

Projekt zespołowy systemu informatycznego

Sprint 1, grupa B , raporty ze spotkań *Scrum Meetings*

Sprint Backlog

- task1:** Inicjalizacja środowiska – konfiguracja plików startowych projektu
- task2:** Utworzenie struktury frontendu (Vite + React + Tailwind + routing, App.tsx, main.tsx, konfiguracja routera)
- task3:** Przygotowanie struktury backendu (Express + routery, konfiguracja błędów)
- task4:** Implementacja bazy danych SQLite – konfiguracja typów i schematów (ORM)
- task5:** Konfiguracja GitHub – dodanie repozytorium i pliku README.md
- task6:** Opracowanie przepływu logowania i rejestracji – formularze i logika sesji
- task7:** Widok profilu użytkownika – dane + działające wylogowanie
- task14:** Formularz rezerwacji i logika widoku klienta
- task16:** Konfiguracja systemu do wysyłki wiadomości e-mail – potwierdzenia rezerwacji i pliki ICS
- task17:** Middleware Express Rate-Limit – ograniczenie (1 rezerwacja / 30 s na IP + e-mail)
- task18:** Widok „Moje rezerwacje” / Dashboard użytkownika
- task19:** Middleware autoryzacji API wg ról OWNER/WORKER + kontekstu companyId
- task20:** Panel ustawień firmy – interfejs właściciela
- task32:** Przygotowanie panelu administratora – układ i komponenty

Raport z pierwszego tygodnia realizacji sprintu 1

06.11.2025

Scrum Meeting 1 (27.10.2024)

Na pierwszym spotkaniu rozmawialiśmy o doprecyzowaniu pomysłu i tym, jak poukładać narzędzia pracy zespołowej. Ustaliliśmy, że idziemy w monorepo i omówiliśmy zasady pracy z Jira i GitHubem: jak ma wyglądać projekt i board w Jira (kolumny, workflow, etykiety), jak nazywamy zadania i gałęzie oraz jaki format commitów chcemy przyjąć. Ustaliliśmy też, że repozytorium na GitHubie będzie naszym centralnym miejscem pracy. W części technicznej uzgodniliśmy, że przygotujemy jedynie szkielet: frontend na Vite + React + Tailwind oraz backend na Expressie z ts-node, a w głównym package.json dodamy skrypty do uruchamiania obu serwerów. Uzgodniliśmy minimalne połączenie front-back (CORS i prosty health-check), żeby można było lokalnie włączyć środowisko i płynnie ruszyć z pracami w kolejnych tygodniach.

Scrum Meeting 2 (06.11.2025)

Na drugim spotkaniu skupiliśmy się na tym, jak rozwijać funkcjonalności na bazie przygotowanego szkieletu. Ustaliliśmy, że po stronie backendu potrzebujemy kluczowych middleware'ów (helmet, cors, sesje, parser JSON) oraz logowania żądań, a autoryzację opieramy na sesji: po zalogowaniu przechowujemy w sesji identyfikator, e-mail i rolę (USER/WORKER/OWNER/PLATFORM_ADMIN), żeby móc sprawnie weryfikować uprawnienia. Po stronie frontendu omówiliśmy pełny przepływ logowania i rejestracji: LoginPage z walidacjami, podglądem hasła, komunikatami i przekierowaniem zależnym od roli oraz RegisterPage z walidacją e-maila, wskaźnikiem siły hasła, potwierdzeniem hasła i akceptacją

Projekt zespołowy systemu informatycznego

regulaminu. Uzgodniliśmy też dodanie AuthContext do obsługi sesji i ProtectedRoute do kontroli dostępu oraz automatycznych przekierowań.

Zadania realizowane przez poszczególnych deweloperów:

Rafał Gola:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Inicjalizacja środowiska – konfiguracja plików startowych projektu	4	ukończone
Konfiguracja GitHub, dodanie repozytorium i pliku	2	ukończone

Przemysław Habdas:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Przygotowanie struktury backendu	3	Ukończone
Middleware autoryzacji API wg ról OWNER/WORKER + kontekstu companyId.	4	W realizacji
Middleware Express Rate-Limit (1 rezerwacja/30 s per IP + e-mail).	2	W realizacji

Kamil:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Utworzenie struktury frontendu (Vite + React + Tailwind, `App.tsx`, `main.tsx`, konfiguracja routera)	5	Ukończone

Jakub:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Opracowanie przepływu logowania i rejestracji – koncepcja widoków i logiki sesji`	2	Ukonczone
Widok profilu użytkownika z danymi i działającym wylogowaniem.	4	W realizacji

Daniel:

