

Projekt Aplikacji Do wypożyczania sprzętu rolniczego Jan Ziolo

[Projekt na Githubie](#)

1. Spis Treści

1. Spis Treści.....	2
2. Opis zagadnienia biznesowego	4
3. Opis funkcjonalności	4
3.1. Funkcje użytkownika	4
3.2. Funkcje administratora	4
3.3. Autoryzacja	4
4. Model bazy danych	5
4.1. Tabela users.....	5
4.2. Tabela machines	5
4.3. Tabela reservations.....	6
5. Technologia i narzędzia	6
6. Widoki aplikacji	6
7. Opis Tras (routes)	7
7.1. Trasy autoryzacyjne (/)	7
7.2. Trasy administratora (/admin).....	7
7.3. Trasy użytkownika (/user)	8
8. Ograniczenia systemowe	9
9. Aktorzy (Użytkownicy)	9
10. Przypadki Użycia.....	9
10.1. Dodanie użytkownika (Administrator)	9
10.2. Edycja danych użytkownika (Administrator)	10
Scenariusz główny:	10
Scenariusz alternatywny:.....	10
10.3. Usunięcie użytkownika (Administrator).....	10
Scenariusz główny:	10
10.4. Dodanie nowej maszyny (Administrator)	10

Scenariusz główny:	10
Scenariusz alternatywny:.....	11
10.5. Edycja maszyny (Administrator)	11
Scenariusz główny:	11
10.6. Usunięcie maszyny (Administrator)	11
Scenariusz główny:	11
10.7. Przeglądanie listy maszyn (Użytkownik)	11
Scenariusz główny:	11
10.8. Rezerwacja maszyny (Użytkownik)	12
Scenariusz główny:	12
Scenariusz alternatywny:.....	12
10.9. Przeglądanie swoich rezerwacji (Użytkownik)	12
Scenariusz główny:	12
10.10. Edycja własnej rezerwacji (Użytkownik).....	12
Scenariusz główny:	13
Scenariusz alternatywny:.....	13
10.11. Anulowanie rezerwacji (Użytkownik)	13
Scenariusz główny:	13
10.12. Usunięcie rezerwacji (Administrator)	13
Scenariusz główny:	13
10.13. Logowanie użytkownika	13
Scenariusz główny:	14
Scenariusz alternatywny:.....	14
10.14. Wylogowanie użytkownika	14
Scenariusz główny:	14
11. Struktura projektu	14

2. Opis zagadnienia biznesowego

Grupa rolników utworzyła spółdzielnię i wspólnie zakupiła maszyny rolnicze, takie jak kombajn, traktory i bronę. Członkowie spółdzielni mogą korzystać z tych maszyn według harmonogramu rezerwacji opartego na zasadzie "kto pierwszy, ten lepszy".

Rezerwacje dotyczą całych dni i nie można ich dokonywać z większym niż trzymiesięcznym wyprzedzeniem. Celem aplikacji jest umożliwienie zarządzania rezerwacjami oraz administrowania maszynami i użytkownikami.

3. Opis funkcjonalności

3.1. Funkcje użytkownika

Przeglądanie listy dostępnych maszyn.

Sprawdzanie harmonogramu rezerwacji danej maszyny.

Tworzenie nowych rezerwacji.

Anulowanie własnych rezerwacji.

Edycja własnych rezerwacji.

3.2. Funkcje administratora

Dodawanie, edytowanie i usuwanie maszyn.

Przeglądanie i usuwanie rezerwacji.

Zarządzanie kontami użytkowników (tworzenie, edycja, usuwanie).

3.3. Autoryzacja

Logowanie i wylogowywanie.

Role: użytkownik (może rezerwować maszyny) i administrator (zarządza aplikacją).

