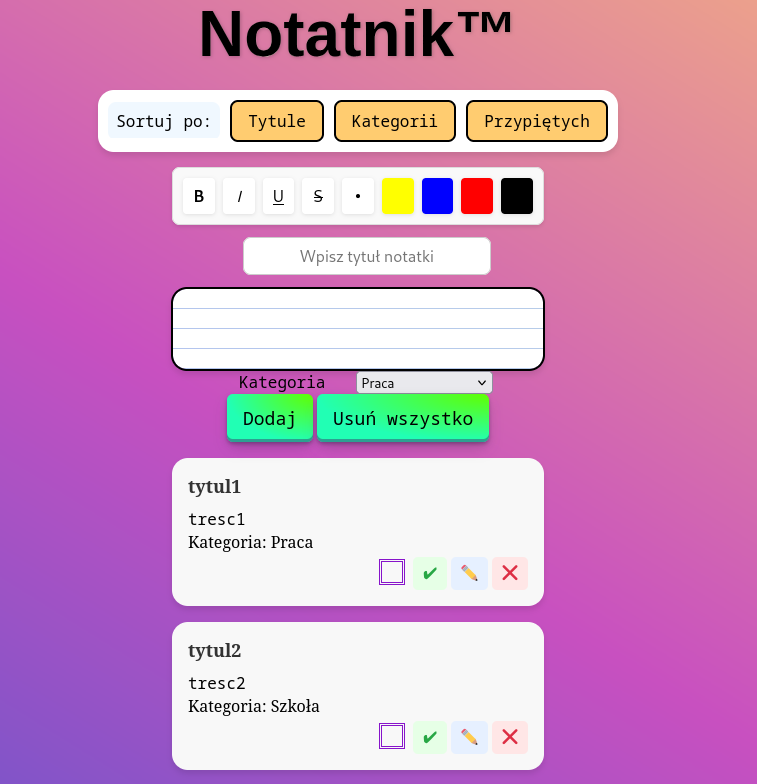
# **Dokumentacja aplikacji "Notatnik"**

## **Opis projektu**

"Notatnik" to prosta aplikacja internetowa, która umożliwia użytkownikowi tworzenie, edytowanie, formatowanie oraz zarządzanie notatkami. Aplikacja została stworzona z wykorzystaniem technologii HTML, CSS oraz JavaScript.

