**Dokumentacja Techniczna Pluginu: F1 Countdown Timer**

**1. Informacje ogólne**

* **Nazwa pluginu**: F1 Countdown Timer
* **Opis**: Plugin wyświetla odliczanie do najbliższej sesji F1 lub informację o trwającej sesji. W przypadku zakończenia sezonu wyświetla odpowiedni komunikat.
* **Wersja**: 1.1
* **Autor**: Marcin
* **Ścieżka pliku głównego**: [f1-countdown.php](vscode-file://vscode-app/c:/Users/nowax/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)

**2. Struktura plików**

f1-countdown/

│

├── assets/

│   ├── script.js          // Skrypt JavaScript obsługujący odliczanie

│   └── style.css          // (opcjonalnie) Plik CSS dla stylizacji

│

├── data/

│   └── f1\_schedule.json   // Plik JSON zawierający harmonogram sesji F1

│

└── f1-countdown.php       // Główny plik pluginu

**3. Funkcjonalności**

1. **Pobieranie danych o sesjach F1**:
   * Plugin odczytuje dane z pliku data/f1\_schedule.json.
   * Na podstawie aktualnego czasu określa, czy sesja jest:
     + Trwająca (active),
     + Nadchodząca (upcoming),
     + Brak danych (no\_data),
     + Zakończenie sezonu (season\_over).
2. **Wyświetlanie odliczania**:
   * Za pomocą shortcode [f1\_countdown] plugin generuje element HTML z informacjami o sesji i odliczaniem.
3. **Stylizacja**:
   * Plugin dodaje stylizację CSS do sekcji <head> strony.
4. **Integracja z JavaScript**:
   * Dane o sesji są przekazywane z PHP do JavaScript za pomocą wp\_localize\_script.
   * Skrypt script.js obsługuje logikę odliczania i aktualizuje elementy HTML.

**4. Szczegóły implementacji**

**4.1. Funkcja f1\_get\_relevant\_session**

* **Opis**: Pobiera dane o aktualnej lub najbliższej sesji F1.
* **Wejście**: Brak.
* **Wyjście**: Tablica z informacjami o sesji:

<?php

[

    'status' => 'active' | 'upcoming' | 'no\_data' | 'season\_over',

    'gp\_name' => 'Nazwa Grand Prix',

    'session\_name' => 'Nazwa sesji',

    'session\_datetime' => 'Data i czas sesji'

]

* **Logika**:
  1. Sprawdza, czy plik f1\_schedule.json istnieje.
  2. Dekoduje dane JSON do tablicy PHP.
  3. Iteruje po sesjach, aby znaleźć:
     + Trwającą sesję (active),
     + Najbliższą przyszłą sesję (upcoming).
  4. Jeśli brak danych, zwraca no\_data.
  5. Jeśli sezon się zakończył, zwraca season\_over.

**4.2. Funkcja f1\_enqueue\_scripts**

* **Opis**: Rejestruje i ładuje skrypt JavaScript oraz przekazuje dane o sesji do frontendu.
* **Działanie**:
  1. Rejestruje skrypt assets/script.js.
  2. Pobiera dane o sesji za pomocą f1\_get\_relevant\_session.
  3. Przekazuje dane do JavaScript za pomocą wp\_localize\_script.

**4.3. Funkcja f1\_countdown\_shortcode**

* **Opis**: Generuje HTML dla shortcode [f1\_countdown].
* **Wyjście**:

<div id="f1-countdown">

    <strong><span id="session-name"></span> - <span id="gp-name"></span></strong><br>

    <span id="countdown-timer">Ładowanie...</span>

</div>

**4.4. Funkcja f1\_add\_styles**

* **Opis**: Dodaje stylizację CSS do sekcji <head> strony.
* **Stylizacja**:
  + Wyśrodkowany element z czerwonym tłem i białym tekstem.
  + Zaokrąglone rogi i responsywna szerokość.

**5. Skrypt JavaScript (script.js)**

**5.1. Funkcjonalności**

1. **Obsługa danych o sesji**:
   * Wyświetla komunikaty w zależności od statusu sesji (no\_data, season\_over, active, upcoming).
2. **Odliczanie**:
   * Oblicza różnicę czasu między aktualnym czasem a czasem rozpoczęcia sesji.
   * Aktualizuje element HTML co sekundę.

**5.2. Kluczowe fragmenty kodu**

* **Sprawdzenie statusu sesji**:

if (!window.f1\_data || f1\_data.status === 'no\_data') {

    countdownEl.textContent = "Brak danych o sesji.";

    return;

}

if (f1\_data.status === 'season\_over') {

    countdownEl.textContent = "Sezon F1 zakończony.";

    return;

}

* **Logika odliczania**:

const targetDate = new Date(f1\_data.session\_datetime);

function updateCountdown() {

    const now = new Date();

    const distance = targetDate - now;

    if (distance <= 0) {

        countdownEl.textContent = "Sesja właśnie trwa!";

        return;

    }

    const days = Math.floor(distance / (1000 \* 60 \* 60 \* 24));

    const hours = Math.floor((distance / (1000 \* 60 \* 60)) % 24);

    const minutes = Math.floor((distance / 1000 / 60) % 60);

    const seconds = Math.floor((distance / 1000) % 60);

    countdownEl.textContent = `${days}d ${hours}h ${minutes}m ${seconds}s`;

}

setInterval(updateCountdown, 1000);

**6. Przykładowy plik JSON (f1\_schedule.json)**

[

    {

        "gp\_name": "Grand Prix Australii",

        "sessions": [

            {

                "name": "Kwalifikacje",

                "datetime": "2025-07-01 14:00:00",

                "duration\_minutes": 60

            },

            {

                "name": "Wyścig",

                "datetime": "2025-07-02 16:00:00",

                "duration\_minutes": 120

            }

        ]

    },

    {

        "gp\_name": "Grand Prix Monako",

        "sessions": [

            {

                "name": "Kwalifikacje",

                "datetime": "2025-07-15 15:00:00",

                "duration\_minutes": 60

            },

            {

                "name": "Wyścig",

                "datetime": "2025-07-16 17:00:00",

                "duration\_minutes": 120

            }

        ]

    }

]

**7. Instalacja i użycie**

1. **Instalacja**:
   * Skopiuj folder f1-countdown do katalogu wp-content/plugins/.
   * Aktywuj plugin w panelu WordPress.
2. **Dodanie shortcode**:
   * Wstaw [f1\_countdown] w dowolnym miejscu na stronie lub w poście.

**8. Testowanie**

* **Scenariusze testowe**:
  1. Brak pliku f1\_schedule.json.
  2. Nieprawidłowy format JSON.
  3. Brak sesji w harmonogramie.
  4. Trwająca sesja.
  5. Nadchodząca sesja.
  6. Zakończenie sezonu.

**9. Uwagi dotyczące bezpieczeństwa**

* **Walidacja danych JSON**:
  + Sprawdź poprawność danych po dekodowaniu JSON.
* **Sanityzacja danych**:
  + Użyj funkcji esc\_html dla danych wyświetlanych w HTML.
* **Obsługa błędów**:
  + Dodaj logowanie błędów w przypadku problemów z plikiem JSON.

**10. Możliwe ulepszenia**

* Przeniesienie stylów do osobnego pliku CSS.
* Dodanie opcji konfiguracji w panelu WordPress.
* Obsługa wielojęzyczności za pomocą funkcji \_\_() i \_e().