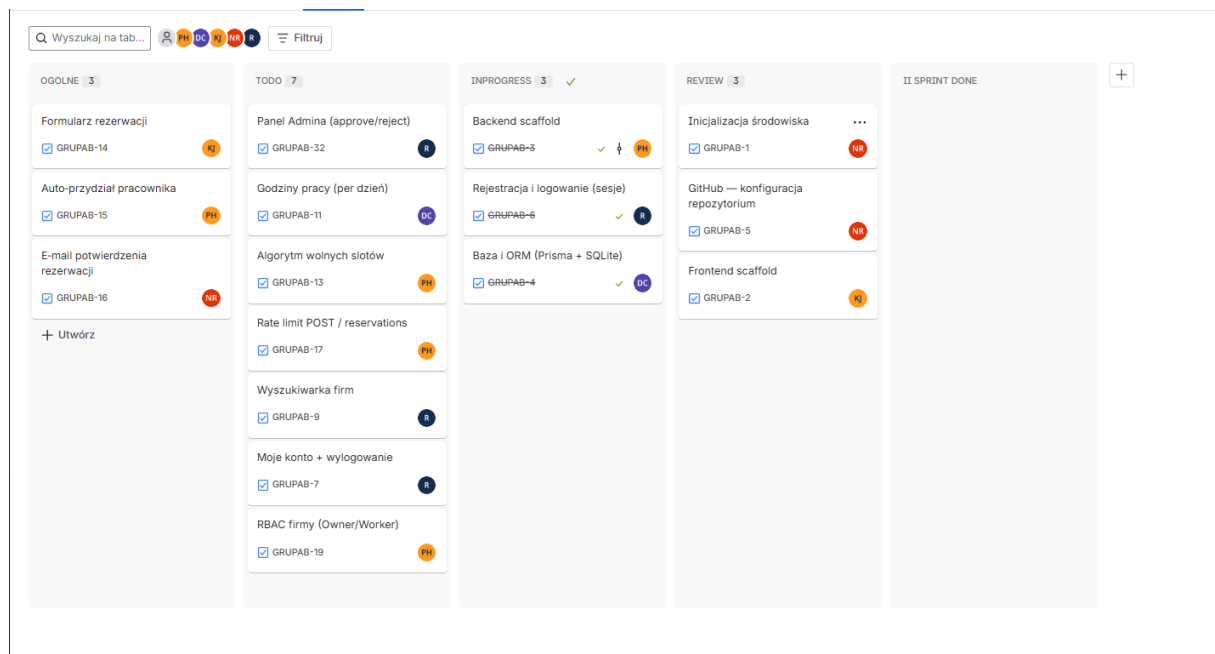
Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Wstępna struktura plików pod przyszłą bazę i ORM	2	Ukonczone

Projekt zespołowy systemu informatycznego

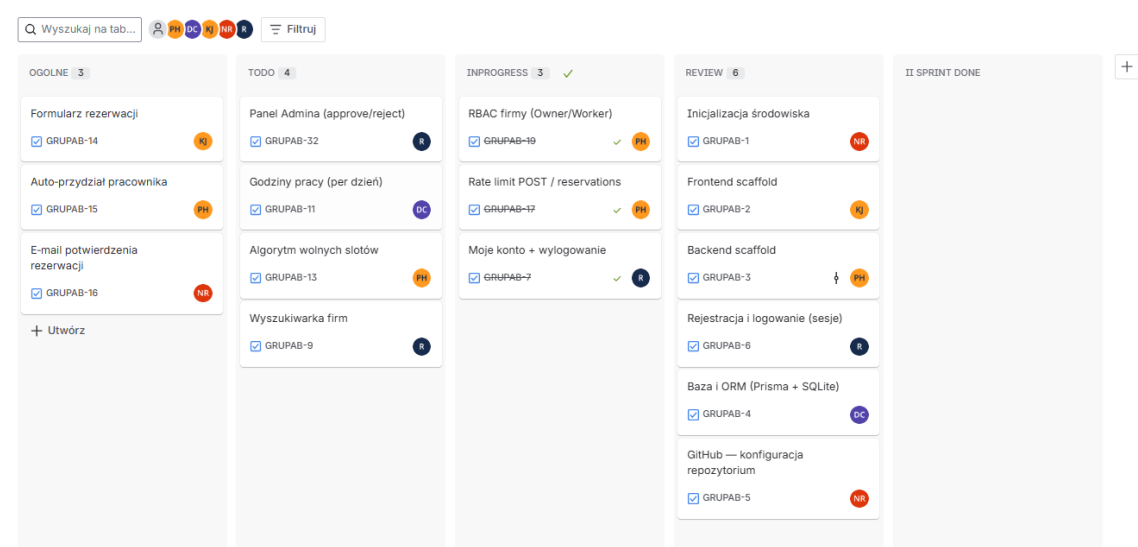
Cele osiągnięte w bieżącym tygodniu

- Ustaliliśmy strukturę projektu w architekturze monorepo (frontend + backend).
- Skonfigurowaliśmy środowisko deweloperskie i repozytorium GitHub.
- Uruchomiliśmy podstawowy szkielet aplikacji (Vite + React + Tailwind + Express).
- Ustaliliśmy zasady pracy zespołowej w Jira oraz format commitów.
- Połączyliśmy frontend z backendem testowym endpointem (health-check).
- Przygotowaliśmy wspólne pliki konfiguracyjne TypeScript.

Stan tablicy Kanban na dzień 27.10.2024



Stan tablicy Kanban na dzień 06.11.2025



Projekt zespołowy systemu informatycznego

Commy z Githuba – stan na dzień 06.11.2025

Fix missing newline at end of AdminDashboard.tsx	kkkejus pushed 1 commit • db3a1c4...76500f • yesterday	...
Fix missing newline at end of RegisterPage.tsx	kkkejus pushed 1 commit • fd2782d...db3a1c4 • yesterday	...
Fix missing newline at end of LoginPage.tsx	kkkejus pushed 1 commit • 3f23bc8...fd2782d • yesterday	...
Refactor HomePage component layout and content	kkkejus pushed 1 commit • 74d1800...3f23bc8 • yesterday	...
Implement ProtectedRoute component for route protection	kkkejus pushed 1 commit • d3381cb...74d1800 • yesterday	...
Add Logo component with customizable properties	kkkejus pushed 1 commit • bf94902...d3381cb • yesterday	...
Implement routing with React Router	kkkejus pushed 1 commit • 48390aa...bf94902 • yesterday	...
Refactor App component with navigation and footer	kkkejus pushed 1 commit • 336fd0b...48390aa • yesterday	...
Add main.tsx to initialize React application	kkkejus pushed 1 commit • 4517e0b...336fd0b • yesterday	...
Add initial HTML structure for frontend	kkkejus pushed 1 commit • bf7bd4d...4517e0b • yesterday	...
Add companies route with CRUD operations	x-PriMo pushed 1 commit • d8142e...bf7bd4d • 3 days ago	...
Refactor auth routes to use local db methods