## **Zawartość plików**

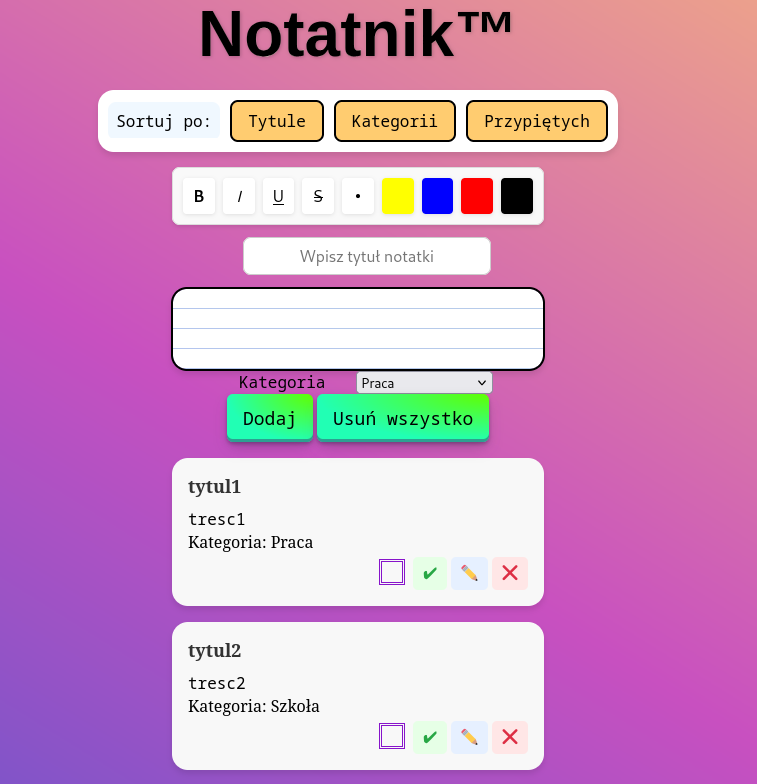
## **Plik: index.html**



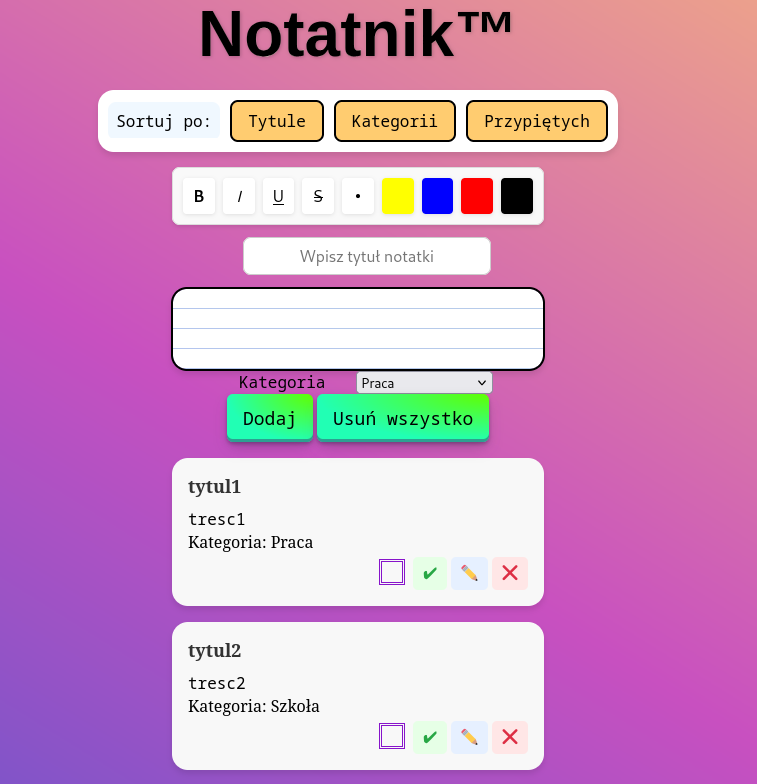
Ten plik zawiera strukturę HTML aplikacji, definiując układ interfejsu użytkownika oraz elementy interaktywne.

#### **Najważniejsze sekcje:**

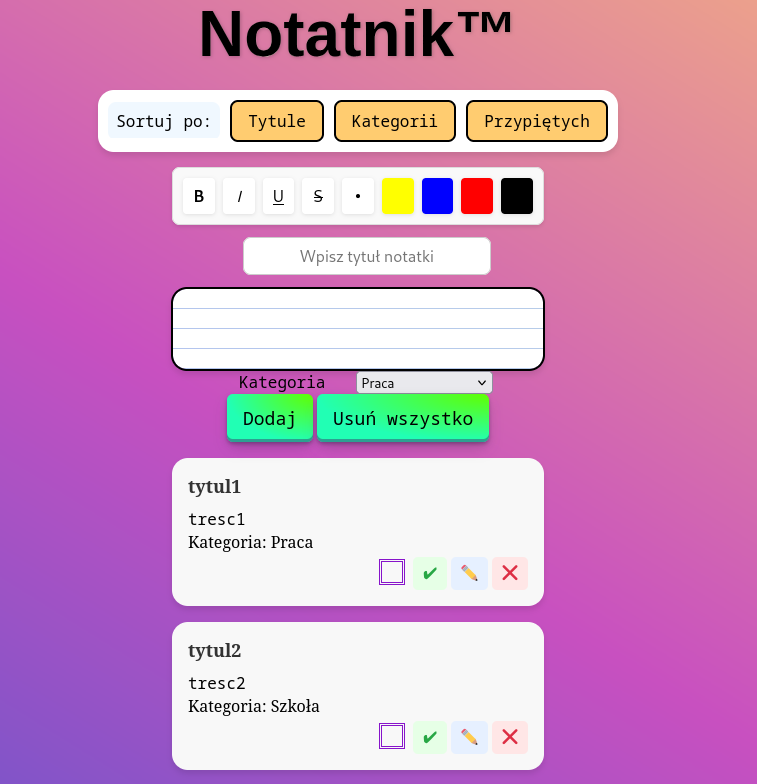
1. **Nagłówek strony**
   * Tytuł aplikacji: "Notatnik™"



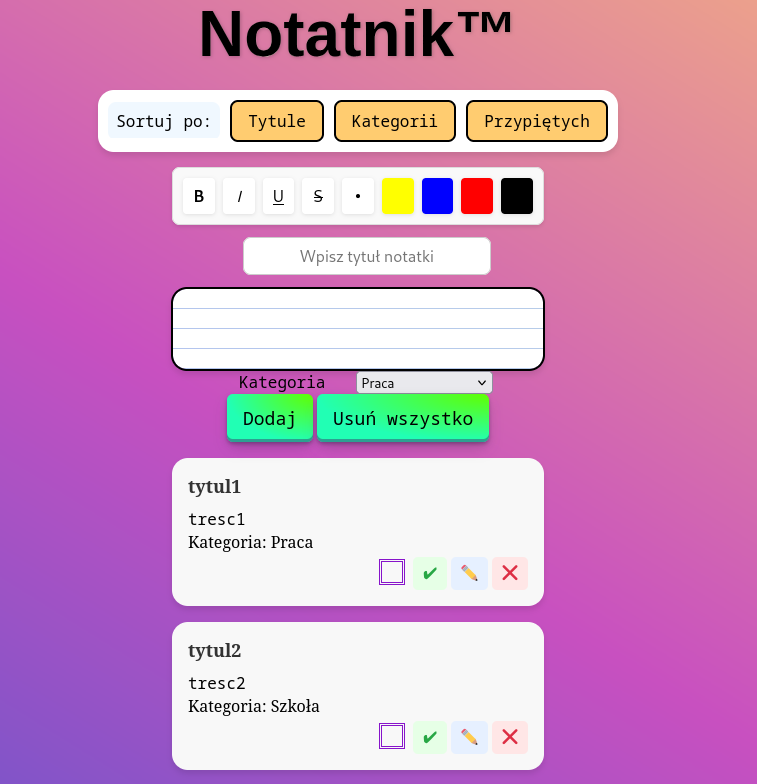
1. **Sortowanie notatek**
   * Pasek sortowania notatek według kryteriów: tytuł, kategoria, przypięte notatki.



