

# GearUpScheduler

APLIKACJA DO REJESTROWANIA WIZYT U MECHANIKA

PROJEKT SYSTEMU - APLIKACJE MULTIMEDIALNE

OSOBY ODPOWIADAJĄCE ZA APLIKACJE:

- MATEUSZ ZIENTEK
- DOMINIK BOGUSZ
- ERYK HOLIKA
- SZYMON WARSZAWSKI
- SEBASTIAN JASIŃSKI

# Spis treści

Wstęp.....	2
Temat .....	2
Cel .....	2
Struktura projektu .....	2
Funkcje .....	2
Wymagania .....	2
Podział prac.....	3
Opis wykonanych prac: .....	3
Instalacja i konfiguracja .....	4
Instalacja .....	4
Konfiguracja .....	4
Uruchomienie projektu .....	5
Logowanie.....	5
Specyfikacja zewnętrzna .....	6
Wymagania niefunkcjonalne .....	11
Opis technologii użytych w projekcie .....	12
React .....	12
Node.js .....	12
MongoDB .....	12
Spis ilustracji.....	13

# Wstęp

## Temat

Głównym założeniem zadania było zaprojektowanie oraz napisanie dowolnej aplikacji mobilnej w dowolnym języku programowania.

## Cel

Głównym założeniem projektu było napisanie aplikacji internetowej „GearUpScheduler” z wykorzystaniem freme worka React do zbudowania frontendu, Node.js do zbudowania backendu oraz Mongoddb db aby utworzyć bazę danych. Aplikacja ma na celu umożliwienie łatwiejszego zarezerwowania wizyty u mechanika dla klienta oraz łatwiejsze zorganizowanie wizyt i klientów dla właściciela warsztatu.

## Struktura projektu

Projekt został podzielony na dwie główne części:

- frontend - interfejs użytkownika stworzony w React
- backend - serwer stworzony w Node.js z wykorzystaniem Express oraz baza danych MongoDB

## Funkcje

Funkcje aplikacji:

- Rezerwowanie wizyt u mechanika
- Przeglądanie dostępności terminów
- Edycja terminów i ich anulowanie
- Powiadomienia o zbliżających się wizytach
- Historia wszystkich wizyt

## Wymagania

Wymagane oprogramowanie, bazy i środowiska do uruchomienia aplikacji:

- Node.js (w wersji x.x.x)
- MongoDB (w wersji x.x.x)
- Przeglądarka internetowa z obsługą JavaScript

## Podział prac

### Backend oraz baza danych:

- MATEUSZ ZIENTEK
- SZYMON WARSZAWSKI

### Frontend

- DOMINIK BOGUSZ
- ERYK HOLIKA

### Dokumentacja projektu

- SEBASTIAN JASIŃSKI

## Opis wykonanych prac:

- Utworzenie struktury strony
- Utworzenie serwera, baza danych, konfiguracja
- Dodanie schematu wizyt
- Dodanie endpointów logowania, rejestracji, wylogowania oraz użytkowników
- Utworzenie głównej strony z wyborem serwisu
- Dodanie logowania
- Utworzenie wstępnej dokumentacji
- Zaimplementowanie widoczności opartej na adresach URL dla modułów uwierzytelniania
- Dodanie endpointów do wizyt (zapisywanie, usuwanie wizyty)
- Dodanie wybierania czasu wizyty do formularza
- Dodanie widoku administratora oraz poprawa wyglądu głównej strony
- Dodanie endpointów do sprawdzania dostępnych godzin
- Dodanie funkcjonalności zapisywania wizyt
- Dodanie sortowania wizyt po dacie
- Dodanie widoku ilości dni do wizyty
- Naprawa bugów
- Utworzenie finałowej dokumentacji technicznej z instrukcją umożliwiającą uruchomienie projektu

# Instalacja i konfiguracja

## Instalacja

Link do repozytorium projektu na GitHub -> <https://github.com/wasu-code/GearUpScheduler/>

Instalacja aplikacji na komputerze

1. Sklonuj repozytorium na swój lokalny komputer:

```
git clone https://github.com/wasu-code/GearUpScheduler
```

2. Wejdź do katalogu projektu:

```
cd GearUpScheduler
```

3. Zainstaluj zależności Frontend

- a. Wejdź do katalogu frontend:

```
cd frontend
```

- b. Zainstaluj zależności:

```
npm install
```

4. Zainstaluj zależności Backend

- a. Wejdź do katalogu backend:

```
cd backend
```

- b. Zainstaluj zależności:

```
npm install
```

## Konfiguracja

Aby skonfigurować aplikację, wejdź do plik backend/.env i uzupełnij odpowiednie zmienne środowiskowe.

