© KOMPLEKSOWY PLAN DZIAŁANIA - FRONTEND DEVELOPMENT

Fieldworker Multi-Sector SaaS

Data opracowania: 8 września 2025

Status backend: 370/370 testów przechodzi - 22/22 kontrolerów API kompletnych

Cel: Stworzenie nowoczesnego, responsywnego interfejsu użytkownika



Tech Stack:

• Framework: React 18 + TypeScript

• Backend Integration: Laravel Inertia.js

• Styling: Tailwind CSS + Headless UI

• Icons: Heroicons + React Icons

• State Management: React Query + Zustand

• Forms: React Hook Form + Zod validation

• Charts: Chart.js / Recharts

• Maps: Leaflet (dla lokalizacji pracowników)

Architektura:

```
Frontend/
                  # Inertia pages (główne widoki)
─ Pages/
 — Components/
                  # Reusable components
 — Layouts/
                  # Layout templates
  - Hooks/
                 # Custom React hooks
 — Types/
                 # TypeScript definitions
                 # Helper functions
 - Utils/
  - Stores/
                 # Zustand stores
└── Styles/
                  # Tailwind config + custom CSS
```

STRUKTURA APLIKACJI - USER INTERFACE

1. DASHBOARD GŁÓWNY

Priorytet: WYSOKI - Pierwszy ekran po logowaniu

Layout i Komponenty:

```
Dashboard/
├─ TopNavigation  # Search, notifications, user menu
├─ Sidebar  # Main navigation menu
├─ StatsCards  # KPI cards (workers, jobs, assets)
├─ ChartsSection  # Activity charts, utilization
├─ RecentActivity  # Latest activities feed
└─ QuickActions  # Shortcuts to common tasks
```

Funkcjonalności:

• 🔟 Statystyki w czasie rzeczywistym

- Liczba aktywnych pracowników
- o Zadania w toku / ukończone
- Wykorzystanie zasobów
- Przychody / koszty (basic)

• Wykresy i wizualizacje

- Activity timeline
- Worker performance charts
- Asset utilization graphs
- Sector statistics

• D Live updates

- Real-time notifications
- Auto-refresh data
- WebSocket integration (opcjonalnie)

2. ZARZĄDZANIE PRACOWNIKAMI

Priorytet: WYSOKI - Core functionality

2.1 Lista Pracowników

```
Workers/
├─ WorkersTable # Sortable, filterable table
├─ SearchFilters # Advanced filtering
├─ BulkActions # Mass operations
├─ ExportTools # CSV/PDF export
└─ WorkerCard # Individual worker preview
```

Funkcjonalności:

• 🔾 Zaawansowane wyszukiwanie

- o Filtrowanie po umiejętnościach
- Status (aktywny/nieaktywny)
- o Lokalizacja
- Certyfikaty

• 🖺 Operacje grupowe

- Przypisywanie zadań
- Aktualizacja statusu
- Export danych

2.2 Profil Pracownika

```
WorkerProfile/

├── PersonalInfo # Basic info + photo
├── SkillsSection # Skills with levels
├── CertificationsTab # Active/expired certs
├── JobsHistory # Past and current jobs
├── PerformanceTab # Stats and ratings
└── DocumentsTab # Attachments and signatures
```

Funkcjonalności:

- 📝 Edycja profilu inline editing
- **Zarządzanie certyfikatami** upload, renewal alerts
- Tracking lokalizacji mapa z historia
- Performance metrics completion rates, ratings

🖉 3. ZARZĄDZANIE ZADANIAMI

Priorytet: WYSOKI - Business core

3.1 Lista Zadań

```
Jobs/
├─ JobsKanban # Kanban board view
├─ JobsTable # Traditional table view
├─ JobsCalendar # Calendar view
├─ JobFilters # Status, priority, date filters
└─ CreateJobModal # Quick job creation
```

Widoki:

- 🖺 Kanban Board Do zrobienia, W toku, Ukończone
- **Calendar View** Zadania w kalendarzu
- **Table View** Szczegółowa tabela z sortowaniem

3.2 Szczegóły Zadania

```
JobDetails/

— JobHeader # Title, status, priority

— JobInfo # Description, location, dates

— AssignedWorkers # Workers with roles

— RequiredSkills # Skills and certifications needed

— ProgressTracker # Task completion status

— DocumentsTab # Forms, attachments, signatures

— TimelineTab # Activity log

— CommentsSection # Team communication
```

Funkcjonalności:

- Przypisywanie pracowników drag & drop
- Formularze dynamiczne custom fields per job type
- **P** Lokalizacja mapa z pinami
- Start/stop timers

4. SYSTEM FORMULARZY

Priorytet: SREDNI - Unique selling point

4.1 Constructor Formularzy

```
FormBuilder/

|-- FieldPalette  # Drag & drop field types
|-- FormCanvas  # Visual form builder
|-- FieldProperties  # Field configuration panel
|-- PreviewMode  # Live form preview
|-- PublishSettings  # Form deployment options
```