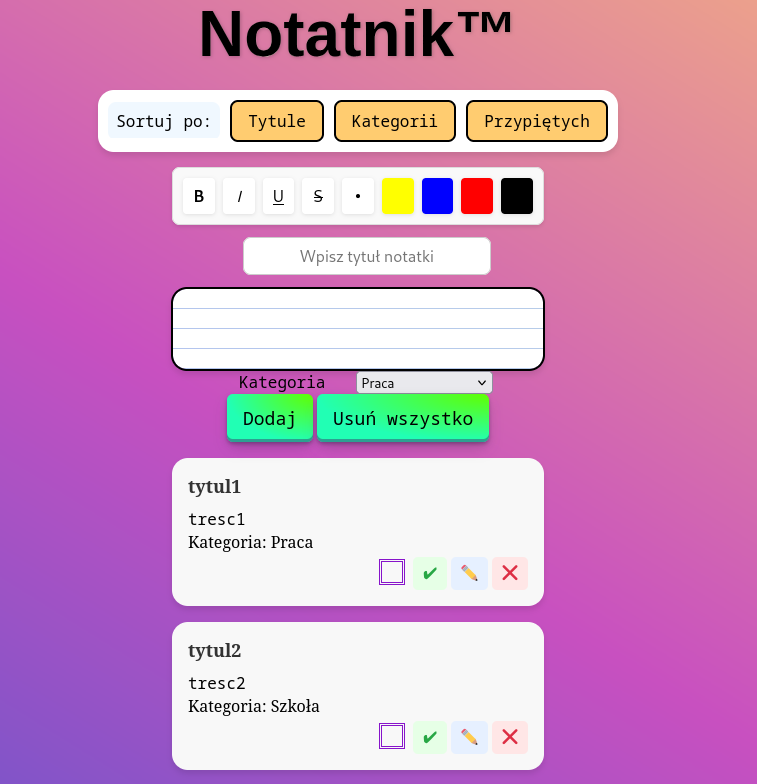
1. **Pasek narzędzi do formatowania tekstu**
   * Przyciski umożliwiające formatowanie tekstu: pogrubienie, kursywa, podkreślenie, przekreślenie, wypunktowanie oraz zmiana koloru tekstu.



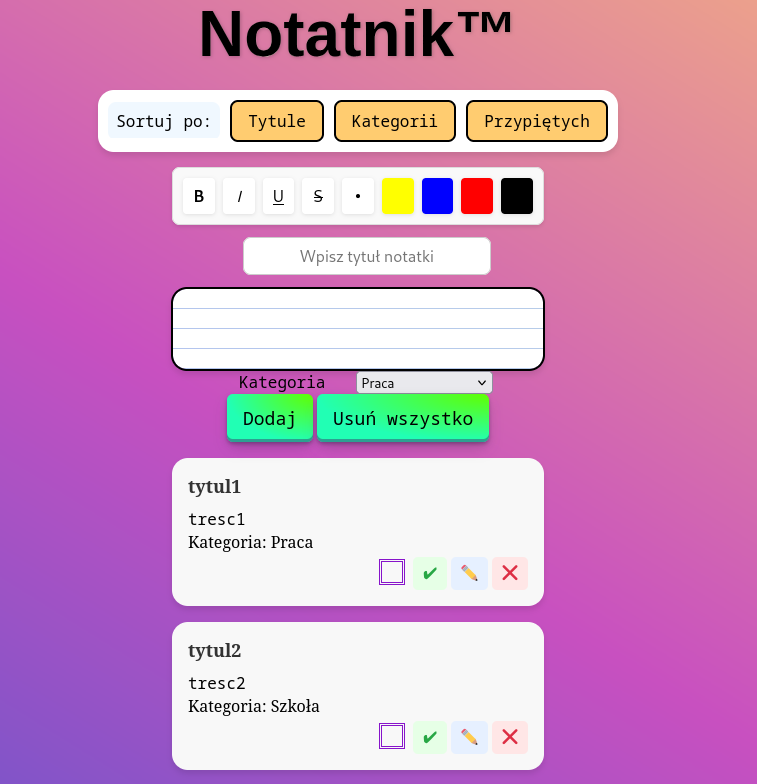
1. **Formularz notatki**
   * Pole tekstowe do wprowadzenia tytułu notatki.
   * Główne pole edycyjne z możliwością formatowania treści.
   * Lista rozwijana do wyboru kategorii notatki (z możliwością dodania własnej kategorii).