4. Model bazy danych

4.1. Tabela users

Kolumna	Typ	Opis
ID	INTEGER (PK, AutoIncrement)	Unikalny identyfikator użytkownika.
firstName	STRING	Imię użytkownika.
lastName	STRING	Nazwisko użytkownika.
username	STRING (unique)	Login użytkownika, wykorzystywany przy logowaniu.
password	STRING	Haszowane hasło użytkownika (bcrypt, 10 salt rounds).
role	ENUM ('user', 'admin')	Rola użytkownika: user lub admin

Uwagi:

- Hasła są **haszowane** przy tworzeniu nowego użytkownika.

4.2. Tabela machines

Kolumna	Typ	Opis
ID	INTEGER (PK, AutoIncrement)	Unikalny identyfikator maszyny.
name	STRING	UNIQUE, NOT NULL
description	TEXT	Opis techniczny maszyny.
image_url	STRING	Ścieżka do zdjęcia maszyny w folderze /public/images/machines. Akceptowane formaty plików: .jpg, .jpeg, .png, .gif.

Uwagi:

- Podczas dodawania maszyny plik graficzny jest wgrywany przy pomocy Multera.
- Maksymalny rozmiar pliku to 5 MB.

4.3. Tabela reservations

Kolumna	Typ	Opis
ID	INTEGER (PK, AutoIncrement)	AUTO_INCREMENT
user_id	INTEGER (FK)	Identyfikator użytkownika rezerwującego.
machine_id	INTEGER (FK)	Identyfikator maszyny zarezerwowanej.
date	DATE	Data rezerwacji (w formacie YYYY-MM-DD).

Uwagi:

- Każda rezerwacja dotyczy konkretnego dnia (rezerwacje są codziennymi wpisami).
- Walidacja dat:
 - Data rezerwacji musi być co najmniej **jutrzejsza**.
 - Maksymalny termin rezerwacji to **3 miesiące** do przodu od dnia rezerwacji.
- Brak automatycznego usuwania rezerwacji

5. Technologia i narzędzia

- **Node.js + Express.js** – serwer backendowy
- **Sequelize** – ORM dla bazy danych SQLite
- **SQLite** – baza danych
- **EJS** – silnik szablonów HTML
- **Passport.js** – obsługa logowania i sesji
- **Multer** – obsługa uploadu zdjęć
- **connect-flash** – komunikaty błędów i potwierdzeń
- **CSS** – stylowanie aplikacji
- **Flatpickr** – wybór daty w kalendarzu
- **bcrypt.js** – haszowanie haseł użytkowników
- **dotenv** – obsługa zmiennych środowiskowych

6. Widoki aplikacji

System korzysta z szablonów EJS podzielonych na:

- **Widoki logowania:**
 - login.ejs – Formularz logowania.
- **Widoki dla administratora:**
 - admin/index.ejs – Panel główny admina (lista rezerwacji).
 - admin/machines.ejs – Zarządzanie maszynami.
 - admin/users.ejs – Zarządzanie użytkownikami.
- **Widoki dla użytkownika:**
 - user/index.ejs – Lista dostępnych maszyn i możliwość rezerwacji.
 - user/reservations.ejs – Lista własnych rezerwacji, możliwość edycji lub usunięcia.

Stylizacja:

- Plik style.css odpowiada za główny wygląd strony.
- Plik login.css odpowiada za wygląd strony logowania.

7. Opis Tras (routes)

W aplikacji występują trzy główne grupy tras:

- **Autoryzacyjne (/)**
- **Panel admina (/admin)**
- **Panel użytkownika (/user)**

7.1. Trasy autoryzacyjne (/)

Metoda	Adres	Opis
GET	/login	Wyświetla formularz logowania.
POST	/login	Obsługuje logowanie użytkownika.
GET	/logout	Wylogowuje użytkownika i przekierowuje na stronę główną.

