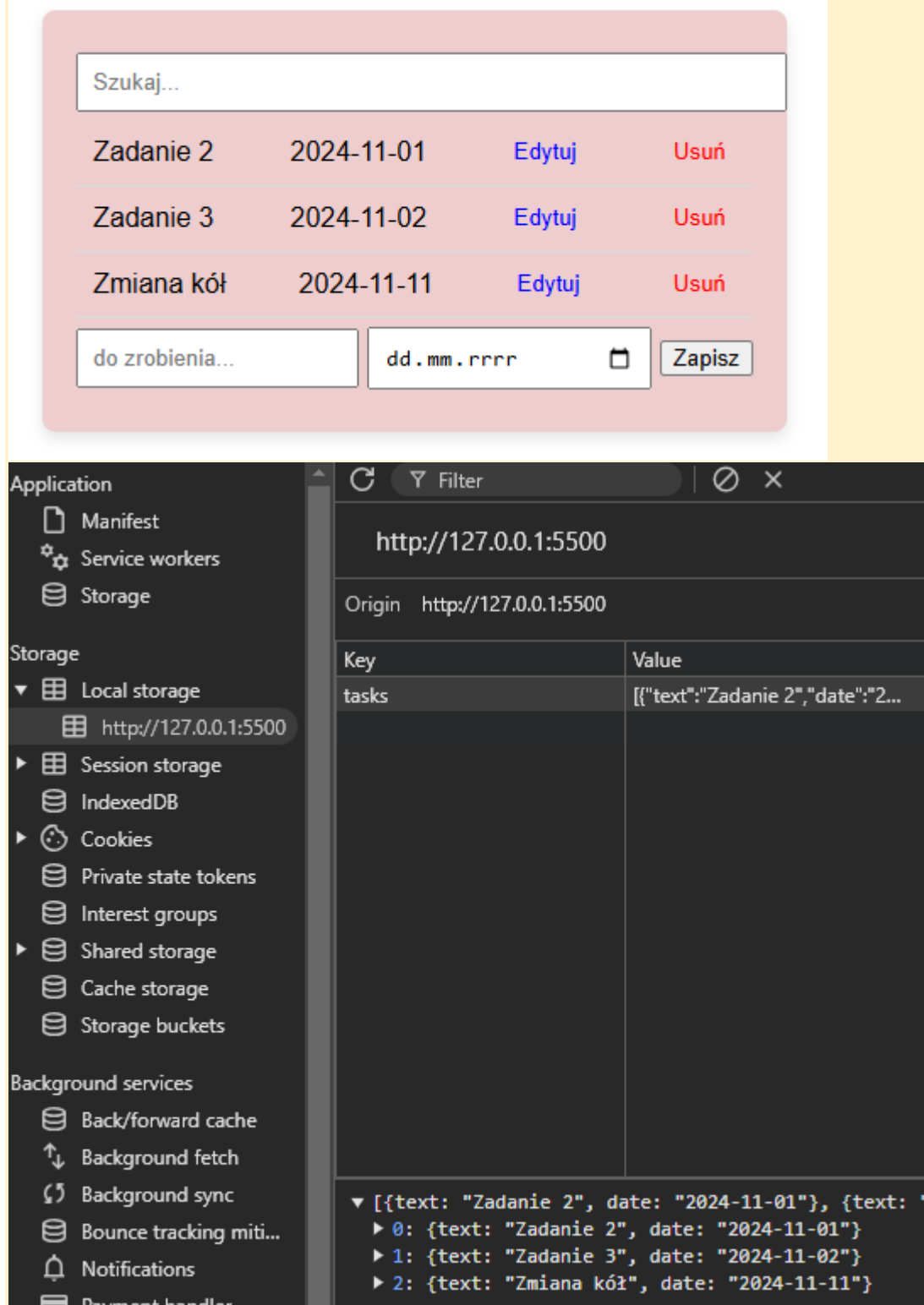
1

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()`.

Zbadaj element -> application -> po lewej local storage

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:

Szukaj...

Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zmiana kół	2024-11-11	Edytuj	Usuń
Mycie okien	2024-11-16	Edytuj	Usuń

do zrobienia...

dd.mm.rrrr

Zapisz

Manifest

Service workers

Storage

Storage

Local storage

http://127.0.0.1:5500

Session storage

IndexedDB

Cookies

Private state tokens

Interest groups

Shared storage

Cache storage

Storage buckets

Background services

Back/forward cache

Background fetch

Background sync

Bounce tracking miti...