Typy pól:

- **Text inputs** single/multi line
- **Checkboxes** single/multiple
- Radio buttons single choice
- **Date pickers** date/time/datetime
- File uploads images, documents
- Signatures digital signature capture
- **P** Location GPS coordinates
- Rating scales 1-5 stars, 1-10 scale

4.2 Wypełnianie Formularzy

```
FormFilling/

— ProgressIndicator # Form completion progress

— DynamicFields # Conditional field showing

— ValidationErrors # Real-time validation

— DraftSaving # Auto-save drafts

— SubmissionConfirm # Confirmation screen
```

Funkcjonalności:

- 🖺 Auto-save nie można stracić danych
- 🔍 Walidacja w czasie rzeczywistym
- Mobile-first touch-friendly
- Conditional logic show/hide fields based on answers

5. ZARZĄDZANIE KLIENTAMI (TENANTS)

Priorytet: SREDNI - Admin functionality

5.1 Lista Klientów

```
Tenants/
├─ TenantCards # Grid view with key metrics
├─ TenantTable # Detailed table view
├─ SearchFilters # Name, status, plan filters
└─ CreateTenant # New tenant setup wizard
```

5.2 Szczegóły Klienta

```
TenantDetails/

TenantInfo # Company details, logo

UsageStats # Quota utilization

UsersManagement # Tenant users list

SubscriptionTab # Plan details, billing

SettingsTab # Tenant-specific settings

AuditLogTab # Activity history
```

Funkcjonalności:

- Monitoring wykorzystania storage, users, API calls
- **Quota alerts** warnings przed przekroczeniem
- **Billing integration** plan upgrades, invoicing
- **Security settings** 2FA requirements, IP restrictions

6. USTAWIENIA I ADMINISTRACJA

Priorytet: NISKI - Admin tools

6.1 Zarządzanie Użytkownikami

6.2 Systemowe

```
SystemSettings/

— GeneralSettings # App name, logo, timezone

— EmailSettings # SMTP configuration

— SecuritySettings # Password policies, 2FA

— APISettings # Rate limits, API keys

— BackupSettings # Database backup config
```

© DESIGN SYSTEM

(3) Kolory i Branding

Typography

```
Font Family: Inter (clean, professional)

Sizes:

Display: 2.25rem (36px) - Page titles

Heading: 1.5rem (24px) - Section headers

Body: 1rem (16px) - Main text

Small: 0.875rem (14px) - Labels, captions

Tiny: 0.75rem (12px) - Helper text
```

Layout Grid

REPORT OF THE PROPERTY OF THE

FAZA 1: FUNDAMENT (Tydzień 1)

Dzień 1-2: Setup i Konfiguracja

```
# 1. Instalacja i konfiguracja
npm install @inertiajs/react react react-dom
npm install @tailwindcss/forms @headlessui/react
npm install react-query axios react-hook-form

# 2. Struktura folderów
mkdir resources/js/{Pages,Components,Layouts,Hooks,Types,Utils}

# 3. TypeScript setup
npm install -D typescript @types/react @types/react-dom
```

Rezultat: Środowisko development gotowe

Dzień 3-4: Layout i Nawigacja

```
// 1. AuthenticatedLayout component
// 2. Sidebar navigation
// 3. Top navigation bar
// 4. Mobile responsive menu
// 5. Dark mode toggle (opcjonalnie)
```

Rezultat: Podstawowy layout aplikacji

Dzień 5-7: Dashboard

```
// 1. Dashboard page
// 2. Stats cards components
// 3. Charts integration
// 4. Recent activity feed
// 5. Quick actions panel
```

Rezultat: Funkcjonalny dashboard

FAZA 2: CORE FUNCTIONALITY (Tydzień 2)

Dzień 1-2: Zarządzanie Pracownikami

```
// 1. Workers list page
// 2. Worker profile page
// 3. Search and filters
// 4. Create/Edit worker forms
// 5. Skills management
```

Dzień 3-4: Zarządzanie Zadań

```
// 1. Jobs list (table + kanban)
// 2. Job details page
// 3. Job creation wizard
// 4. Worker assignment interface
// 5. Status updates
```

Dzień 5-7: Formularze

```
// 1. Form builder interface
// 2. Dynamic form renderer
// 3. Form responses viewer
// 4. Field types implementation
// 5. Validation system
```

FAZA 3: ZAAWANSOWANE FUNKCJE (Tydzień 3)

Dzień 1-2: Tenant Management

```
// 1. Tenants overview
// 2. Tenant details page
// 3. Usage monitoring
// 4. Quota visualization
// 5. Settings panels
```

Dzień 3-4: Administracja

```
// 1. User management
// 2. Roles and permissions
// 3. System settings
// 4. Audit logs viewer
// 5. Backup management
```

Dzień 5-7: Polish i Optymalizacja

```
// 1. Performance optimization
// 2. Loading states
// 3. Error boundaries
// 4. Accessibility improvements
// 5. Mobile responsiveness
```