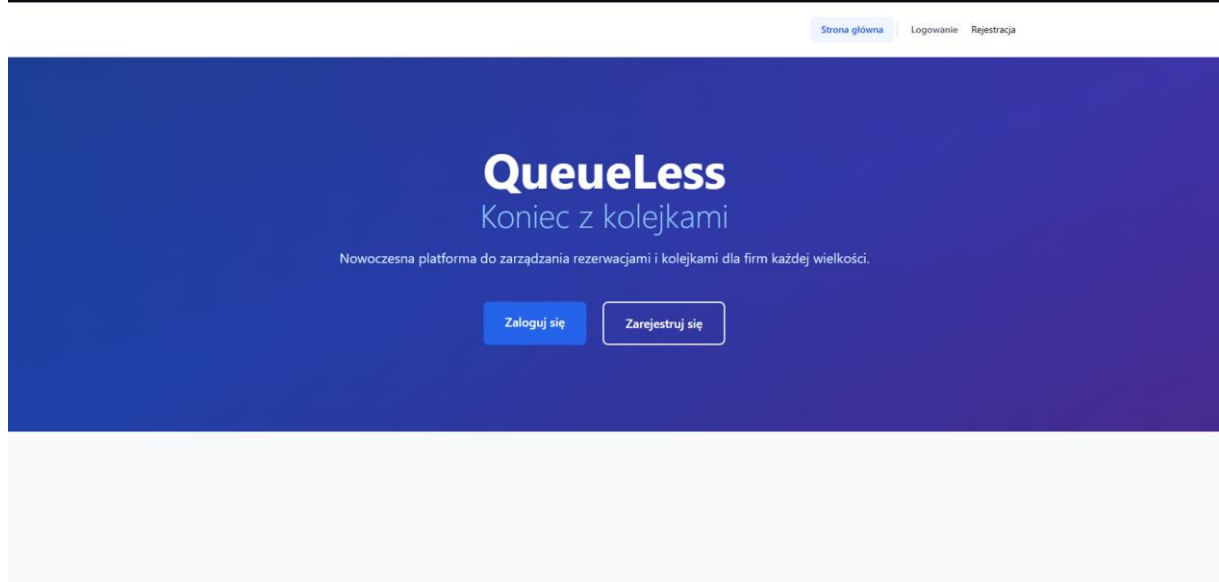
Refactor LoginPage component for improved structure	Kubaaa04 pushed 1 commit • cc7a7a...a50030d • 3 days ago	...
Update README.md	Rafaue pushed 1 commit • 3d1c4d0...cc7a7a • 3 days ago	...
Add PostCSS configuration with Tailwind CSS	Rafaue pushed 1 commit • c1ef1e4...3d1c4d0 • 3 days ago	...
Modify Tailwind config for content and plugins	Rafaue pushed 1 commit • 16e7ebd...c1ef1e4 • 3 days ago	...
Add Tailwind CSS configuration file	Rafaue pushed 1 commit • efdc834...16e7ebd • 4 days ago	...
Add Vite configuration for React app with proxy	Rafaue pushed 1 commit • a632a52...efdc834 • 4 days ago	...
Add TypeScript configuration for frontend package	Rafaue pushed 1 commit • 1e91067...a632a52 • 4 days ago	...
Create tsconfig.json for backend package	Rafaue pushed 1 commit • 2a59866...1e91067 • 4 days ago	...
Add package.json for frontend with dependencies	Rafaue pushed 1 commit • b8ef39e...2a59866 • 4 days ago	...
Initialize package.json for backend project	Rafaue pushed 1 commit • 9c3066f...b8ef39e • 4 days ago	...
Add entries to .gitignore file	Rafaue pushed 1 commit • 45f14c2...9c3066f • 4 days ago	...
Add Prettier configuration file	Rafaue pushed 1 commit • 1e45645...45f14c2 • 4 days ago	...
Add ESLint configuration for TypeScript project	Rafaue pushed 1 commit • 330d594...1e45645 • 4 days ago	...
Add tsconfig.base.json with compiler options

Projekt zespołowy systemu informatycznego

Add ESLint configuration for TypeScript project	...
Rafaue pushed 1 commit • 330d594...1e45645 • 4 days ago	
Add tsconfig.base.json with compiler options	...
Rafaue pushed 1 commit • 54fc793...330d594 • 4 days ago	
Create package-lock.json	...
Rafaue pushed 1 commit • 955c123...54fc579 • 4 days ago	
Refactor package.json for monorepo structure	...
Rafaue pushed 1 commit • 306e032...955c123 • 4 days ago	
GRUPAB-3: Backend scaffold	...
x-PriMo pushed 1 commit • c53dda4...306e032 • 9 days ago	
Create reservationService.ts	...
Dan69Dan pushed 1 commit • c628cd4...c53dda4 • 9 days ago	
Create shifts.ts	...
Dan69Dan pushed 1 commit • 6b3ef9e...c628cd4 • 9 days ago	
Add ReservationsPage component for managing reservations	...
kkkejkus pushed 1 commit • 7e14bd6...6b3ef9e • 9 days ago	
Add HomePage component with features and CTA sections	...
kkkejkus pushed 1 commit • 3dea901...7e14bd6 • 9 days ago	
Initialize App component with routing setup	...
kkkejkus pushed 1 commit • aada48f...3dea901 • 9 days ago	
Create seed.ts	...
Dan69Dan pushed 1 commit • a1087e3...aada48f • 9 days ago	
Add initial package.json configuration	...
Rafaue pushed 1 commit • d742dd7...a1087e3 • 9 days ago	
Create schema.prisma	...
Dan69Dan pushed 1 commit • ec02e53...d742dd7 • 9 days ago	
Add CI workflow configuration for QueueLess project	...
Rafaue pushed 1 commit • c881226...ec02e53 • 10 days ago	

Implementowane widoki aplikacji


Ekran startowy:



Projekt zespołowy systemu informatycznego

Ekran logowania:

[Strona główna](#) [Logowanie](#) [Rejestracja](#)