1. **Przyciski zarządzania notatkami**
   * "Dodaj" - dodaje nową notatkę lub aktualizuje edytowaną.
   * "Usuń wszystko" - usuwa wszystkie zapisane notatki.



1. **Lista zapisanych notatek**
   * Miejsce, w którym wyświetlane są wszystkie utworzone notatki.

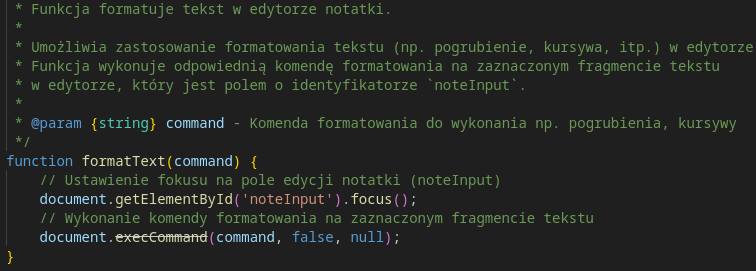


### **Plik: script.js**

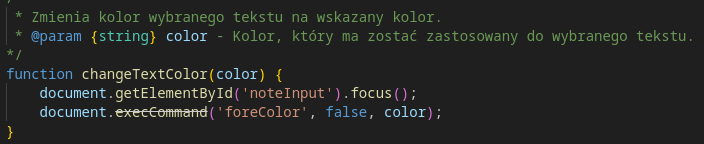
Plik JavaScript odpowiada za interaktywność aplikacji oraz logikę zarządzania notatkami.

#### **Najważniejsze funkcje:**

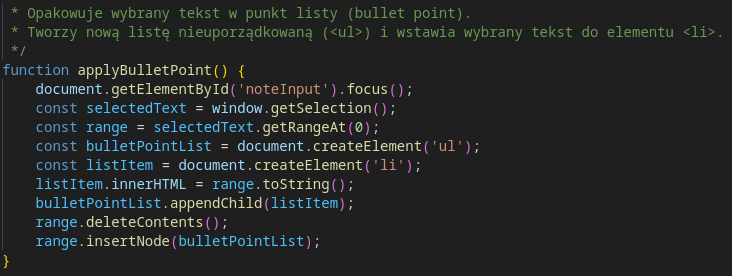
1. **Funkcje formatowania tekstu**
   * formatText(command) - Umożliwia stosowanie stylów (pogrubienie, kursywa itp.) na zaznaczonym tekście.



* + changeTextColor(color) - Zmienia kolor zaznaczonego tekstu.



* + applyBulletPoint() - Dodaje wypunktowanie do zaznaczonego tekstu.



1. **Zarządzanie notatkami**
   * addOrUpdateNote() - Dodaje nową notatkę lub aktualizuje istniejącą.

function addOrUpdateNote() {

// Pobranie wartości tytułu notatki z pola tekstowego

const noteTitle = document.getElementById('noteTitle').value.trim(); // Ładowanie tytułu

// Pobranie zawartości notatki z edytowalnego pola

const noteInput = document.getElementById('noteInput').innerHTML;

// Pobranie wybranej kategorii z listy rozwijanej

const categorySelect = document.getElementById('categorySelect');

const customCategoryInput = document.getElementById('customCategoryInput');

// Ustalenie kategorii notatki: jeśli użytkownik wybrał 'custom', używamy wartości z pola tekstowego

let category = categorySelect.value === 'custom' ? customCategoryInput.value.trim() : categorySelect.value;

// Sprawdzenie, czy zawartość notatki jest pusta

if (!noteInput) {

alert("Nie wpisano tekstu notatki.");

return;

}

// Sprawdzenie, czy wybrano kategorię

if (!category) {

alert("Proszę wybrać kategorię przed zapisaniem notatki.");

return;

}

// Sprawdzenie, czy tytuł notatki jest pusty

if (!noteTitle) {

alert("Proszę wpisać tytuł notatki.");

return;

}

// Pobranie listy notatek, do której dodamy nową notatkę lub zaktualizujemy istniejącą

const noteList = document.getElementById('noteList');

// Sprawdzenie, czy edytujemy istniejącą notatkę

if (editingNoteCard) {

// Zaktualizowanie tytułu, treści i kategorii istniejącej notatki

const noteCardTitle = editingNoteCard.querySelector('.note-card-title');

const noteCardContent = editingNoteCard.querySelector('.note-card-content');

const noteCardCategory = editingNoteCard.querySelector('.note-card-category');

// Ustawienie nowych wartości dla tytułu, treści i kategorii notatki

noteCardTitle.textContent = noteTitle; // Ładowanie tytułu

noteCardContent.innerHTML = noteInput;

noteCardCategory.textContent = `Kategoria: ${category}`;

// Po zakończeniu edycji, resetowanie zmiennej `editingNoteCard`

editingNoteCard = null;

} else {

// Tworzenie nowej notatki, jeśli nie edytujemy istniejącej

const noteCard = document.createElement('div');

noteCard.classList.add('note-card');