PORT=3000

MONGO\_URI=mongodb://localhost:27017/gearup

# Uruchomienie projektu

## Uruchomienie aplikacji:

1. Uruchomienie bazy danych
2. Uruchomienie aplikacji Frontend
  - a. Wejdź do katalogu frontend:  
`cd frontend`
  - b. Uruchom aplikację:  
`npm run dev`
3. Uruchomienie aplikacji Backend
  - a. Wejdź do katalogu backend:  
`cd backend`
  - b. Uruchom aplikację  
`npm run dev`

4. Otwórz przeglądarkę i przejdź do:

`http://localhost:3000`

## Logowanie

Zalogowanie się do użytkowników w aplikacji

Aplikacja posiada 2 typy użytkowników

- Mechanik – ma dostęp do panelu w którym widać wszystkie zarejestrowane wizyty od wszystkich klientów

Dane logowania do użytkownika Mechanik

Login: [admin@example.com](mailto:admin@example.com)

Hasło: ABCabc123!@#

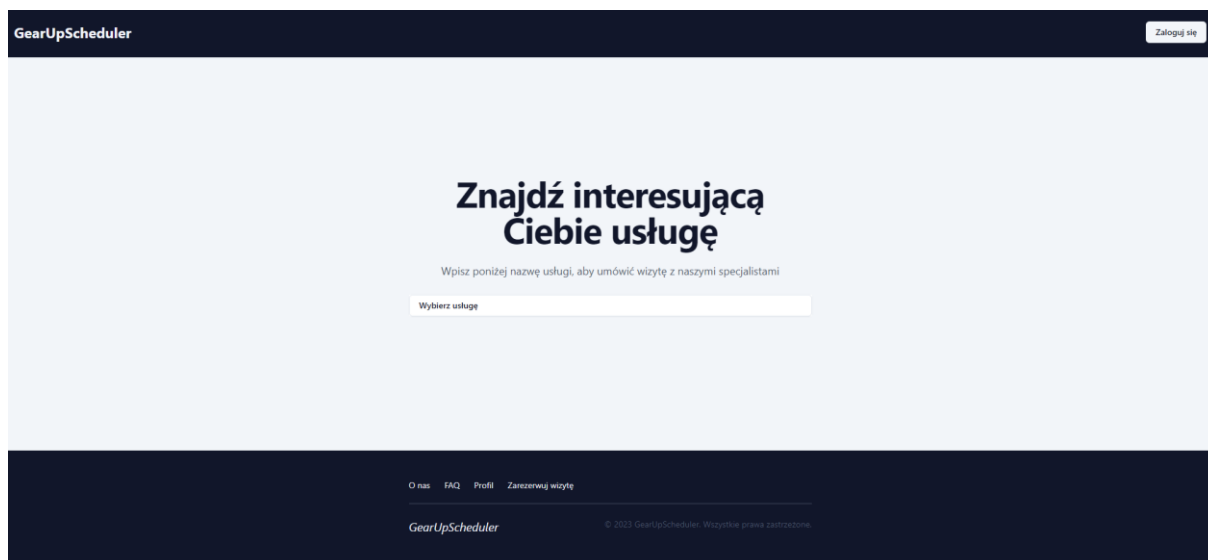
- Klient – po zarejestrowaniu lub zalogowaniu ma możliwość wybrania interesującej go usługi, wybrania daty i godziny wykonania wizyty oraz podejrzenia w swoim panelu użytkownika daty, godziny i nazwy usługi. Ma on również możliwość odwołania wizyty

Dane logowania do użytkownika Klient:

Każdy klient ma swój unikatowy login i hasło podawane przy rejestracji

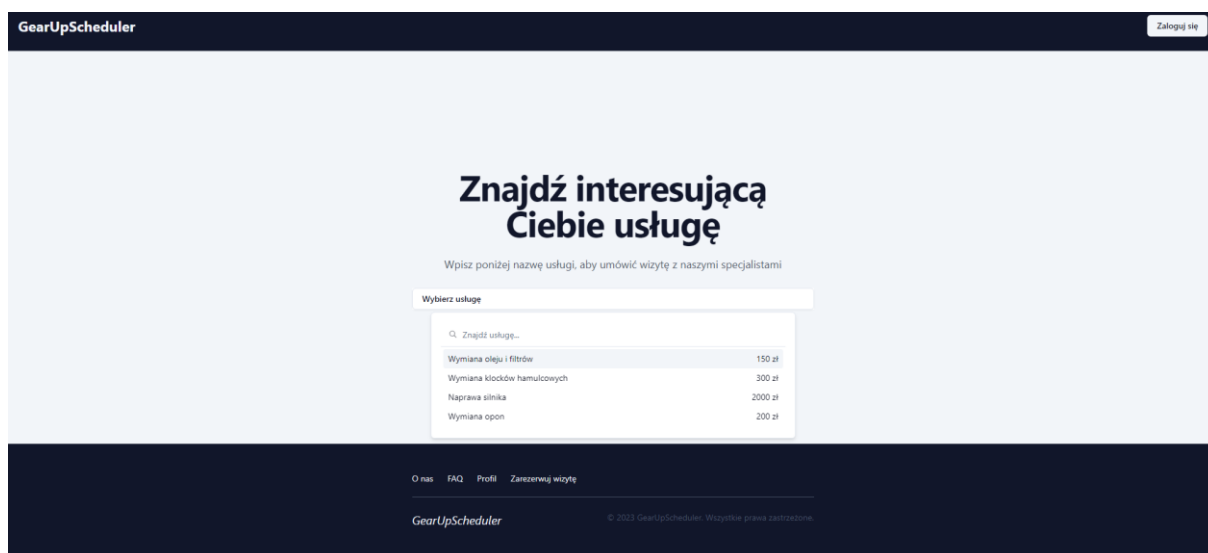
## Specyfikacja zewnętrzna

Po uruchomieniu strony internetowej (://localhost:3000) użytkownik powinien zobaczyć stronę główną aplikacji GearUpScheduler która pozwala na wybór usługi w warsztacie samochodowym.



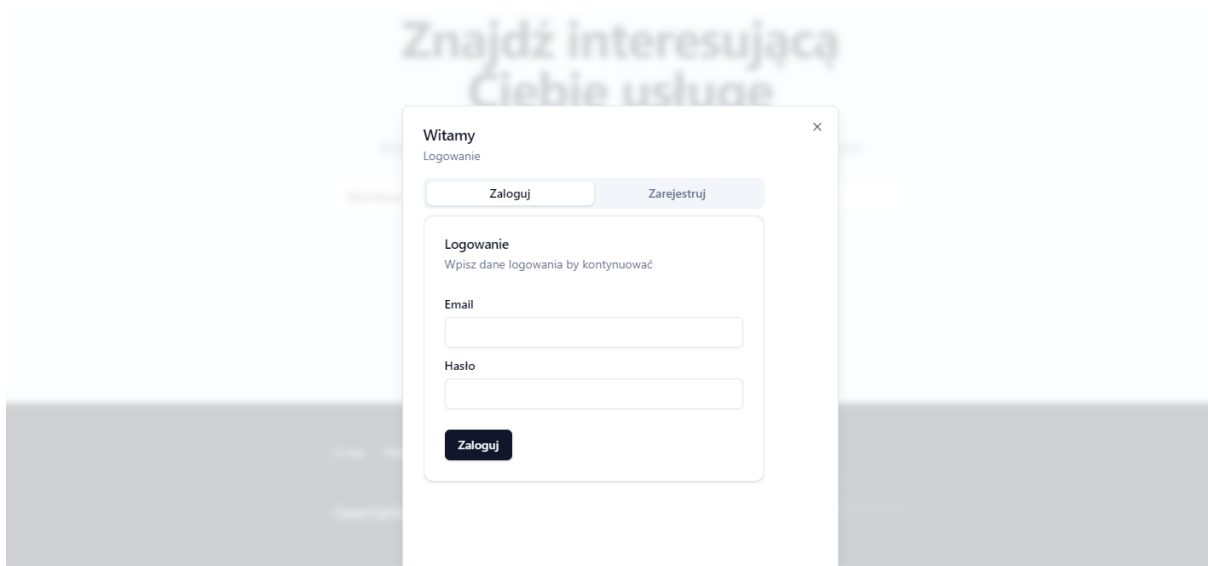
Rysunek 1- strona tytułowa

Po kliknięciu w przycisk „Wybierz usługę” pojawia się rozwijające menu z wyszukiwarką. Znajdują się w nim usługi oraz ich ceny.