Zaloguj się


Witaj ponownie! Zaloguj się do swojego konta

Adres e-mail


@

np.jan.kowalski@example.com

Hasło



Wprowadź swoje hasło



☐ Zapamiętaj mnie

[Zapomniałeś hasła?](#)

Zaloguj się

Nie masz jeszcze konta? [Zarejestruj się](#)

Konta demo:

Admin: admin@admin.com / admin123

Projekt zespołowy systemu informatycznego

Ekran Rejestracji:

Utwórz konto
Dołącz do QueueLess i zacznij zarządzać kolejkami

Adres e-mail
ddd@ddd.doc

Hasło
Dobre

Potwierdź hasło

☒ Akceptuję regulamin i politykę prywatności

Utwórz konto

Failed to fetch

Masz już konto? Zaloguj się

Korzyści z QueueLess:
✓ Zarządzanie kolejkami online

Kod źródłowy – istotne fragmenty opracowane w bieżącym tygodniu

```
QueueLess > packages > backend > src > routes > TS auth.ts > router.post('/register') callback
12
13 router.post('/register', async (req, res) => {
14   const parsed = registerSchema.safeParse(req.body);
15   if (!parsed.success) return res.status(400).json({ error: 'Invalid input' });
16
17   const { email, password } = parsed.data;
18
19   // Sprawdź czy użytkownik już istnieje
20   const existingUser = db.getUserByEmail(email);
21   if (existingUser) return res.status(409).json({ error: 'Email already registered' });
22
23   // Hashuj hasło i utwórz użytkownika
24   const hash = await bcrypt.hash(password, 10);
25   const user = db.createUser({
26     email,
27     password: hash,
28     role: 'USER',
29     isActive: true
30   });
31
```

Ten fragment kodu obsługuje rejestrację użytkownika w aplikacji. Sprawdza poprawność danych wejściowych, weryfikuje, czy podany e-mail nie jest już zarejestrowany, a następnie

Projekt zespołowy systemu informatycznego

hashuje hasło za pomocą bcrypts i tworzy nowego użytkownika z rolą „USER” i statusem aktywnym.

```
QueueLess > packages > frontend > src > components > ProtectedRoute.tsx > ProtectedRoute

10
11 export default function ProtectedRoute({
12   children,
13   requiredRoles = [],
14   requireAuth = true
15 }: ProtectedRouteProps) {
16   const { user, isLoading } = useAuth();
17
18   if (isLoading) {
19     return (
20       <div className="min-h-screen flex items-center justify-center">
21         <div className="text-center">
22           <div className="brand-spinner mx-auto mb-4"></div>
23           <p className="text-gray-600">Ładowanie...</p>
24         </div>
25       </div>
26     );
27   }
28
29   // Jeśli wymaga uwierzytelnienia, ale użytkownik nie jest zalogowany
30   if (requireAuth && !user) {
31     return <Navigate to="/login" replace />;
32   }
33
34   // Jeśli nie wymaga uwierzytelnienia i użytkownik jest zalogowany, przekieruj do dashboardu
35   if (!requireAuth && user) {
36     return <Navigate to={getDashboardPath(user.role)} replace />;
37   }
38
39   // Jeśli są wymagane konkretne role
40   if (requiredRoles.length > 0 && user && !requiredRoles.includes(user.role)) {
41     return <Navigate to="/unauthorized" replace />;
42   }
43
44   return <>{children}</>;
45 }
```

Ten fragment kodu odpowiada za ochronę tras w aplikacji — komponent ProtectedRoute sprawdza, czy użytkownik jest zalogowany i czy ma odpowiednią rolę dostępu. W zależności od wyniku przekierowuje go na stronę logowania, dashboard lub „unauthorized”, a jeśli wszystko jest poprawne, wyświetla żądany widok.

Projekt zespołowy systemu informatycznego

Raport z drugiego tygodnia realizacji sprintu 1

10.11.2025

Na meetingu 3 skoncentrowaliśmy się na tym, żeby logowanie działało po prostu bez stresu. Zaczęliśmy od przejrzenia ekranu i konsoli, bo część osób widziała komunikat „Not found”. Potem sprawdziliśmy „ręcznie” połączenie z serwerem i szybko okazało się, że aplikacja woła pod inny adres niż ten, którego oczekuje backend. Ustaliliśmy prostą, konkretną poprawkę: przepisać ścieżkę w ustawieniach frontu tak, żeby trafiała we właściwe miejsce, zrestartować oba serwery i sprawdzić jeszcze raz cały przepływ. Zrobiliśmy krótki test po zmianie, żeby mieć pewność, że sesja się poprawnie zakłada, a użytkownik dostaje czytelną informację. Przy okazji uporządkowaliśmy detale na ekranie logowania: poprawne dane demo, prostsze komunikaty w błędnych przypadkach i sensowne podpowiedzi co wpisać. Ustaliliśmy też, że w najbliższych dniach ruszymy z pierwszymi „widocznymi” elementami — listą firm i szkicem panelu admina — tak, żeby każdy mógł zobaczyć postęp bez zaglądania w technikalnia.

14.11.2025

Na meetingu 4 usiedliśmy do podziału ról i zaplanowaliśmy, jak ma wyglądać pierwsza, lekka wersja panelu administratora. Chcieliśmy, żeby była prosta w obsłudze: lista firm, widoczny status firmy (aktywna/wyłączona) i kilka przełączników, które od razu dają poczucie kontroli. Ustaliliśmy też „bezpieczniki” — żeby nikt nie mógł męczyć logowania w nieskończoność i żeby dane, które wpisujemy, były od razu sensownie sprawdzane. Zdecydowaliśmy, że przygotujemy podstawowy zestaw danych startowych (admin, jedna przykładowa firma, pracownik), dzięki czemu demo będzie od pierwszego dnia wyglądać „jak działa”. Doprecyzowaliśmy, co dla nas znaczy „zrobione” w tej części (czyli jasna definicja gotowości), a na koniec ułożyliśmy prosty plan kontroli jakości: czy da się bez problemu wejść, czy sesja trzyma, czy komunikaty są zrozumiałe i czy aplikacja zachowuje się grzecznie w typowych błędach. Na bazie tego zrobiliśmy krótki plan następnego kroku: widok listy firm na froncie, prosty profil firmy z przyciskiem „Zarezerwuj”, oraz pierwszy ekran „kolejki dnia” dla pracownika, gdzie można w prosty sposób zacząć i zakończyć wizytę. Dla użytkownika końcowego chcemy dorzucić przyjazne wiadomości (np. „sprawdź e-mail lub hasło”), kilka drobnych udogodnień na telefonie i szybkie potwierdzenie rezerwacji e-mailem z linkiem do kalendarza, żeby całość czuła się „gotowa do użycia”.

