

Sprawozdanie z projektu

Adam Kasprzyk, Dominik Prabucki, Kacper Walenga

28 maja 2024

1 Opis funkcjonalny systemu

1. Zarządzanie pracownikami

Umożliwia przeglądanie listy pracowników, sortowanie i filtrowanie danych, a także wyszukiwać konkretne osoby według różnych kryteriów.

2. Edycja danych

Aplikacja pozwala na edycję danych pracowników. A

3. Dodawanie danych

System umożliwia dodawanie nowych pracowników do bazy danych.

4. Możliwość utworzenia nowego konta

Aplikacja oferuje możliwość utworzenia nowych kont użytkowników. Nowi użytkownicy mogą zarejestrować się, podając wymagane informacje, adres e-mail, hasło.

5. Możliwość zalogowania się i wylogowania

Aplikacja zapewnia bezpieczne mechanizmy logowania i wylogowania. Użytkownicy mogą logować się, podając swoje dane uwierzytelniające (adres e-mail i hasło).

6. Aplikacja identyfikuje zalogowanego użytkownika

Po zalogowaniu się, aplikacja identyfikuje zalogowanego użytkownika i dostosowuje interfejs oraz dostępne funkcje.

2 Opis technologiczny

W projekcie wykorzystano następujące technologie:

PHP w wersji 8.2

PHP to popularny język skryptowy używany głównie do tworzenia dynamicznych stron internetowych i aplikacji webowych. Wersja 8.2 wprowadza nowe funkcje i optymalizacje, które zwiększają wydajność oraz bezpieczeństwo aplikacji.

Framework Laravel w wersji 11

Laravel to popularny framework PHP, który umożliwia tworzenie aplikacji webowych w sposób szybki i efektywny.

- Architektura MVC: Model-View-Controller
- Kompatybilność z nowoczesnymi narzędziami frontendowymi

Baza danych Postgres 8.0

PostgreSQL to zaawansowany, obiektowo-relacyjny system zarządzania bazami danych.

Redis 7.2.3-alpine

Redis to szybki, otwartoźródłowy system magazynowania danych w pamięci, który jest używany jako baza danych, cache oraz broker komunikatów.

Tailwind CSS

Tailwind CSS to nowoczesny framework CSS, który umożliwia szybkie tworzenie spójnych i atrakcyjnych interfejsów użytkownika. Tailwind dostarcza narzędzia do tworzenia niestandardowych projektów bez opuszczania plików HTML.

3 Wyszczególnione wdrożone zagadnienia kwalifikacyjne

- 1. Framework MVC**
W projekcie wykorzystano framework MVC (Model-View-Controller) na backendzie (laravel).
- 2. Framework CSS**
Do stylowania aplikacji użyto frameworka CSS, który zapewnia spójny wygląd i ułatwia tworzenie responsywnych interfejsów użytkownika (Tailwind CSS).
- 3. Baza Danych**
W projekcie zaimplementowano bazę danych, która umożliwia przechowywanie i zarządzanie danymi pracowników oraz innymi istotnymi informacjami (Postgres).
- 4. Dependency Manager**
Zastosowano system zarządzania zależnościami, który automatyzuje instalację i aktualizację bibliotek oraz narzędzi potrzebnych do działania aplikacji.
- 5. HTML**
HTML został użyty do stworzenia szkieletu aplikacji internetowej.
- 6. CSS**
CSS służył do stylowania aplikacji internetowej.
- 7. JavaScript**
JavaScript został użyty do uinteraktywnienia aplikacji internetowej.
- 8. Routing**
W projekcie zaimplementowano routing. Używany jest w routes.web, routes.auth w celu dostępu do podstron jako pretty url.
- 9. ORM**
Wykorzystano mapowanie obiektowo-relacyjne (ORM) do zarządzania danymi w bazie danych za pomocą obiektów w kodzie aplikacji.
- 10. Uwierzytelnianie**
Używane w celu zapewnienia systemu logowania oraz dostępu do aplikacji jedynie dla zalogowanych użytkowników. Obsługiwane w AuthController.
- 11. Lokalizacja**
Aplikacja umożliwia przełączanie języka interfejsu. Używana jest na każdej wyświetlanej stronie w celu zapewnienia obsługi wielu języków.
- 12. Formularze**
Formularze zostały wykorzystane do przesyłania danych do aplikacji, umożliwiając interakcję użytkownika z systemem. Używane w login, register, employees.edit, leaves, transactions w celu przesyłania wpisywanych danych do serwera.
- 13. Konsumpcja API**
Aplikacja integruje się z zewnętrznymi API, co pozwala na wykorzystanie dodatkowych funkcjonalności i danych.
- 14. RWD (Responsive Web Design)**
Frontend aplikacji jest responsywny, co zapewnia optymalne wyświetlanie na różnych urządzeniach, od komputerów po smartfony.
- 15. Logger**
Używany w EmployeesController w celu zapisywania informacji kto edytował dane innego pracownika, w AuthController w celu zapisywania który użytkownik loguje się oraz wylogowuje się.
- 16. Deployment**
Aplikacja została wdrożona na serwerze produkcyjnym, co umożliwia jej użytkowanie przez końcowych użytkowników.