MOBILE-FIRST APPROACH

Mobile Features

- Touch-friendly interfaces minimum 44px touch targets
- Swipe gestures navigate between sections
- Progressive Web App offline capabilities
- Push notifications job updates, alerts
- Camera integration photo capture for forms
- GPS tracking worker location services

Responsive Breakpoints

INTEGRACJA Z BACKEND API

API Client Setup

```
// axios configuration
import axios from 'axios';
const api = axios.create({
 baseURL: '/api',
  headers: {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
    'X-Requested-With': 'XMLHttpRequest',
 },
});
// Request interceptor for auth token
api.interceptors.request.use((config) => {
  const token = document.querySelector('meta[name="csrf-
token"]')?.getAttribute('content');
  if (token) {
    config.headers['X-CSRF-TOKEN'] = token;
  }
  return config;
});
```

© React Query Integration

```
// Przykład hook dla Workers
export const useWorkers = () => {
  return useQuery({
    queryKey: ['workers'],
    queryFn: () => api.get('/workers').then(res => res.data),
    staleTime: 5 * 60 * 1000, // 5 minutes
  });
};
export const useCreateWorker = () => {
  const queryClient = useQueryClient();
```

```
return useMutation({
   mutationFn: (data: CreateWorkerData) =>
      api.post('/workers', data).then(res => res.data),
    onSuccess: () => {
      queryClient.invalidateQueries({ queryKey: ['workers'] });
   },
 });
};
```

METRYKI SUKCESU

© Performance Metrics

- 4 First Contentful Paint < 1.5s
- Mobile PageSpeed Score > 90
- **Desktop PageSpeed Score** > 95
- **Accessibility Score** > 95
- **Bundle Size** < 500KB (gzipped)

Q User Experience

- **Mobile Usage** > 60% traffic
- Session Duration > 5 minutes average
- **© Bounce Rate** < 20%
- **Task Completion Rate** > 85%
- **(3)** User Satisfaction > 4.5/5

Developer Experience

- **Component Test Coverage** > 80%
- 4 Hot Reload < 2s
- **TypeScript Coverage** > 95%
- **Q** ESLint Compliance 100%

Ø DEPLOYMENT I MONITORING

Build Process

```
# Production build
npm run build
# Bundle analysis
npm run analyze
# Type checking
```

```
npm run type-check

# Linting
npm run lint

# Testing
npm run test
```

Monitoring Setup

- Error Tracking Sentry integration
- Analytics Google Analytics / Mixpanel
- 4 Performance Web Vitals monitoring
- **Quantity** User Behavior Hotjar heatmaps
- **Q A/B Testing** Feature flags system

☑ ZAAWANSOWANE FUNKCJE (PRZYSZŁOŚĆ)

AI/ML Integration

- III Predictive Analytics optimal worker assignment
- **@ Smart Recommendations** skill development paths
- **Q** Intelligent Search natural language queries
- 🖺 Chatbot Assistant help and guidance

Real-time Features

- Team Chat WebSocket communication
- P Live Tracking real-time worker locations
- **Q** Push Notifications instant updates
- Live Dashboards real-time metrics

Q External Integrations

- **Email Systems** advanced email workflows
- **Calendar Apps** Google/Outlook sync
- **E Payment Gateways** Stripe/PayPal
- **B** SMS Services Twilio integration
- **Maps Services** Google Maps API

© REKOMENDACJE STARTOWE

Ø DZISIAJ (8 września 2025):

1. Przygotowanie środowiska:

```
cd /d D:\konrad\fieldworker-multi-sector
npm install
npm run dev
```

2. Pierwszy komponent - Dashboard Layout:

```
// resources/js/Layouts/AuthenticatedLayout.tsx
// resources/js/Pages/Dashboard.tsx
// resources/js/Components/StatsCard.tsx
```

3. Podstawowe style Tailwind:

```
/* resources/css/app.css */
@tailwind base;
@tailwind components;
@tailwind utilities;
```

El Checklist pierwszego dnia:

- 🔲 🗹 Konfiguracja TypeScript
- ✓ Setup Tailwind CSS
- Podstawowy Layout component
- Dashboard page skeleton
- Navbar component
- Sidebar navigation
- Pierwszy API call (dashboard stats)

Trinalne podsumowanie

✓ GOTOWOŚĆ DO STARTU:

- Backend: 370 testów przechodzi
- API: 22 kontrolerów gotowych
- Plan Frontend: Kompleksowy i szczegółowy
- **Tech Stack:** Nowoczesny i proven

© OCZEKIWANY REZULTAT:

Nowoczesna, responsywna aplikacja SaaS z intuicyjnym interfejsem, gotowa do użytku przez teams terenowe w różnych sektorach przemysłu.

Ø NASTĘPNY KROK:

Rozpoczęcie implementacji - Layout i Dashboard!

Autor: GitHub Copilot **Data:** 8 września 2025

Status: Ready to implement! 💋