// Tworzenie i ustawianie tytułu nowej notatki

const noteCardTitle = document.createElement('div');

noteCardTitle.classList.add('note-card-title');

noteCardTitle.textContent = noteTitle; // Ładowanie tytułu

// Tworzenie i ustawianie treści nowej notatki

const noteCardContent = document.createElement('div');

noteCardContent.classList.add('note-card-content');

noteCardContent.innerHTML = noteInput;

// Tworzenie i ustawianie kategorii nowej notatki

const noteCardCategory = document.createElement('div');

noteCardCategory.classList.add('note-card-category');

noteCardCategory.textContent = `Kategoria: ${category}`;

// Tworzenie kontenera akcji dla notatki

const actionsContainer = createActionsContainer(noteCard);

// Dodanie wszystkich elementów (tytuł, treść, kategoria, akcje) do nowej notatki

noteCard.appendChild(noteCardTitle); // Ładowanie tytułu

noteCard.appendChild(noteCardContent);

noteCard.appendChild(noteCardCategory);

noteCard.appendChild(actionsContainer);

// Dodanie nowej notatki do listy notatek

noteList.appendChild(noteCard);

}

// Resetowanie formularza: czyszczenie pól tytułu, treści, kategorii i ustawienie domyślnej wartości kategorii

document.getElementById('noteTitle').value = '';

document.getElementById('noteInput').innerHTML = '';

customCategoryInput.value = '';

categorySelect.value = 'Praca';

customCategoryInput.style.display = 'none';

// Zapisanie notatek (np. w localStorage lub w innym miejscu)

saveNotes();

// Dostosowanie wysokości pola edycji notatki, aby pasowała do zawartości

adjustTextareaHeight(document.getElementById('noteInput'));

}

* + editNote(noteCard) - Przenosi dane z wybranej notatki do formularza edycji, umożliwiając ich modyfikację.

function editNote(noteCard) {

// Ustawienie zmiennej `editingNoteCard`, która przechowuje odniesienie do edytowanej notatki

editingNoteCard = noteCard;

// Pobranie tytułu notatki z elementu HTML i przypisanie go do pola edycji

const noteTitle = noteCard.querySelector('.note-card-title').textContent;

// Pobranie zawartości notatki (z zachowaniem linii przerwy) i przypisanie jej do pola edycji

const noteContent = noteCard.querySelector('.note-card-content').innerHTML.replace(/<br>/g, '\n'); // Zamiana <br> na \n

// Pobranie kategorii notatki i usunięcie prefiksu "Kategoria: "

const noteCategory = noteCard.querySelector('.note-card-category').textContent.replace('Kategoria: ', '');

// Ustawienie tytułu i zawartości notatki w formularzu edycji

document.getElementById('noteTitle').value = noteTitle; // Ustawienie tytułu w polu tekstowym

document.getElementById('noteInput').innerHTML = noteContent.replace(/\n/g, '<br>'); // Ustawienie treści w polu edycji (z powrotem zamiana \n na <br>)

// Pobranie elementów formularza kategorii

const categorySelect = document.getElementById('categorySelect');

const customCategoryInput = document.getElementById('customCategoryInput');

// Sprawdzenie, czy kategoria jest standardowa, czy niestandardowa

if (['Praca', 'Szkoła', 'Zakupy', 'Do zrobienia'].includes(noteCategory)) {

// Jeśli kategoria jest standardowa, ustawić odpowiednią kategorię w formularzu i ukryć pole dla niestandardowej kategorii

categorySelect.value = noteCategory;

customCategoryInput.style.display = 'none';

} else {

// Jeśli kategoria jest niestandardowa, ustawić kategorię na 'custom' i pokazać pole do wprowadzenia własnej kategorii

categorySelect.value = 'custom';

customCategoryInput.value = noteCategory;

customCategoryInput.style.display = 'block';

