

To-Do List

Projekt Technologie Internetowe
JavaScript

Technologie Internetowe

Jakub Jakubowski

125125, lab1

II rok informatyki stacjonarnie

26.01.2024 – v1.0

Cel aplikacji

Projekt To-Do List ma na celu stworzenie szablonu strony do dodawania codziennych zadań do swojej listy wykorzystując JavaScript. Strona będzie umożliwiała użytkownikowi dodawanie, usuwanie i edycję zadań które ma wykonać w ciągu dnia bieżącego, tak aby już nigdy więcej nie zapomnieć o rzeczach ważnych czy mniej ważnych do wykonania w ciągu dnia.

Projekt będzie obsługiwał formularz, który umożliwi użytkownikowi wprowadzanie nowych zadań na stronie internetowej. Dodane zadania są reprezentowane jako dynamiczne elementy HTML, dzięki wykorzystaniu języka JavaScript, co pozwala na bieżącą aktualizację widoku strony bez konieczności odświeżania.

Problem do realizacji

Rozwiązanie problemów poprawi użyteczność i poprawność aplikacji. Korzystanie z witryny zostanie dzięki temu znacznie ułatwione na różnych typach urządzeń, czy to dekstopowych czy mobilnych. W ten sposób, rozwiązując wszystkie problemy poprawimy ogólną satysfakcję użytkowników. W tym celu należy rozwiązać następujące problemy:

- Przycisk „Add Task” w skrajnych warunkach, gdzie wyświetlacz jest bardzo minimalistyczny, ciężko jest zatwierdzić dodanie nowego zadania
- Karuzela zdjęć z sekcji „About Me”, nie może być przewijana za pomocą przycisków, nie można dostosować standardowego podejścia animacji do tego problemu, zachowując walory estetyczne, prawidłowo działającej karuzeli
- Lista zadań wraz z odświeżeniem strony staje się pusta, zadania znikają wraz z zamknięciem sesji
- Zadania pomimo udanej walidacji nie wyświetlają się na liście zadań. Nazwa wpisanego zadania pojawia się w linku przeglądarki

Sposób rozwiązania

- Funkcje języka JavaScript umożliwiły dodanie zadania w przypadkach kiedy użytkownik uruchomi witrynę na skrajnie małym wyświetlaczu np. smartwatch. Wykorzystując funkcję EventListener umożliwiły zatwierdzenie nowych zadań poprzez naciśnięcie klawiszy funkcjonalnych na urządzeniach mobilnych a w przypadku urządzeń desktopowych jest to klawisz „Enter”.
- Karuzela posiada przyciski oraz zdjęcia które zostały dodane dynamicznie dzięki DOM i są w pełni obsługiwane.
- Wykorzystanie funkcji JSON oraz Local Storage umożliwiły zapis sesji użytkownika. Dodając lub usuwając zadania, wykonuje to lokalnie, więc mimo odświeżenia witryny, zadania dalej wyświetlają się i użytkownik ich nie traci, o ile ich ręcznie nie usunął.
- Przycisk „Add Task” ma format submit, więc zadania były przekazywane a nie wyświetlane. Dzięki zastosowaniu JavaScript a dokładniej funkcji preventDefault, zadania wyświetlają się poprawnie w liście zadań a nie nadpisują link w przeglądarce.

Sposób Wykonania

Strona internetowa będzie zawierała karuzelę zdjęć dla lepszej obsługi sekcji „o mnie”, dzięki temu zostaną wykorzystane animacje za pomocą języka JavaScript.

Aplikacja składa się z pliku app.js który składa się z:

1) "TaskManager"

- Inicjalizacja klasy pozwala na odczyt zadań z magazynu lokalnego, używając JSON, zapewniając, że zadania pozostaną po odświeżeniu strony.
- Dodawanie nowego zadania po wprowadzeniu go w polu tekstowym i zatwierdzeniu przyciskiem "Add Task".
- Sprawdzanie walidacji poprawności wprowadzonego zadania, wyświetlanie go i zapisywanie w lokalnym magazynie.
- Metoda Render Task generuje dynamiczny HTML dla każdego zadania z listy.
- Nasłuchiwanie dla zadań, umożliwiające ich usunięcie lub edycję treści.