Rysunek 2 - wybór usługi

Aby zarezerwować usługę należy być zalogowanym więc jeśli tego nie zrobił użytkownik wcześniej zostanie przekierowany na ekran logowania lub rejestracji



The image shows a modal window titled "Witamy" (Welcome) with the subtitle "Logowanie" (Login). At the top, there are two buttons: "Zaloguj" (Login) and "Zarejestruj" (Register). The "Zaloguj" button is active. Below the buttons, the section is titled "Logowanie" with the instruction "Wpisz dane logowania by kontynuować" (Enter login details to continue). There are two input fields: "Email" and "Hasło" (Password). Below the "Hasło" field is a "Zaloguj" button.

Jeśli użytkownik posiada konto wpisuje swój adres email oraz hasło a jeśli nie posiada przechodzi do zakładki „Zarejestruj” która pozwala na rejestrację nowych użytkowników. Użytkownik musi wprowadzić Imię, Nazwisko, Hasło (musi zawierać co najmniej 8 liter długości, jedną dużą literę oraz jedną liczbę), powtórzyć hasło oraz adres email

Rysunek 3 - ekran logowania



The image shows a modal window titled "Witamy" (Welcome) with the subtitle "Logowanie" (Login). At the top, there are two buttons: "Zaloguj" (Login) and "Zarejestruj" (Register). The "Zarejestruj" button is active. Below the buttons, the section is titled "Rejestracja" (Registration) with the instruction "Wybierz dane jakich będziesz używał w serwisie. Pamiętaj o podaniu silnego hasła." (Choose the data you will use in the service. Remember to provide a strong password.). There are four input fields: "Imię" (First Name), "Nazwisko" (Last Name), "Hasło" (Password), and "Powtórz Hasło" (Repeat Password).

Rysunek 4 - ekran rejestracji



Po zalogowaniu się użytkownik powinien zobaczyć kalendarz z możliwością wyboru daty oraz godziny wizyty

**GearUpScheduler** [Sprawdź wizytę](#) [Wyloguj](#)

## Znajdź interesującą Ciebie usługę

Wpisz poniżej nazwę usługi, aby umówić wizytę z naszymi specjalistami

Wymiana oleju i filtrów

**Wybierz datę**

**Wybierz godzinę**

Wizyty są dostępne w godzinach: 10:00-18:00

13:00

Koszt wizyty: 150zł  
Szacunkowy czas trwania: 1h

[Zapisz wizytę](#)

Rysunek 5 - wybór daty i godziny usługi

Po wyborze daty i godziny wizyty w warsztacie użytkownikowi powinno pokazać się podsumowanie z wszystkimi informacjami dotyczącymi usługi

## Znajdź interesującą Ciebie usługę

Wpisz poniżej nazwę usługi, aby umówić wizytę z naszymi specjalistami

Wymiana oleju i filtrów

**Pomyślnie zaplanowano wizytę!**

**Wybrana usługa:**  
Wymiana oleju i filtrów

**Data wizyty:**  
30.11.2023

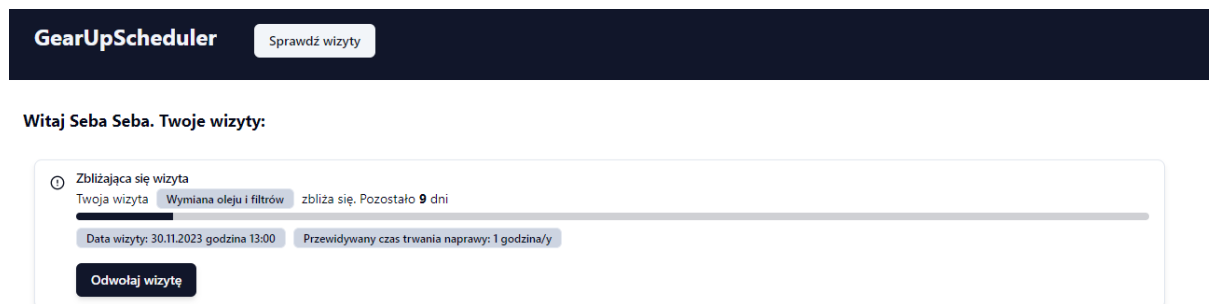
**Koszt wizyty:**  
150zł

**Wizyta w godzinach:**  
13:00 - 14:00

[Zobacz szczegóły](#)