7.2. Trasy administratora (/admin)

Metoda	Adres	Opis
GET	/admin	Strona główna panelu admina (przegląd rezerwacji).
POST	/admin/reservations/delete	Usuwa rezerwację wybranego użytkownika w wybranym zakresie dat.
GET	/admin/machines	Wyświetla wszystkie maszyny.
POST	/admin/machines/add	Dodaje nową maszynę (wraz ze zdjęciem).
GET	/admin/machines/:id	Pobiera szczegóły konkretnej maszyny (AJAX).
POST	/admin/machines/:id/edit	Aktualizuje dane maszyny (nazwa, opis, zdjęcie).
POST	/admin/machines/:id/delete	Usuwa maszynę.
GET	/admin/users	Wyświetla listę użytkowników.
POST	/admin/users	Dodaje nowego użytkownika.
POST	/admin/users/edit/:id	Edytuje dane istniejącego użytkownika.
POST	/admin/users/delete/:id	Usuwa użytkownika.

Dostęp: tylko użytkownicy o roli admin.

7.3. Trasy użytkownika (/user)

Metoda	Adres	Opis
GET	/user	Strona główna użytkownika (lista maszyn).
POST	/user/reserve	Dodanie nowej rezerwacji maszyny.
GET	/user/reservations	Lista rezerwacji użytkownika.
POST	/user/reservations/edit	Edycja wybranej rezerwacji (zakres dat).
GET	/user/reservations/delete	Usunięcie wybranej rezerwacji (zakres dat).
GET	/user/reservations/:machineId	Pobiera zajęte daty dla maszyny (AJAX, pod edycję rezerwacji).

Dostęp: tylko użytkownicy o roli user.

8. Ograniczenia systemowe

Użytkownik może zarezerwować maszynę maksymalnie na 3 miesiące do przodu.

Nie można rezerwować maszyny na godziny, tylko na całe dni.

Jedna maszyna może być używana tylko przez jedną osobę danego dnia.

9. Aktorzy (Użytkownicy)

1. **Zwykły użytkownik (rolnik)** – Może przeglądać listę maszyn, rezerwować maszyny i anulować oraz edytować swoje rezerwacje.
2. **Administrator** – Może dodatkowo zarządzać użytkownikami i maszynami (dodawać, edytować, usuwać).
3. **System** – Aplikacja webowa, która przetwarza żądania użytkowników i komunikuje się z bazą danych.

10. Przypadki Użycia

10.1. Dodanie użytkownika (Administrator)

Aktor: Administrator

Opis: Administrator chce dodać nowego użytkownika do systemu.

Scenariusz główny:

1. Administrator loguje się do systemu.
2. Przechodzi na stronę /admin/users.
3. Klika przycisk „Dodaj użytkownika”.
4. Wypełnia formularz (imię, nazwisko, nazwa użytkownika, hasło, rolę).
5. Klikając „Zapisz”, przesyła dane.
6. System tworzy nowego użytkownika i zapisuje w bazie danych.

Scenariusz alternatywny:

(4a) Jeśli nazwa użytkownika już istnieje – system wyświetla komunikat o błędzie.

10.2. Edycja danych użytkownika (Administrator)

Aktor: Administrator

Opis: Administrator chce zmienić dane istniejącego użytkownika.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin/users.
2. Klikając na użytkownika, otwiera formularz edycji.
3. Edytuje dane (imię, nazwisko, nazwa użytkownika, hasło, rola).
4. Klikając „Zapisz zmiany”, przesyła formularz.
5. System aktualizuje dane użytkownika.

Scenariusz alternatywny:

- (3a) Jeśli administrator nie poda wszystkich wymaganych danych – system wyświetla komunikat o błędzie.

10.3. Usunięcie użytkownika (Administrator)

Aktor: Administrator

Opis: Administrator chce usunąć konto użytkownika.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin/users.
2. Klikając na użytkownika, otwiera formularz edycji.
3. Klika „Usuń”.
4. System usuwa użytkownika i powiązane z nim rezerwacje.

10.4. Dodanie nowej maszyny (Administrator)

Aktor: Administrator

Opis: Administrator chce dodać nową maszynę do systemu.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin/machines.
2. Klika przycisk „Dodaj maszynę”.
3. Wypełnia formularz (nazwa, opis, zdjęcie).