Projekt zespołowy systemu informatycznego

Zadania realizowane przez poszczególnych deweloperów:

Rafał Gola:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Konfiguracja GitHub oraz utworzenie repozytorium i pipeline CI (T5)	2	Ukończone
Konfiguracja systemu do wysyłki wiadomości e-mail – potwierdzenia rezerwacji i ICS (T16)	3	W realizacji

Przemysław Habdas:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Przygotowanie struktury backendu (Express + routery, konfiguracja błędów) (T3)	2	Ukończone
Implementacja middleware autoryzacji w rolach OWNER/WORKER i kontekstu companyId (T19)	3	W realizacji
Middleware Express Rate-Limit (autoryzacja/rezerwacje) (T17)	2	W realizacji

Kamil:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Utworzenie struktury frontendu (Vite + React + Tailwind + routing) (T2)	5	Ukończone
Formularz rezerwacji i logika widoku klienta (T14)	4	W realizacji
Widok „Moje rezerwacje” / Dashboard użytkownika (T18)	3	W realizacji
Panel ustawień firmy – interfejs właściciela (T20)	4	W realizacji

Jakub:

Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Opracowanie przepływu logowania i rejestracji – formularze i logika sesji (T6)	2	Ukończone
Widok profilu użytkownika – dane + wylogowanie (T7)	2	W realizacji
Przygotowanie panelu administratora – układ i komponenty (T32, backlog Sprintu 2)	4	Zaplanowane

Daniel:

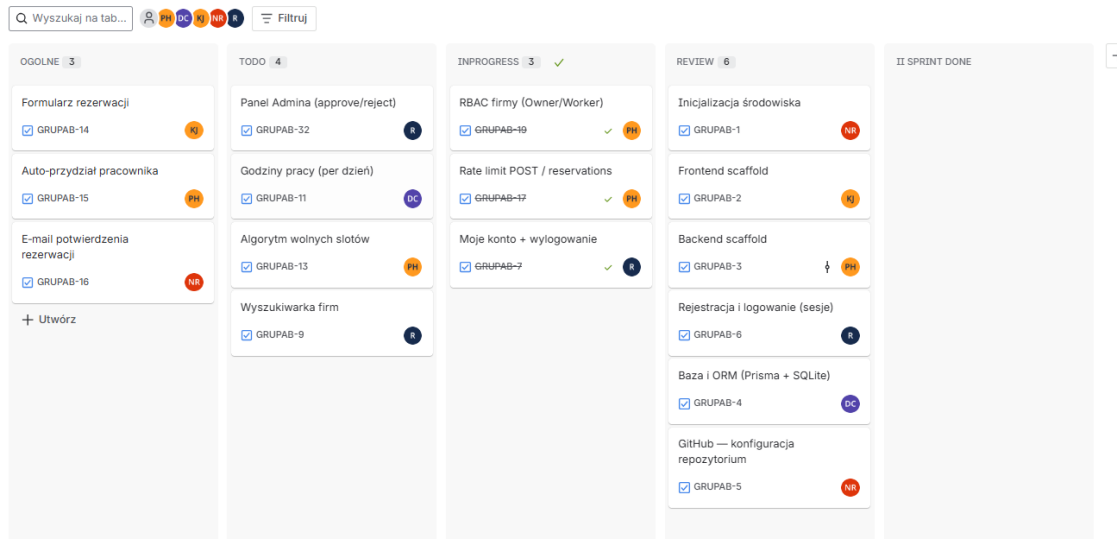
Realizowane zadanie	Tygodniowy czas pracy [h]	Status realizowanego zadania
Implementacja bazy danych SQLite– konfiguracja typów i schematów (T4)	4	W realizacji
Migracje i pliki inicjalizujące strukturę tabel oraz dane przykładowe (T4)	2	Ukończone