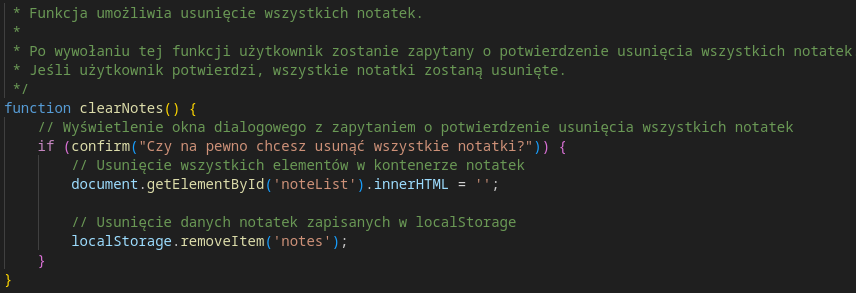
}

// Dostosowanie wysokości pola edycji notatki, aby pasowała do zawartości

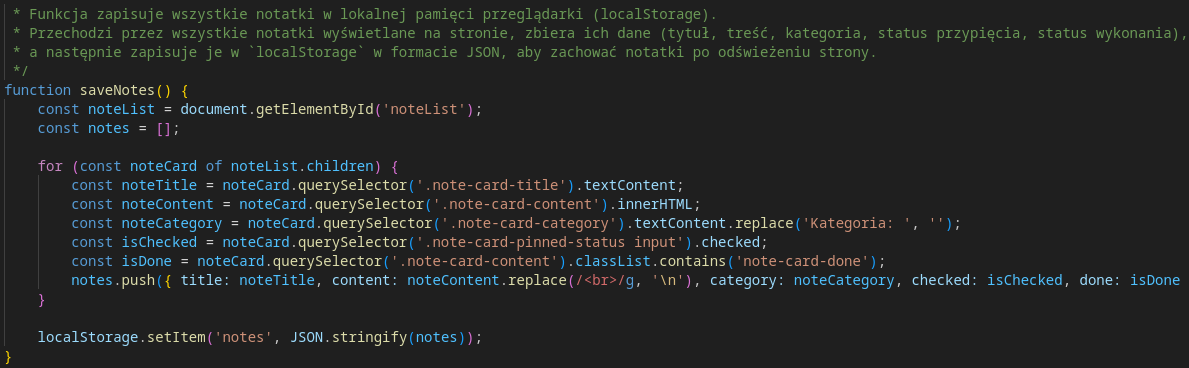
adjustTextareaHeight(document.getElementById('noteInput'));

}

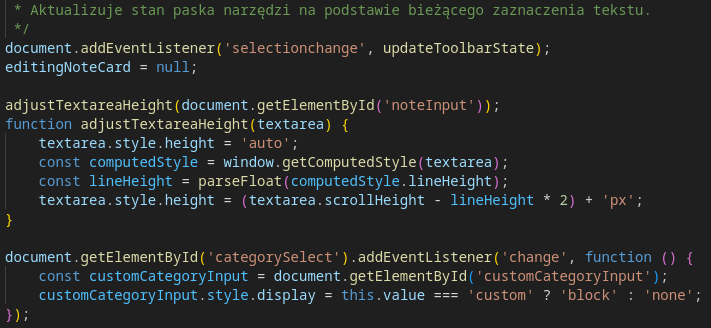
* + clearNotes() - Usuwa wszystkie zapisane notatki po potwierdzeniu przez użytkownika.



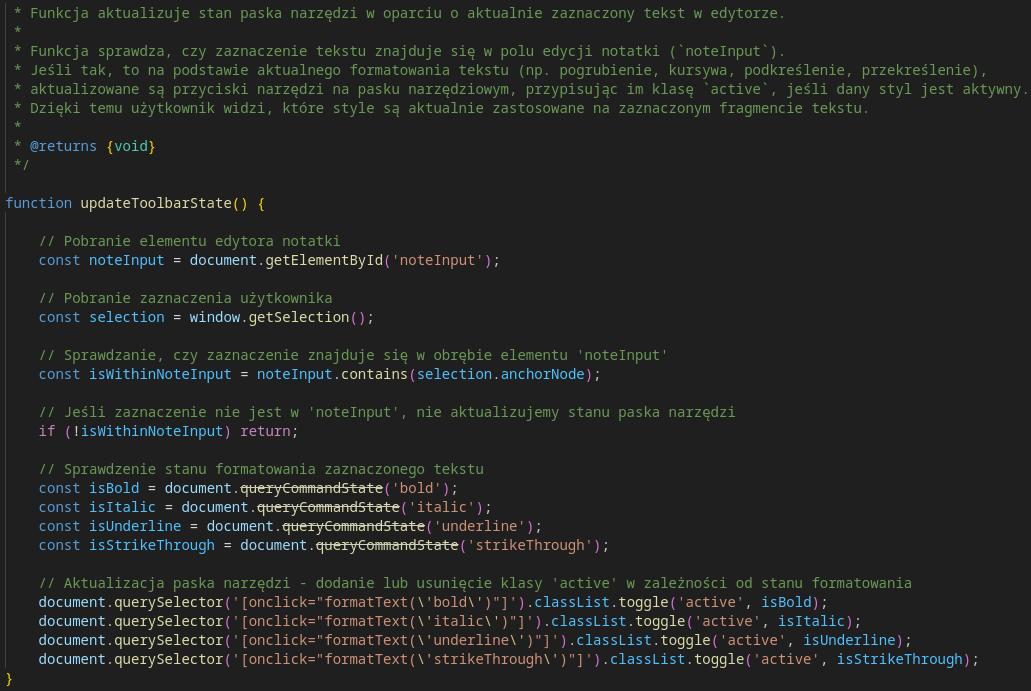
* + saveNotes() - Zapisuje notatki w lokalnej pamięci przeglądarki.



1. **Interfejs użytkownika**
   * adjustTextareaHeight(textarea) - Dostosowuje wysokość pola edycji do zawartości.