4. Klikając „Zapisz”, przesyła dane.
5. System zapisuje nową maszynę w bazie danych.

Scenariusz alternatywny:

- (3a) Jeśli nazwa maszyny już istnieje – system wyświetla komunikat o błędzie.

10.5. Edycja maszyny (Administrator)

Aktor: Administrator

Opis: Administrator chce zmienić dane istniejącej maszyny.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin/machines.
2. Klikając na maszynę, otwiera formularz edycji.
3. Edytuje nazwę, opis lub zmienia zdjęcie.
4. Klikając „Zapisz”, aktualizuje dane maszyny.

10.6. Usunięcie maszyny (Administrator)

Aktor: Administrator

Opis: Administrator chce usunąć maszynę.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin/machines.
2. Klikając na maszynę, otwiera formularz edycji.
3. Klika „Usuń”.
4. System usuwa maszynę i wszystkie związane z nią rezerwacje.

10.7. Przeglądanie listy maszyn (Użytkownik)

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce przeglądać dostępne maszyny.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /user.
2. System wyświetla listę wszystkich maszyn dostępnych w systemie.

10.8. Rezerwacja maszyny (Użytkownik)

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce zarezerwować maszynę na wybrany przedział czasu.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /user.
2. Klikając na maszynę, otwiera formularz rezerwacji.
3. Wybiera datę początkową i końcową.
4. Klikając „Zarezerwuj”, przesyła dane.
5. System sprawdza dostępność maszyny.
6. Jeśli maszyna jest dostępna – system tworzy rezerwację.

Scenariusz alternatywny:

- (5a) Jeśli maszyna jest zajęta na wybrane daty – system wyświetla komunikat o niedostępności.
- (5b) Jeśli zakres dat przekracza 3 miesiące – system wyświetla komunikat o błędzie.

10.9. Przeglądanie swoich rezerwacji (Użytkownik)

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce zobaczyć swoje aktualne rezerwacje.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /user/reservations.
2. System wyświetla wszystkie jego rezerwacje (maszyna, daty).

10.10. Edycja własnej rezerwacji (Użytkownik)

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce zmodyfikować wcześniej dokonaną rezerwację.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /user/reservations.
2. Klikając na rezerwację, otwiera formularz edycji.
3. Wybiera nowe daty.
4. Klikając „Zapisz zmiany”, przesyła dane.
5. System sprawdza dostępność i aktualizuje rezerwację.

Scenariusz alternatywny:

(4a) Jeśli nowy zakres dat jest zajęty – system odrzuca aktualizację.

10.11. Anulowanie rezerwacji (Użytkownik)

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce usunąć własną rezerwację.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /user/reservations.
2. Klikając na rezerwację, otwiera formularz edycji.
3. Klika „Usuń rezerwację”.
4. System usuwa rezerwację z bazy danych.

10.12. Usunięcie rezerwacji (Administrator)

Aktor: Administrator

Opis: Administrator chce usunąć dowolną rezerwację użytkownika.

Scenariusz główny:

1. Administrator przechodzi na stronę /admin (lista rezerwacji).
2. Klikając przycisk „Usuń” przy rezerwacji, wysyła żądanie usunięcia.
3. System usuwa rezerwację.

10.13. Logowanie użytkownika

Aktor: Użytkownik (admin lub user)

Opis: Użytkownik loguje się do systemu.

Scenariusz główny:

1. Użytkownik przechodzi na stronę /login.
2. Wprowadza nazwę użytkownika i hasło.
3. Klikając „Login”, wysyła dane.
4. Jeśli dane są poprawne – użytkownik jest przekierowany do odpowiedniego panelu.

Scenariusz alternatywny:

(4a) Jeśli dane są niepoprawne – system wyświetla komunikat o błędzie.

10.14. Wylogowanie użytkownika

Aktor: Zalogowany użytkownik

Opis: Użytkownik chce się wylogować.