2) Funkcja "updateCarousel":

- Aktualizuje zawartość karuzeli ze zdjęciami na stronie.

- Dodaje obsługę zdarzeń przycisków dla poprzedniego i następnego zdjęcia.

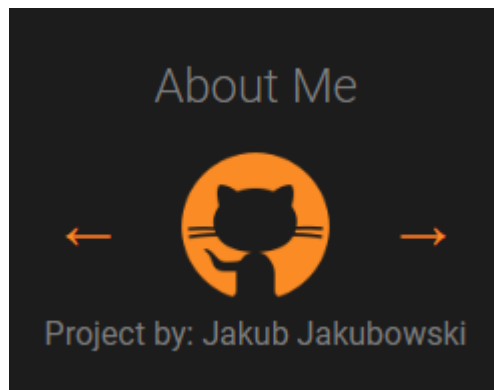
3) EventListener:

- Dla pól tekstowych umożliwia zatwierdzenie działania za pomocą klawiatury na urządzeniach desktopowych lub mobilnych.

Testowanie

1) Testy Wydajności Karuzeli Zdjęć:

- a. Przeprowadzono testy wydajności dla karuzeli zdjęć w sekcji "O mnie".
- b. Oceniono płynność animacji, czas ładowania i karuzeli na różnych urządzeniach.

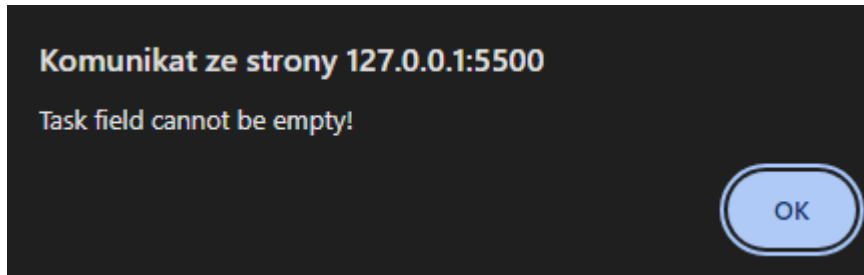


2) Testy walidacji formularza dodającego zadania

- a. Przeprowadzono testy dla których pole tekstowe na wpisanie zadania jest puste, użytkownik nie wprowadzi żadnego znaku i zatwierdzi dodaniem zadania.

Zatwierdzenie pustego pola:

Komunikat:

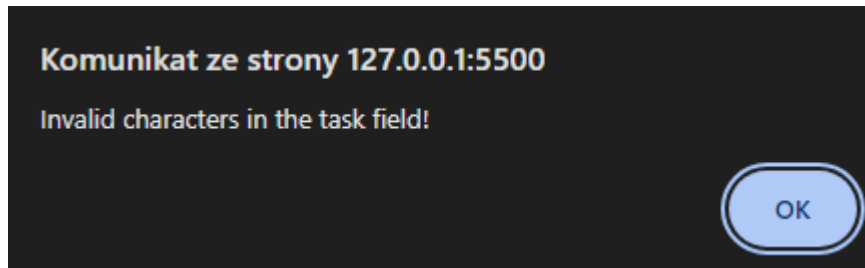


- b. Wykonano test dla pola tekstowego, w którym wprowadzone zostały znaki specjalne, które nie powinny się znajdować w polu zadania. Użytkownik wprowadza znaki specjalne i zatwierdza dodanie zadania

Przykład wprowadzenia znaków specjalnych



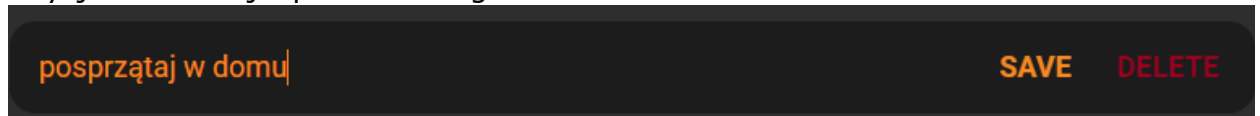
Komunikat:



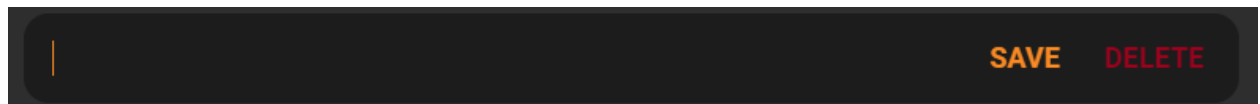
3) Testy walidacji listy zadań

- a. Testowanie edycji zadań dla zmiany poprawnie wprowadzonego zadania, na próbę zatwierdzenia edycji zostawiając puste pole tekstowe, bez znaków.

