



OrionHRS

Opis systemu i wykorzystanych technologii

Opis Systemu OrionHRS

OrionHRS to system zarządzania zasobami ludzkimi zaprojektowany z myślą o firmach, które chcą usprawnić procesy związane z zarządzaniem pracownikami, obiegiem dokumentów oraz wewnętrzną komunikacją. System został stworzony w oparciu o wzorzec MVC (Model-View-Controller), co pozwala na przejrzystość, modularność i łatwość rozbudowy. Głównie założenie systemu to uproszczenie codziennych procesów HR oraz zwiększenie efektywności zarządzania w organizacji.

Problemy, które rozwiązuje OrionHRS

1. Rejestracja i zarządzanie obecnością

Problem: Manualne rejestrowanie obecności pracowników bywa czasochłonne i podatne na błędy.

Rozwiązanie: System umożliwia pracownikom zgłaszanie obecności w pracy (np. zdalnie, na miejscu, na delegacji), co jest automatycznie rejestrowane i przetwarzane.

2. Zarządzanie urlopami

Problem: Trudności w monitorowaniu wniosków urlopowych, ich zatwierdzania oraz koordynacji między działami.

Rozwiązanie: System oferuje pełny moduł obsługi wniosków urlopowych, umożliwiając składanie, akceptowanie oraz monitorowanie wniosków przez pracowników i przełożonych.

3. Elektroniczny obieg dokumentów

Problem: Tradycyjny, papierowy obieg dokumentów jest nieefektywny i trudny do śledzenia.

Rozwiązanie: System pozwala na przesyłanie, przechowywanie i zatwierdzanie dokumentów w formacie PDF, co eliminuje konieczność używania dokumentów papierowych.

4. Rekrutacja wewnętrzna

Problem: Brak przejrzystości w dostępnych ofertach pracy wewnątrz organizacji.

Rozwiązanie: System zawiera moduł rekrutacji wewnętrznej, który pozwala pracownikom aplikować na stanowiska otwarte w firmie, usprawniając rozwój kariery wewnątrz organizacji.

5. Bezpieczeństwo i kontrola dostępu

Problem: Brak odpowiedniego mechanizmu zarządzania sesją użytkowników.

Rozwiązanie: System wykorzystuje mechanizm sesji użytkownika, co umożliwia wylogowanie się oraz ograniczenie dostępu do funkcji jedynie dla zalogowanych użytkowników.

Funkcjonalności systemu

Zarządzanie pracownikami:

- Przechowywanie danych pracowników, takich jak dane kontaktowe, stanowisko, przełożony, status zatrudnienia.
- Rejestrowanie przełożonych i zastępców.

Rejestracja obecności:

- Zgłaszanie obecności przez pracowników (np. obecność w biurze, praca zdalna, delegacja).
- Automatyczne zatwierdzanie obecności przez system lub przełożonego.

Zarządzanie wnioskami urlopowymi:

- Składanie wniosków urlopowych (wypoczynkowych, zdrowotnych, na żądanie itp.).
- Akceptowanie lub odrzucanie wniosków przez przełożonych.
- Wyświetlanie historii urlopów pracownika.

Elektroniczny obieg dokumentów:

- Przesyłanie dokumentów PDF z poziomu systemu.
- Przechowywanie dokumentów w bazie danych.
- Zatwierdzanie dokumentów przez kilku wyznaczonych zatwierdzających.

Rekrutacja wewnętrzna:

- Wyświetlanie listy otwartych ofert pracy w firmie.
- Szczegóły ofert pracy (opis, wymagania, widełki płacowe).
- Możliwość aplikacji na wewnętrzne stanowiska.

Bezpieczeństwo i zarządzanie sesją:

- Mechanizm logowania i wylogowania.
- Ograniczenie dostępu do funkcji wyłącznie dla zalogowanych użytkowników.

Estetyczny i przyjazny interfejs:

- Centralnie umieszczone formularze (np. logowania) w estetycznych, zaokrąglonych kartach z efektami hover i cieniami.
- Responsywny design oparty na Bootstrap, dostosowany do różnych urządzeń.

Użyte technologie

Backend:

- C# (ASP.NET Core) - platforma do tworzenia aplikacji webowych oparta na wzorcu MVC.
- Entity Framework Core - ORM używany do komunikacji z bazą danych PostgreSQL.
- PostgreSQL - relacyjna baza danych, przechowująca informacje o pracownikach, obecności, wnioskach urlopowych i dokumentach.

Frontend:

- Razor Pages - do generowania widoków serwera.
- Bootstrap - framework CSS używany do responsywnego projektowania.

Inne:

- Session Management - zarządzanie sesją użytkowników w ASP.NET Core.
- HTML5 & CSS3 - do tworzenia i stylizowania elementów interfejsu użytkownika.
- JavaScript/jQuery - do obsługi dynamicznych interakcji użytkownika.

Użyte wzorce projektowe

Model-View-Controller (MVC):

Podział aplikacji na trzy warstwy: Model (logika danych), View (prezentacja) i Controller (logika aplikacji). Dzięki temu kod jest przejrzysty i łatwy do utrzymania.