4 Instrukcję lokalnego i zdalnego uruchomienia systemu

4.1 Lokalne uruchamianie systemu

1. Uruchom PowerShell'a w trybie administratora.

2. Włącz WSL:

```
1  dism.exe /online /enable-feature  
2  /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

3. Włącz Virtual Machine Platform:

```
1  dism.exe /online /enable-feature  
2  /featurename:VirtualMachinePlatform /all /norestart
```

4. Ustaw WSL na wersję 2:

```
1  wsl --set-default-version 2
```

5. Zainstaluj WSL z GNU/Linux Ubuntu:

```
1  wsl.exe --install
```

6. Stwórz nowego użytkownika (podaj nazwę i hasło).

7. Zaktualizuj GNU/Linux Ubuntu:

```
1  sudo apt update ; sudo apt upgrade
```

8. Zainstaluj Dockera:

```
1  sudo chmod +x ./install-docker  
2  ./install-docker
```

9. Sprawdź, czy wszystko działa:

```
1  docker run hello-world
```

10. Sklonuj repozytorium:

```
1  git clone https://github.com/PrabuckiDominik/2024-ppi.git
```

11. Przejdź do folderu z projektem:

```
1  cd 2024-ppi
```

12. Zminimalizuj terminal i otwórz Visual Studio Code.

13. Dodaj rozszerzenia WSL i Docker, autorstwa Microsoft.

14. Zamknij Visual Studio Code, wróć do terminala z GNU/Linuxem.

15. Uruchom projekt w Visual Studio Code:

```
1  code .
```

16. Zamknij terminal.

17. Uruchom terminal w Visual Studio Code (Ctrl + `).

18. Zbuduj obrazy Dockera:

```
1  docker compose build
```

19. Stwórz plik .env i dostosuj jego konfigurację:

```
1  cp .env.example .env
```

20. Uruchom Dockera:

```
1  docker compose up -d
```

21. Po zakończeniu procesu uruchamiania projekt będzie dostępny pod adresem <http://localhost>.

5 Wnioski projektowe

1. **Zrozumienie Wagi Frameworków:** Praca nad projektem umożliwiła nam głębsze zrozumienie roli i znaczenia frameworków w naszej dziedzinie. Zauważyliśmy, że wybór odpowiedniego frameworka może mieć kluczowe znaczenie dla efektywności i skuteczności naszego projektu.
2. **Nabywanie Umiejętności Korzystania z Frameworków:** Przez aktywne zaangażowanie się w projekt, zdobyliśmy praktyczne umiejętności w zakresie korzystania z różnych frameworków. Poprzez eksperymentowanie, testowanie i dostosowywanie ich do naszych potrzeb, staliśmy się bardziej pewni w ich użyciu i zyskaliśmy umiejętność szybkiego przystosowywania się do nowych narzędzi.
3. **Doskonalenie Pracy Zespołowej:** Praca w trzyosobowym zespole pozwoliła nam na doskonalenie naszych umiejętności współpracy i komunikacji. Dzięki regularnym spotkaniom, dzieleniu się pomysłami i rozwiązywaniu problemów jako grupa, zauważyliśmy wzrost efektywności naszej pracy. Wspólnie opracowane strategie i podział obowiązków pozwoliły nam efektywnie realizować cele projektowe.