

# ***Kalkulatoreact - dokumentacja techniczna***

## **Opis Projektu**

Projekt "Kalkulatoreact" jest aplikacją kalkulatora internetowego stworzoną przy użyciu wielu technologii webowych. Aplikacja umożliwia wykonywanie działań matematycznych, dostosowywanie wyglądu wielu elementów witryny (takich jak tło witryny, rozmiar cyfr kalkulatora, obramowanie/tło/kolor cyfr przycisków), zapisywanie ich i przywracanie ustawień domyślnych. Na witrynie możliwe jest także ocenianie jej przy pomocy systemu oceny z animowanymi gwiazdkami, po oddaniu głosu użytkownicy widzą podziękowanie, a informacja na temat opinii jest wysyłana do bazy danych.

## **Technologie**

**(JavaScript, HTML5, CSS3, XML, React, jQuery, PHP, SQL, JSON, grafika komputerowa, AJAX)**

### **JavaScript**

Język programowania używany do tworzenia dynamicznych i interaktywnych elementów strony.

**Przykład:** Użycie JavaScript do odtwarzania dźwięku kliknięcia przy każdym naciśnięciu przycisku kalkulatora.

```
const clickSoundFile = './assets/click.mp3'; //  
Ścieżka do pliku dźwiękowego  
const clickSound = new Audio(clickSoundFile); //  
Utworzenie nowego obiektu Audio  
  
// Funkcja odtwarzająca dźwięk kliknięcia przy każdym  
naciśnięciu przycisku kalkulatora  
document.querySelectorAll('.calculator-grid>button').  
forEach(button => {  
    button.addEventListener('click', () => {  
        clickSound.play();  
    });  
});
```

## HTML5

Język znaczników dla struktury strony internetowej.

**Przykład:** Definicja formularza w komponencie React do zmiany ustawień kalkulatora.

```
<input type="color" id="color1Picker" value={color1}  
onChange={handleColor1Change} />
```

## CSS3

Język stylów używany do definiowania wyglądu strony.

**Przykład:** Ustawienie tła strony za pomocą gradientu w CSS.

```
body {  
    background: linear-gradient(to right, #991A1A,  
#000000);  
}
```

## XML/JSX

JSX to składnia przypominająca XML, używana w React do definiowania struktury komponentów.

**Przykład:** JSX w komponencie `Settings`, który opisuje strukturę formularza ustawień.

```
<div className="settings-container">  
    <div className="header">  
        <h2 style={{ marginTop: "0", marginBottom:  
"0" }}>Settings</h2>  
        <button className="toggle-button"  
onClick={toggleExpansion}>  
            {isExpanded ? '↑' : '↓'}  
        </button>  
    </div>  
    {isExpanded && (  
        <div className="settings-content">
```

```
        {/* Additional JSX code */}  
    </div>  
  )}  
</div>
```

## React

Biblioteka JavaScript do budowania interfejsów użytkownika, wykorzystywana do zarządzania stanem i renderowania komponentów.

**Przykład:** Komponent `Settings` zarządza stanem aplikacji i renderuje interfejs użytkownika dla ustawień kalkulatora.

```
const Settings = () => {  
  // ...stan i renderowanie komponentu  
};
```

## jQuery

Biblioteka JavaScript używana do manipulacji DOM i stylów w projekcie.

**Przykład:** Użycie jQuery do stylizacji przycisków kalkulatora na podstawie aktualnych ustawień.

```
$('.calculator-grid > button').each(function () {  
  $(this).css({  
    fontSize: fontSize,  
    borderColor: borderColor,  
    backgroundColor: backgroundColor,  
    color: numberColor  
  });  
});
```

```
});
```

## PHP

Język skryptowy używany na serwerze, zastosowany w backendzie.

**Przykład:** Skrypt `submit_rating.php` obsługuje zapis ocen do bazy danych i zwraca odpowiedź w formacie JSON.

```
$stmt = $conn->prepare("INSERT INTO ratings (rating)
VALUES (?)" );
$stmt->bind_param("i", $rating);
```

## SQL

Język zapytań do baz danych, stosowany do komunikacji z bazą danych, zastosowany w skrypcie PHP.

**Przykład:** Zapytanie SQL do wstawienia oceny do tabeli w bazie danych.

```
INSERT INTO ratings (rating) VALUES (?)
```

## JSON

Format wymiany danych, używany do przesyłania danych między klientem a serwerem.

**Przykład:** Odpowiedź w formacie JSON z serwera w skrypcie `submit_rating.php`.

```
echo json_encode(["message" => "Rating saved
successfully"]);
```

## Grafika komputerowa

GIFy i obrazki używane do poprawy estetyki i funkcjonalności strony.

## AJAX

Technika umożliwiająca asynchroniczne przesyłanie danych między klientem a serwerem.

**Przykład:** Użycie AJAX do przesyłania ocen z formularza na serwer i aktualizacji interfejsu bez przeładowania strony.

```
fetch('submit_rating.php', {  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json'  
  },  
  body: JSON.stringify({ rating: userRating })  
}).then(response => response.json())  
  .then(data => console.log(data));
```

## Skalowalność i Rozbudowa

**Projekt jest skalowalny i możliwy do rozbudowy na kilka sposobów:**

- **Dodanie nowych funkcji kalkulatora:** Można wprowadzić dodatkowe operacje matematyczne, takie jak potęgowanie czy pierwiastkowanie.

- **Rozbudowa systemu oceny:** Można dodać więcej opcji oceny, jak recenzje tekstowe, lub integrację z innymi systemami oceniania.
- **Poprawa interfejsu użytkownika:** Dalsza personalizacja i optymalizacja stylów CSS dla lepszej użyteczności i estetyki.