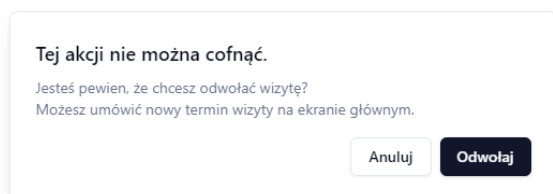
Rysunek 6 - podsumowanie usługi

Po naciśnięciu na przycisk „Sprawdź wizyty” w lewym górnym rogu użytkownik powinien zobaczyć wszystkie swoje wizyty oraz ile czasu do nich pozostało



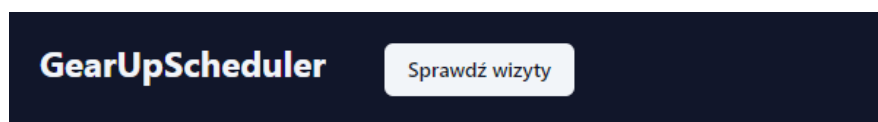
Rysunek 7 - panel użytkownika do monitorowania wizyt

Użytkownik ma również możliwość odwołania wizyty



Rysunek 8 - odwołanie wizyty

Po tej akcji z panelu użytkownika powinna zniknąć wizyta



**Witaj Seba Seba. Twoje wizyty:**

Brak zaplanowanych wizyt

Rysunek 9 - pusty panel użytkownika

Po zalogowaniu się na konto mechanika mamy dostęp do panelu na którym widać wszystkie zaplanowane wizyty od różnych użytkowników

## Sprawdź wizyty

Imię	Nazwisko	Usługa	Data wizyty	Godzina wizyty	Koszt	
mateusz	mateusz	Naprawa silnika	2023-12-28	12:00 - 17:00	2000zł	🗑️
Test	Test	Wymiana klocków hamulcowych	2023-11-23	13:00 - 15:00	300zł	🗑️
mateusz	mateusz	Naprawa silnika	2024-01-17	11:00 - 16:00	2000zł	🗑️
Test	Test	Wymiana opon	2023-11-23	12:00 - 13:00	200zł	🗑️
Test	Test	Wymiana opon	2023-11-23	17:00 - 18:00	200zł	🗑️
Test	Test	Wymiana klocków hamulcowych	2023-11-22	16:00 - 18:00	300zł	🗑️
Test	Test	Wymiana klocków hamulcowych	2023-11-23	15:00 - 17:00	300zł	🗑️
Test	Test	Wymiana klocków hamulcowych	2023-11-24	14:00 - 16:00	300zł	🗑️
mateusz	mateusz	Wymiana klocków hamulcowych	2023-12-30	15:00 - 17:00	300zł	🗑️
mateusz	mateusz	Wymiana klocków hamulcowych	2023-11-21	15:00 - 17:00	300zł	🗑️
Liczba wizyt: 10						
Lista umówionych wizyt						

Rysunek 10 - panel mechanika

# Wymagania niefunkcjonalne

## Wydajność:

- Aplikacja powinna działać płynnie i responsywnie, zapewniając płynne animacje i reakcje na interakcje użytkownika.
- Minimalne opóźnienia w odpowiedzi na interakcje użytkownika.

## Kompatybilność:

- Aplikacja powinna być kompatybilna z różnymi przeglądarkami internetowymi takimi jak: Safari, Chrome, Opera, Edge

## Responsywność:

- Aplikacja powinna dostosowywać się do rozdzielczości ekranu i zachowywać odpowiednie proporcje elementów graficznych.
- Intuicyjne sterowanie, które zapewnia dobrą responsywność i precyzję.