Projekt zespołowy systemu informatycznego

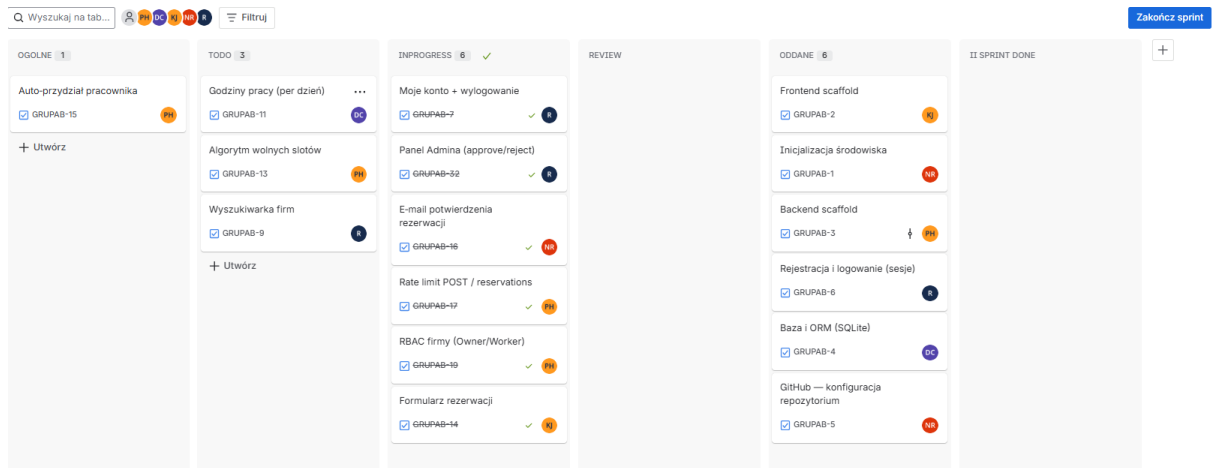
Cele osiągnięte w bieżącym tygodniu

- Zainicjalizowaliśmy projekt w architekturze monorepo (frontend + backend).
- Skonfigurowaliśmy środowisko programistyczne oraz repozytorium GitHub z działającym CI.
- Utworzyliśmy i połączyliśmy bazę danych SQLite.
- Zaimplementowaliśmy system logowania i rejestracji użytkowników (formularze + sesje).
- Przygotowaliśmy wspólne pliki konfiguracyjne TypeScript, ESLint, Prettier.
- Ustaliliśmy zasady pracy zespołowej i strukturę commitów w repozytorium.

Stan tablicy Kanban na dzień 10.11.2025



Stan tablicy Kanban na dzień 14.11.2025





















Commity z Githuba – stan na dzień 14.11.2025

Projekt zespołowy systemu informatycznego

Fix missing newline at end of UnauthorizedPage.tsx	...
Kubaaa04 pushed 1 commit to main • d69093c...5e82d70 • yesterday	
Delete packages/frontend/src/pages/UnauthorizedPage	...
Kubaaa04 pushed 1 commit to main • 6d47b56...d69093c • yesterday	
Create UnauthorizedPage component for access control	...
Kubaaa04 pushed 1 commit to main • a902c4e...6d47b56 • yesterday	
Update RegisterPage.tsx	...
Kubaaa04 pushed 1 commit to main • f35ece0...a902c4e • yesterday	
Update demo account credentials on LoginPage	...
Kubaaa04 pushed 1 commit to main • 88f106b...f35ece0 • yesterday	
Refactor mailer to use Ethereum test account	...
Rafaue pushed 1 commit to main • f04ce9...88f106b • yesterday	
Update TypeScript config to include db directory	...
Rafaue pushed 1 commit to main • f47a695...f04ce9 • yesterday	
Update package.json for backend configuration	...
Rafaue pushed 1 commit to main • 9abff1...f47a695 • yesterday	
Add imports for migration and seeding scripts	...
Rafaue pushed 1 commit to main • 4e60522...9abff1 • yesterday	

Rename path	...
Dan69Dan pushed 1 commit to main • 28f0105...dc76cde • 5 hours ago	
Create seed.ts	...
Dan69Dan pushed 1 commit to main • be13edc...28f0105 • 5 hours ago	
Create 001_init.sql	...
Dan69Dan pushed 1 commit to main • 5ca5fd0...be13edc • 5 hours ago	
Update db.ts	...
Dan69Dan pushed 1 commit to main • 75125bd...5ca5fd0 • 5 hours ago	
Create migrate.ts	...
Dan69Dan pushed 1 commit to main • b2c3532...75125bd • 5 hours ago	
Create .gitkeep	...
kkkejulus pushed 1 commit to main • 88f933a...b2c3532 • 6 hours ago	
Implement Worker Dashboard component	...
kkkejulus pushed 1 commit to main • 4e78625...88f933a • 6 hours ago	
Add OwnerDashboard component for managing companies	...
kkkejulus pushed 1 commit to main • 860f82d...4e78625 • 6 hours ago	
Add UserDashboard component for managing reservations	...
kkkejulus pushed 1 commit to main • f649639...860f82d • 6 hours ago	
Add QueuePage component to display queue status	...
kkkejulus pushed 1 commit to main • 437a408...f649639 • 6 hours ago	
Create CompanySettingsPage.tsx	...
kkkejulus pushed 1 commit to main • d9f927a...437a408 • 6 hours ago	
Add ReservationPage component for booking process	...
kkkejulus pushed 1 commit to main • f9e4858...d9f927a • 6 hours ago	
Ensure root elements have full height	...
kkkejulus pushed 1 commit to main • 76f9305...f9e4858 • 6 hours ago	
Fix CSS syntax for body-responsive class	...
kkkejulus pushed 1 commit to main • c480537...76f9305 • 6 hours ago	
Revise homepage text and links for clarity	...
kkkejulus pushed 1 commit to main • f81f5b1...c480537 • 6 hours ago	
Fix formatting in ProtectedRoute component	...
kkkejulus pushed 1 commit to main • fc19fc3...f81f5b1 • 6 hours ago	
Fix formatting in Logo.tsx component	...
kkkejulus pushed 1 commit to main • cb73695...fc19fc3 • 6 hours ago	
Add protected routes for reservation and dashboards	...
kkkejulus pushed 1 commit to main • 671d571...cb73695 • 6 hours ago	