Scenariusz główny:

1. Użytkownik klika „Wyloguj” w menu.
2. System kończy sesję użytkownika.
3. Użytkownik zostaje przekierowany na stronę logowania.

11. Struktura projektu

- models/
 - index.js – Inicjalizacja modeli i relacji w Sequelize
 - User.js – Model użytkownika, przechowujący dane takie jak imię, nazwisko, login, hasło i rolę
 - Machine.js – Model maszyny rolniczej wraz z opisem i zdjęciem
 - Reservation.js – Model rezerwacji maszyn przez użytkowników
- controllers/
 - adminController.js – Obsługa panelu administratora: zarządzanie maszynami, rezerwacjami i użytkownikami
 - userController.js – Obsługa panelu użytkownika: rezerwowanie maszyn i przeglądanie własnych rezerwacji
 - authController.js – Obsługa logowania, rejestrowania i wylogowania użytkowników
- middleware/

- authMiddleware.js – Middleware sprawdzający, czy użytkownik jest zalogowany oraz czy posiada odpowiednią rolę (user/admin)
- multerMiddleware.js – Middleware konfigurujący Multer do obsługi przesyłania plików (zdjęcia maszyn)
- views/
 - admin/
 - index.ejs – Panel rezerwacji dla administratora
 - machines.ejs – Panel zarządzania maszynami (dodawanie, edytowanie, usuwanie maszyn)
 - users.ejs – Panel zarządzania użytkownikami (dodawanie, edytowanie, usuwanie użytkowników)
 - user/
 - index.ejs – Strona główna użytkownika z listą dostępnych maszyn do rezerwacji
 - reservations.ejs – Lista i edycja własnych rezerwacji użytkownika
 - login.ejs – Formularz logowania użytkownika
- routes/
 - adminRoutes.js – Trasy dla administratora: zarządzanie maszynami, rezerwacjami i użytkownikami
 - userRoutes.js – Trasy dla użytkowników: tworzenie i edytowanie własnych rezerwacji
 - authRoutes.js – Trasy obsługujące logowanie i wylogowywanie
- public/
 - css/
 - style.css – Główny arkusz stylów dla całej aplikacji
 - login.css – Stylizacja strony logowania
 - images/ – Folder przechowujący zdjęcia maszyn przesyłane przez administratora
 - js/
 - admin.js – Skrypt zarządzania rezerwacjami w panelu administratora
 - adminMachines.js – Skrypt zarządzania maszynami w panelu administratora
 - adminUsers.js – Skrypt zarządzania użytkownikami w panelu administratora
 - user.js – Skrypt obsługujący rezerwowanie maszyn przez użytkowników
 - userReservations.js – Skrypt umożliwiający edycję i usuwanie rezerwacji przez użytkowników
- config/

- database.js – Konfiguracja połączenia z bazą danych MySQL za pomocą Sequelize
- multer.js – Konfiguracja Multera (upload zdjęć maszyn do katalogu public/images)
- passport.js – Konfiguracja Passport.js dla uwierzytelniania użytkowników
- session.js – Konfiguracja sesji użytkowników Express-session
- app.js – Główny plik aplikacji Express, inicjalizujący serwer oraz podłączający wszystkie middleware i trasy

Dla tabeli z maszynami w dokumentacji zapisać co rozumiemy przez nazwę tzn podać przykłady Traktor ursus c60 itp

zastanowić się czy opis jest konieczny ale jeśli będzie to warto powiedzieć do czego będzie on wykorzystywany tzn podać przykłady co tam będzie wpisane,

Dla URL trzeba wskazać na co te url będą wskazywać tzn w moim przypadku one będą wskazywać na naszą aplikację tzn dajemy możliwość wgrywania zdjęć do systemu

w tabeli użytkowników hasło musi być zahashowane należy podać jaki algorytm np bcrypt - to również należy dodać do specyfikacji

sposób pisania dat napisać również w specyfikacji

Nie dodawać automatycznego usuwania danych