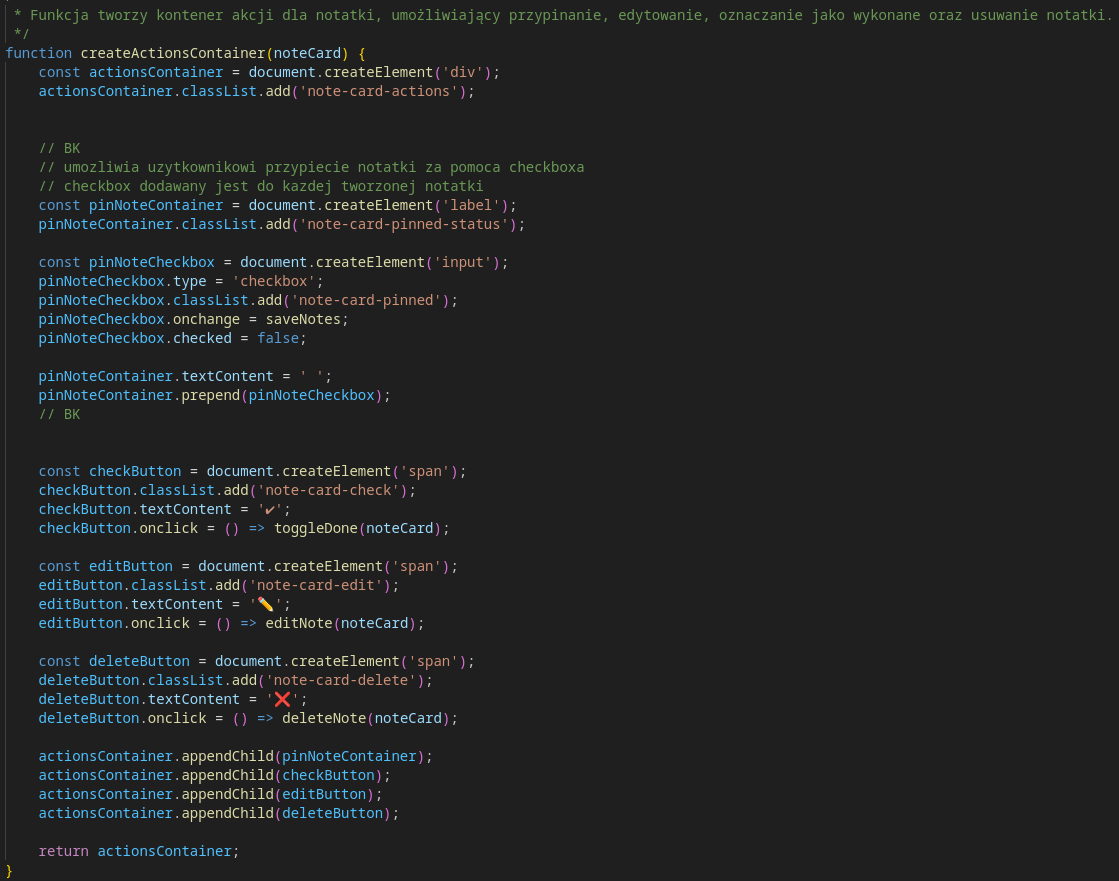
* + updateToolbarState() - Aktualizuje stan przycisków paska narzędzi w zależności od zastosowanego formatowania tekstu.



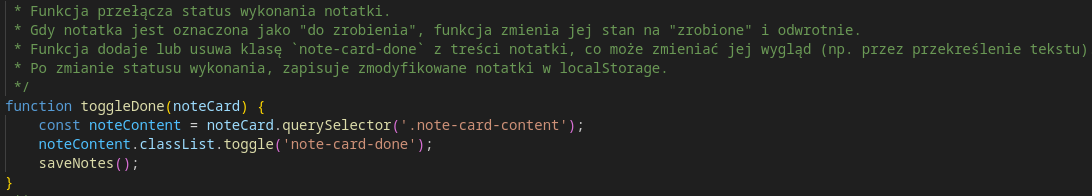
1. **Sortowanie notatek**
   * sortNotes(criteria) - Sortuje notatki według wybranego kryterium (np. tytuł, kategoria, przypięte notatki).



1. **Dodatkowe funkcje**
   * createActionsContainer(noteCard) - Tworzy kontener akcji dla notatki (przypinanie, edytowanie, usuwanie itp.).



* + toggleDone(noteCard) - Oznacza notatkę jako wykonaną.

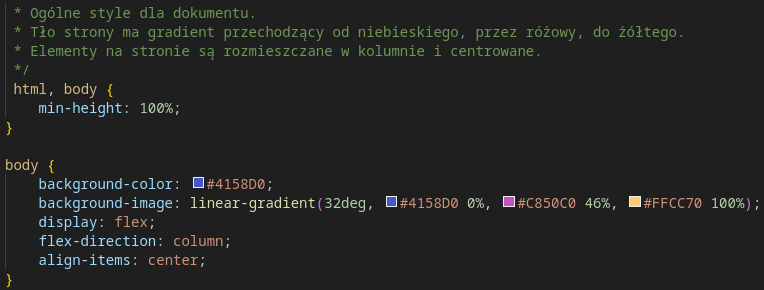


## **Struktura CSS (plik styles.css)**

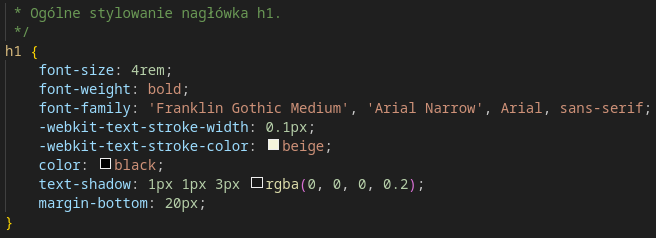
Plik styles.css odpowiada za wygląd aplikacji oraz zapewnia responsywność. Zawiera następujące sekcje:

### **1. Style ogólne**

* Ustawienia podstawowe:

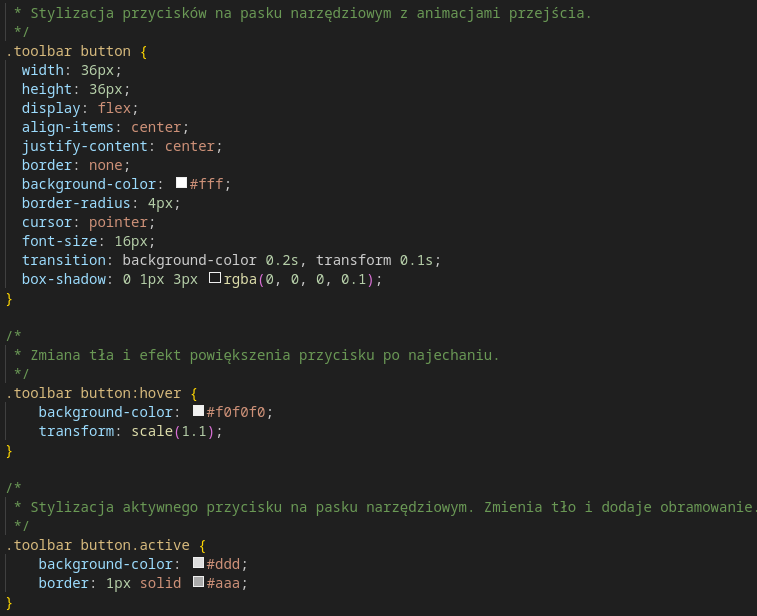


* Stylizacja nagłówka:

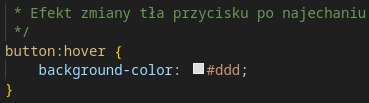


### **2. Style interaktywne**

* Przyciski paska narzędzi:

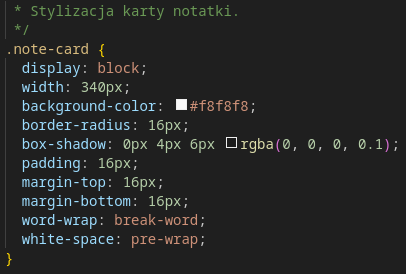


* Podświetlanie aktywnego przycisku:

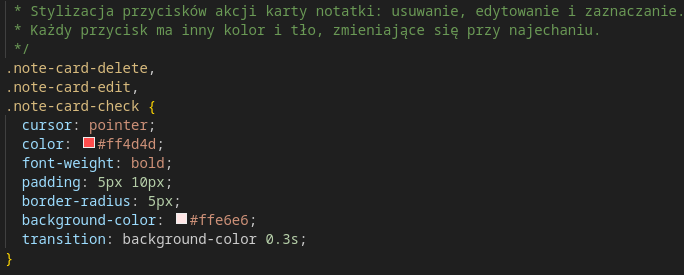


### **3. Lista notatek**

* Notatki wyświetlane w formie kart:



* Akcje notatek (przypinanie, edycja, usuwanie):



## **Instrukcja użytkowania**

1. **Tworzenie nowej notatki**
   * Wprowadź tytuł, treść i wybierz kategorię. Możesz użyć paska narzędzi do formatowania tekstu.
   * Kliknij przycisk "Dodaj", aby zapisać notatkę.
2. **Edycja istniejącej notatki**
   * Kliknij ikonę ✏️ na notatce, aby przenieść jej dane do formularza.
   * Wprowadź zmiany i kliknij "Dodaj", aby zaktualizować notatkę.
3. **Usuwanie notatek**
   * Kliknij ❌ na konkretnej notatce, aby ją usunąć.
   * Użyj przycisku "Usuń wszystko", aby usunąć wszystkie notatki.
4. **Sortowanie notatek**
   * Użyj paska sortowania, aby uporządkować notatki według wybranego kryterium.

## **Dalsze ulepszenia**

* Dodanie obsługi daty tworzenia i edycji notatek.
* Możliwość eksportu notatek do pliku (np. PDF, TXT).
* Integracja z backendem w celu synchronizacji danych między urządzeniami.

## **Podsumowanie**

Aplikacja "Notatnik" to narzędzie wspierające organizację i zarządzanie codziennymi notatkami. Dzięki prostocie obsługi oraz funkcjom formatowania tekstu stanowi praktyczne rozwiązanie dla użytkowników potrzebujących szybkiego dostępu do zapisanych informacji.