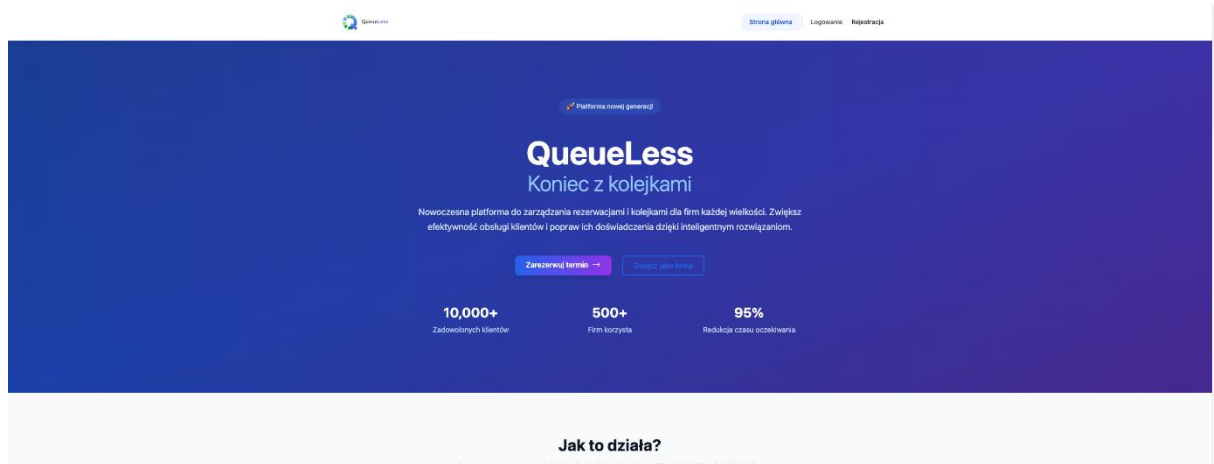
Projekt zespołowy systemu informatycznego

Ensure root elements have full height	...
 kikkejus pushed 1 commit to main • 769305...f9e4858 • 6 hours ago	
Fix CSS syntax for body-responsive class	...
 kikkejus pushed 1 commit to main • c480537...769305 • 6 hours ago	
Revise homepage text and links for clarity	...
 kikkejus pushed 1 commit to main • f81f5b1...c480537 • 6 hours ago	
Fix formatting in ProtectedRoute component	...
 kikkejus pushed 1 commit to main • fc19fc3...f81f5b1 • 6 hours ago	
Fix formatting in Logo.tsx component	...
 kikkejus pushed 1 commit to main • cb73695...fc19fc3 • 6 hours ago	
Add protected routes for reservation and dashboards	...
 kikkejus pushed 1 commit to main • 671d571...cb73695 • 6 hours ago	
Update user role handling in dashboard path	...
 kikkejus pushed 1 commit to main • afe7a85...671d571 • 6 hours ago	
Fix formatting in main.tsx	...
 kikkejus pushed 1 commit to main • 7c25ab2...afe7a85 • 6 hours ago	
Fix missing newline at end of index.html	...
 kikkejus pushed 1 commit to main • 44ad662...7c25ab2 • 6 hours ago	
Refactor reservations route with Zod and direct DB access	...
 x-PrIMo pushed 1 commit to main • 9b34ee8...44ad662 • 7 hours ago	
Implement rate limiting middleware	...
 x-PrIMo pushed 1 commit to main • 6cca462...9b34ee8 • 7 hours ago	
Refactor authentication routes with database queries	...
 x-PrIMo pushed 1 commit to main • 31ab363...6cca462 • 7 hours ago	
Enhance company routes with timezone and logo upload	...
 x-PrIMo pushed 1 commit to main • da86224...31ab363 • 7 hours ago	
Add reservations routes to the backend	...
 x-PrIMo pushed 1 commit to main • a9b37a7...da86224 • 7 hours ago	
Add user management and loading functionality	...
 Kubaaa04 pushed 1 commit to main • a61ec49...a9b37a7 • yesterday	
Handle error response in API calls	...
 Kubaaa04 pushed 1 commit to main • c93d49b...a61ec49 • yesterday	
Fix missing newline at end of AuthContext.tsx	...
 Kubaaa04 pushed 1 commit to main • 22c2e1d...c93d49b • yesterday	
Fix missing newline at end of AuthContext.tsx	...
 Kubaaa04 pushed 1 commit to main • 5e82d70...22c2e1d • yesterday	

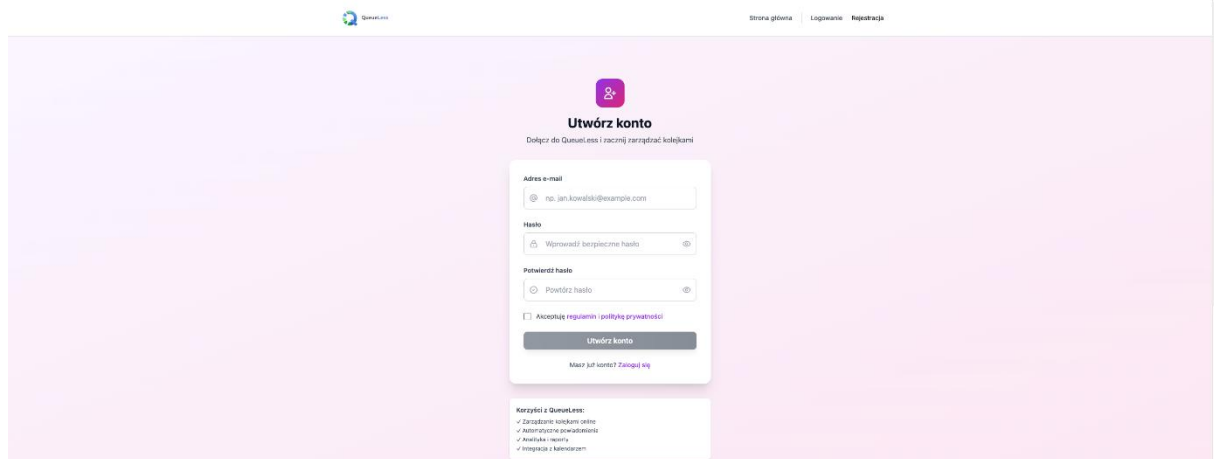
Projekt zespołowy systemu informatycznego