## Estetyka:

- Przyjemna dla oka i atrakcyjna wizualnie oprawa graficzna, uwzględniająca odpowiednie kolory, animacje i efekty wizualne.
- Czytelny interfejs użytkownika, który zapewnia wygodną nawigację i odczytywalność informacji.

## Stabilność:

- Aplikacja powinna działać bez awarii i błędów, zapewniając stabilne działanie nawet przy dłuższym uruchomieniu.

## Rozszerzalność:

- Możliwość rozbudowy aplikacji poprzez dodawanie nowych elementów, funkcjonalności.

## Dokumentacja:

- Pełna i czytelna dokumentacja techniczna, opisująca strukturę aplikacji,

## Wymagania dotyczące wydania:

- Gotowy do publikacji build aplikacji, który może być łatwo wdrożony na każdy serwer

# Opis technologii użytych w projekcie

## React

React to popularna biblioteka JavaScript do tworzenia interfejsów użytkownika (UI) i jednostronicowych aplikacji internetowych. Została stworzona przez Facebooka i jest powszechnie używana do tworzenia aplikacji internetowych o dynamicznych, interaktywnych i responsywnych interfejsach użytkownika. React jest znany ze swojej architektury opartej na komponentach, która pozwala programistom tworzyć UI poprzez komponowanie wielokrotnego użytku i niezależnych komponentów.

React posiada dużą i aktywną społeczność, która oferuje wiele zasobów, bibliotek i narzędzi dla programistów. Często jest używany w połączeniu z innymi technologiami, takimi jak Redux do zarządzania stanem aplikacji czy Axios do wykonywania zapytań API. React Native, framework zbudowany na bazie React, jest również używany do tworzenia aplikacji mobilnych na platformy iOS i Android.

## Node.js

Node.js to środowisko uruchomieniowe oparte na języku JavaScript, które umożliwia wykonywanie kodu JavaScript po stronie serwera. Zostało stworzone przez Ryana Dahla i jest rozwijane jako projekt open source. Node.js jest zaprojektowane w sposób asynchroniczny i oparte na zdarzeniach, co sprawia, że jest wyjątkowo efektywne w obsłudze wielu równoległych połączeń i operacji wejścia/wyjścia (I/O).

Node.js Express (często nazywane po prostu Express) to popularny framework aplikacji internetowych oparty na Node.js. Express dostarcza zestaw narzędzi i funkcji, które ułatwiają tworzenie aplikacji internetowych i API. Jest to jedno z najpopularniejszych narzędzi do tworzenia serwerów HTTP w ekosystemie Node.js.

Node.js Express jest szczególnie przydatne do tworzenia serwerów API, aplikacji internetowych i mikrousług. Dzięki prostocie i elastyczności jest popularnym wyborem dla wielu programistów JavaScript działających w środowisku Node.js.

## MongoDB

MongoDB to popularna, dokumentowa baza danych typu NoSQL, która jest wykorzystywana w aplikacjach internetowych, mobilnych i innych systemach. MongoDB jest jednym z przedstawicieli bazy danych typu dokumentowego, co oznacza, że dane są przechowywane w postaci dokumentów w formacie BSON (Binary JSON).

MongoDB jest często wykorzystywane w aplikacjach, które wymagają dużej elastyczności w strukturze danych i potrzebują skalowalności. Jest popularne wśród firm technologicznych, które korzystają z rozproszonych systemów i przetwarzania dużych ilości danych. Aby zacząć pracę z MongoDB, musisz zainstalować bazę danych na swoim serwerze, a następnie korzystać z odpowiednich bibliotek i narzędzi dostępnych dla różnych języków programowania, takich jak Node.js, Python, Java i inne.

## Spis ilustracji

Rysunek 1- strona tytułowa .....	6
Rysunek 2 - wybór usługi.....	6
Rysunek 3 - ekran logowania .....	7
Rysunek 4 - ekran rejestracji.....	7
Rysunek 5 - wybór daty i godziny usługi.....	8
Rysunek 6 - podsumowanie usługi .....	8
Rysunek 7 - panel użytkownika do monitorowania wizyt .....	9
Rysunek 8 - odwołanie wizyty .....	9
Rysunek 9 - pusty panel użytkownika.....	9
Rysunek 10 - panel mechanika .....	10