

Sprawozdanie z projektu

Adam Kasprzyk, Dominik Prabucki, Kacper Walenga

4 czerwca 2024

1 Opis funkcjonalny systemu

1. Zarządzanie pracownikami

Umożliwia przeglądanie listy pracowników oraz niezweryfikowanych użytkowników.

2. Edycja danych

Aplikacja pozwala na edycję danych pracowników oraz na weryfikację kont niezweryfikowanych użytkowników.

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, nazwę użytkownika oraz hasło.

5. Możliwość zalogowania się i wylogowania

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

6. Aplikacja identyfikuje zalogowanego użytkownika

Po zalogowaniu się, aplikacja identyfikuje zalogowanego użytkownika i dostępne dla niego funkcje.

7. Mailing

Użytkownik otrzymuje emaila gdy zakłada konto, zostaje zweryfikowany oraz gdy resetowane jest jego hasło.

8. Resetowanie hasła

Gdy użytkownik zapomniał hasła, ma możliwość zresetowania go. Na jego skrzynkę pocztową zostaje wysłany mail z linkem, umożliwiającym zmianę hasła.

9. Statystyki

Pracownik ma możliwość wyświetlenia raportu o statystykach kupna, sprzedaży i ilości posiadanego przez firmę materiału. Wyświetla również ilość pracowników.

10. Transakcje

Pracownik ma możliwość zakupu oraz sprzedaży wybranego materiału.

11. Lokalizacja

Użytkownik ma możliwość zmiany języku na wersję polską bądź angielską. System jest przystosowany dla osób z dysfunkcjami.

2 Opis technologiczny

W projekcie wykorzystano następujące technologie:

PHP w wersji 8.2

Framework Laravel w wersji 11

Baza danych Postgres 16.1-alpine3.19

Redis 7.2.3-alpine

Node 20-alpine

Tailwind CSS

Nginx 1.25.3-alpine

Docker

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. Cache**
Zapewniany przez framework redis, używany do zapisywania danych pobieranych z API i aktualizowana co 2 godziny.
- 5. Dependency Manager**
Zastosowano system zarządzania zależnościami, który automatyzuje instalację i aktualizację bibliotek oraz narzędzi potrzebnych do działania aplikacji. Możemy znaleźć je w package.json
- 6. HTML**
HTML został użyty do stworzenia szkieletu aplikacji internetowej.
- 7. CSS**
CSS został użyty do stylowania aplikacji internetowej.
- 8. JavaScript**
JavaScript został użyty do uinteraktywnienia aplikacji internetowej.
- 9. Routing**
W projekcie zaimplementowano routing. Używany jest w routes.web, routes.auth w celu dostępu do podstron jako pretty url.
- 10. ORM**
Wykorzystano mapowanie obiektowo-relacyjne (ORM) do zarządzania danymi w bazie danych za pomocą obiektów w kodzie aplikacji.
- 11. Uwierzytelnianie**
Używane w celu zapewnienia systemu logowania oraz dostępu do aplikacji jedynie dla zalogowanych użytkowników. Obsługiwane w AuthController.
- 12. Lokalizacja**
Aplikacja umożliwia przełączanie języka interfejsu. Używana jest na każdej wyświetlanej stronie w celu zapewnienia obsługi wielu języków.
- 13. Mailing**
Aplikacja wysyła mail po utworzeniu konta, przy resetowaniu hasła oraz po zweryfikowaniu konta.
- 14. 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.
- 15. Asynchroniczne Interakcje**
Podczas pobytu na stronie statystyki co 10 minut dane są automatycznie aktualizowane bez odświeżania przez użytkownika.

16. Konsumpcja API

Aplikacja integruje się z zewnętrznymi API, co pozwala na wykorzystanie dodatkowych funkcjonalności i danych.

17. Publikacja API

Używany w controllers/StatisticsController w celu opublikowania danych o statystykach przetwarzania materiału.

18. RWD (Responsive Web Design)

Frontend aplikacji jest responsywny, co zapewnia optymalne wyświetlanie na różnych urządzeniach, od komputerów po smartfony.

19. 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.

20. Deployment

Przeniesiono aplikację z fazy rozwoju do środowiska produkcyjnego, gdzie aplikacja jest dostępna dla użytkowników końcowych.

4 Instrukcja lokalnego uruchomienia projektu w środowisku developerskim

4.1 Lokalne uruchamianie systemu

1. Zainstaluj Dockera na swoim systemie.

2. Sklonuj repozytorium:

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

3. Przejdź do folderu z projektem:

```
1 cd 2024-ppsi
```

4. Stwórz plik .env i dostosuj jego konfigurację:

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

5. Zbuduj obrazy Dockera:

```
1 make init
```

6. Uruchom projekt:

```
1 make dev
```

7. 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.