Implementowane widoki aplikacji

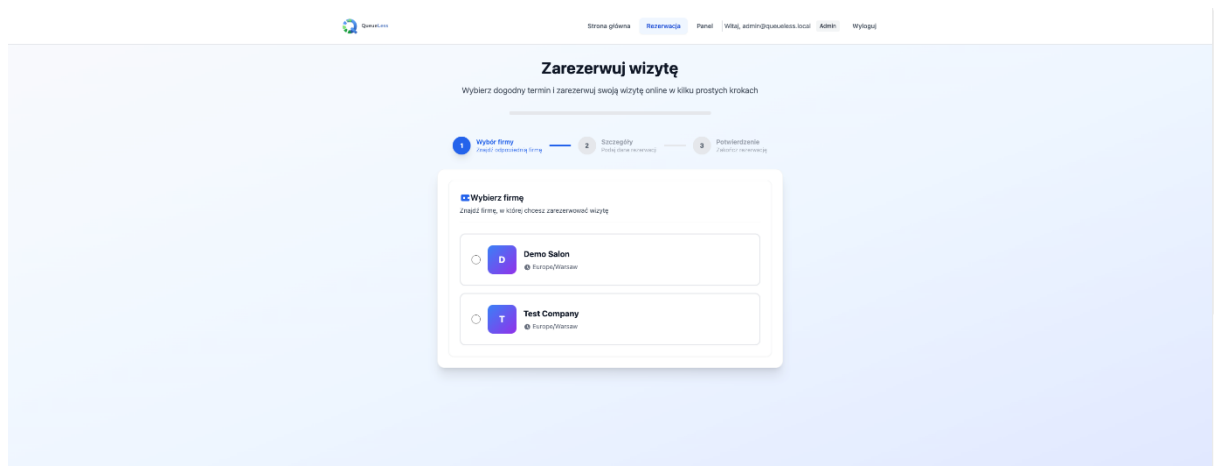
Strona główna:



Zakładanie konta użytkownika:



Strona rezerwacji wizyty:



Projekt zespołowy systemu informatycznego

Panel Administratora (Dla klienta prowadzącego firmę):

Queueless

Strona głównaRezerwacjaPanelWitaj, admin@queueless.localAdminWyloguj

Panel administratora

Witaj, admin@queueless.local! Zarządzaj firmami i użytkownikami platformy Queueless.

Firmy2

Użytkownicy0

Aktywne Firmy0

Aktywni użytkownicy0

Dodaj nową firmę

Nazwa firmy

Wprowadź nazwę firmy

Dodaj firmę

Kategoria (opcjonalnie)

np. Salon fryzjerski

Zarządzanie firmami

NAZWA	KATEGORIA	STATUS	DATA USTWÓRZENIA	
Demo Salon	-	Nieaktywna	Invalid Date	Aktywny
Test Company	-	Nieaktywna	Invalid Date	Aktywny

Zarządzanie użytkownikami

Brak danych o użytkownikach (endpoint może wymagać implementacji)

Panel logowania:

Queueless

Strona głównaLogowanieRejestracja

Zaloguj się

Witaj ponownie! Zaloguj się do swojego konta

Adres e-mail

np. jan.kowalski@example.com

Hasło

Wprowadź swoje hasło

☐ Zapamiętaj mnieZapomniałeś hasło?

Zaloguj się

Nie masz jeszcze konta? Zarejestruj się

Konto demo:

Adres: admin@queueless.local / admin123

Wskazówki: admin@queueless.local / admin123

Projekt zespołowy systemu informatycznego

Kod źródłowy – istotne fragmenty opracowane w bieżącym tygodniu

```
packages > backend > src > middleware > TS rateLimit.ts > ...
1  import { Request, Response, NextFunction } from 'express';
2
3  type Keyer = (req: Request) => string;
4
5  interface Rule {
6    windowMs: number;
7    max: number;
8    keyer: Keyer;
9  }
10
11  const store = new Map<string, { count: number; resetAt: number }>();
12
13  export function rateLimit(rule: Rule) {
14    return (req: Request, res: Response, next: NextFunction) => {
15      const key = rule.keyer(req);
16      const now = Date.now();
17      const entry = store.get(key);
18      if (!entry || now > entry.resetAt) {
19        store.set(key, { count: 1, resetAt: now + rule.windowMs });
20        return next();
21      }
22      if (entry.count >= rule.max) {
23        return res.status(429).json({ error: 'Too many requests' });
24      }
25      entry.count += 1;
26      next();
27    };
28  }
```

Kod dostarcza funkcję `getTransport()` tworzącą transporter SMTP przez `nodemailer`.

W trybie `dev` automatycznie zakłada konto Ethereum i zwraca gotowy transporter (`host smtp.ethereal.email`, `port 587`, `secure: false`).

W bloku `catch` zwracany jest „no-op” transport z metodą `sendMail`, który nic nie robi, dzięki czemu wywołania mailera nie wywołają aplikacji bez sieci/SMTP.

Taki fallback ułatwia lokalny rozwój i testy/CI — kod wysyłki można wołać bez warunków na środowisko.

Projekt zespołowy systemu informatycznego

```
packages > backend > src > lib > TS mailer.ts > getTransport

1  import nodemailer from 'nodemailer';
2
3  export async function getTransport() {
4    try {
5      // W dev korzystamy z konta testowego Ethereum
6      const testAccount = await nodemailer.createTestAccount();
7      const transporter = nodemailer.createTransport({
8        host: 'smtp.ethereal.email',
9        port: 587,
10       secure: false,
11       auth: {
12         user: testAccount.user,
13         pass: testAccount.pass,
14       },
15     });
16     return transporter;
17   } catch (e) {
18     // Brak sieci → zwróć obiekt transportu, który nic nie robi
19     return {
20       sendMail: async () => ({}),
21     } as any;
22   }
}
```

Kod implementuje prosty, generyczny rate-limiter: funkcja `rateLimit(rule)` przyjmuje okno czasowe (`windowMs`), limit (`max`) i funkcję klucza (`keyer`). Stan trzymany jest w `Map` , resetowany po wygaśnięciu okna; przy przekroczeniu zwracany jest HTTP 429.

Gotowe reguły:

`loginRateLimit` : 5 prób logowania na minutę per IP.

`reservationRateLimit` : 1 rezerwacja na 30 s per IP+email (lowercase).