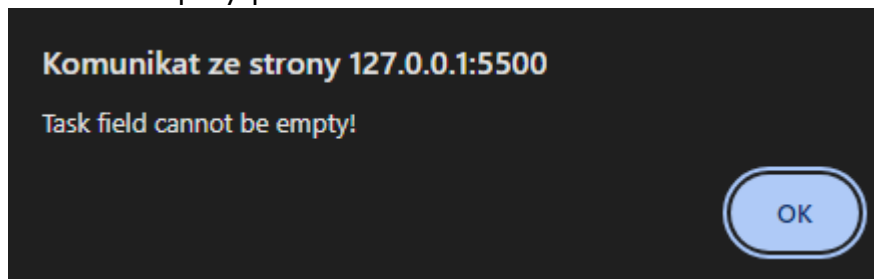
Edycja wcześniej wprowadzonego zadania:



Zamiana na:

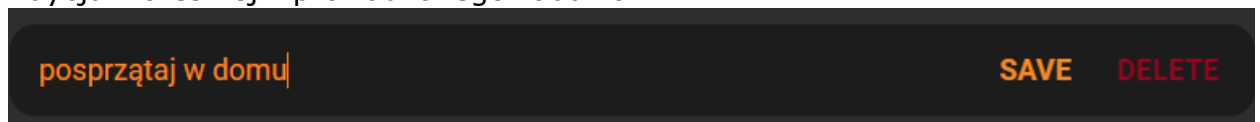


Komunikat przy próbie zatwierdzenia zmian:

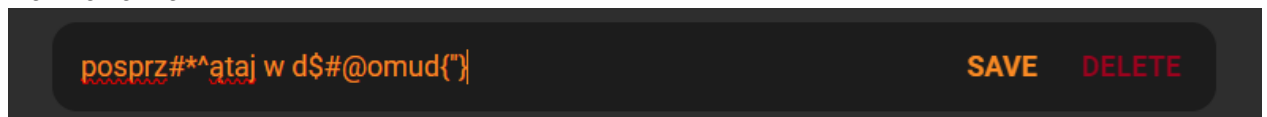


- b. Przeprowadzono testowanie edycji zadań dla zmiany poprawnie wprowadzonego zadania, na zmianę bądź wprowadzenie znaków specjalnych, a następnie zatwierdzenie zmian

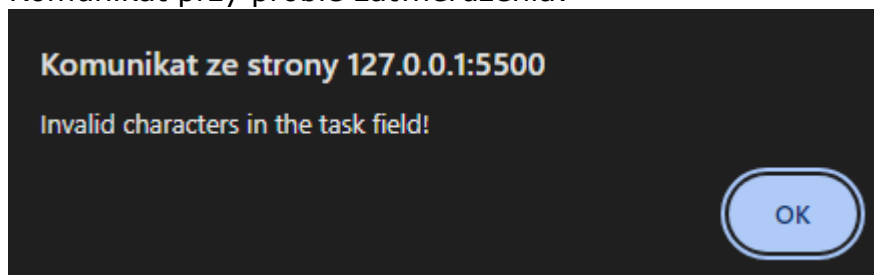
Edycja wcześniej wprowadzonego zadania:



Zamiana na:



Komunikat przy próbie zatwierdzenia:



Dokumentacja JDocs

Dokumentacja znajduje się pod tym [linkiem](#).

Linki

Projekt To-Do List korzysta z źródeł zewnętrznych, dokładniej do grafiki, ikon. Wszystkie użyte zasoby są zgodne z prawem autorskim, a ich źródła są udokumentowane poniżej.

- Ikony: <https://icons8.com/icons>