Notifications

Payment handler

Periodic background

http://127.0.0.1:5500

Origin http://127.0.0.1:5500

Key	Value
tasks	[{"text":"Zadanie 2","date":"2...

▼ [{"text": "Zadanie 2", date: "2024-11-01"}, {"text": "Zadanie 3", date: "2024-11-02"}, {"text": "Zmiana kół", date: "2024-11-11"}, {"text": "Mycie okien", date: "2024-11-16"}]

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:

The screenshot shows a web application interface for a task list. At the top, there is a search bar with the placeholder text "Szukaj...". Below the search bar is a table of tasks. The table has four columns: task name, due date, edit button, and delete button. The tasks listed are "Zadanie 2" (2024-11-01), "Zadanie 3" (2024-11-02), "Zmiana kół" (2024-11-11), and "Mycie okien" (2024-11-16). Each task has a blue "Edytuj" button and a red "Usuń" button. At the bottom of the form, there is a text input field with the placeholder "do zrobienia...", a date input field with the placeholder "dd.mm.rrrr" and a calendar icon, and a "Zapisz" button.

Zadanie	Data	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń
Zmiana kół	2024-11-11	Edytuj	Usuń
Mycie okien	2024-11-16	Edytuj	Usuń

do zrobienia... dd.mm.rrrr Zapisz

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 the same web application interface as before, but the search bar now contains the text "dan". The table of tasks is filtered to show only two results: "Zadanie 2" (2024-11-01) and "Zadanie 3" (2024-11-02). The "Zmiana kół" and "Mycie okien" tasks are no longer visible. The bottom of the form remains the same with the "do zrobienia..." input, the date input, and the "Zapisz" button.

Zadanie	Data	Edytuj	Usuń
Zadanie 2	2024-11-01	Edytuj	Usuń
Zadanie 3	2024-11-02	Edytuj	Usuń

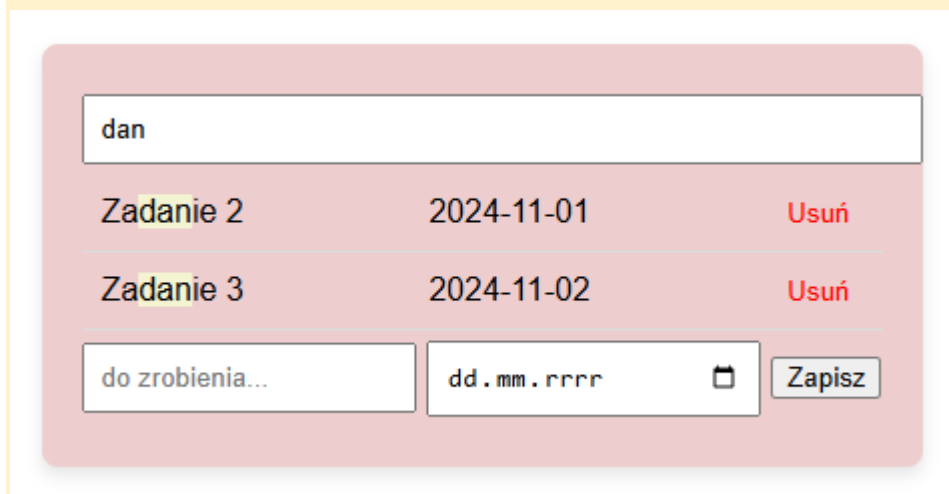
do zrobienia... dd.mm.rrrr Zapisz

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 web application interface. At the top, there is a search bar containing the text 'dan'. Below the search bar, there is a table with two rows of task information. The first row shows 'Zadanie 2' with a date of '2024-11-01' and a red 'Usuń' button. The second row shows 'Zadanie 3' with a date of '2024-11-02' and a red 'Usuń' button. Below the table, there is a form with a text input field containing 'do zrobienia...', a date picker showing 'dd.mm.yyyy', and a 'Zapisz' 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/Xomil/Aplikacje-Internetowe-1/tree/main/lab-b>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Nauczyłem się obsługi formularzy oraz wstawiania html poprzez js.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.