

PPSI2_Wedding_Planner

Jakub Engielski (Leader/PM/Frontend),
Jan Kwiatkowski (Frontend/Backend),
Szymon Michno (Tester),
Michał Myśków (DevOps/Backend),
Anna Sokołowska (Frontend/Backend)

Styczeń 2022

1 Opis funkcjonalny systemu

Celem Projektu "PPSI2_Wedding_Planner" było utworzenie aplikacji do planowania ślubów, która będzie miała w sobie takie elementy jak rejestr gości z potwierdzeniem przybycia (RSVP) oraz możliwością ręcznej modyfikacji tej funkcji przez państwo młodych, planer miejsc siedzących oraz umożliwi stworzenie własnej listy rzeczy do wykonania.

Państwo młodzi otrzymają także unikalny kod, wysyłany wraz z zaproszeniem, który pozwoli gościom na podgląd jak wyglądają miejsca siedzące.

Po wejściu na naszą stronę użytkownik przywitany zostanie menu logowania oraz rejestracji. Oferujemy także możliwość zalogowania się przy pomocy Facebooka, co wymagało odrobiny zabawy aby dostać dostęp do developerskiego API. Istnieje także możliwość przypomnienia hasła w wypadku gdy zostało ono zapomniane.

Po zalogowaniu się użytkownicy otrzymają możliwość utworzenia swojego wesela. Podając dane pana i panny młodych, planowaną datę oraz godzinę oraz wybrać jedną z trzech dostępnych sal. Po utworzeniu wesela istnieje także możliwość edycji go. W wypadku gdy użytkownik posiada już utworzone wesele może do niego dodać gości, przydzielić im miejsca ręcznie z opcją wyświetlania konfliktów. A w razie błędów lub chęci zmiany któregokolwiek usadzenia mieć taką możliwość. Na stronie głównej pojawi się też informacja ile gości zostało zaproszonych, ile przyjęło już zaproszenia oraz ile dni pozostało do wesela.

Zaimplementowany jest także system checklisty, która jest w pełni tworzona przez użytkownika. Może on wpisać tam dowolne zadania, edytować je a także zaznaczać czy zostały one już ukończone. Umożliwiamy także dostęp do kalkulatora kosztów wesela.

Poza tym mamy system RSVP, który pozwala na wysłanie maila potwierdzającego obecność dla gościa. Jeśli jednak dany gość jest osobą mniej zaznajomioną z komputerami pozwalamy także na wydrukowanie gotowych tekstów zaproszeń aby móc wysłać je metodą tradycyjną.

2 Streszczenie opisu technologicznego

Korzystamy z framework'u Symfony 5.3.9 wspierającego język PHP, dzięki któremu jesteśmy w stanie utworzyć nasz projekt w jakimkolwiek logicznym odstępie czasu. Wybraliśmy Symfony zamiast na przykład Laravela, gdyż chcieliśmy spróbować coś czego nie używaliśmy wcześniej przez co poszerzyć nasze możliwości.

Do bazy danych użyliśmy systemu zarządzania relacyjną bazą danych MariaDB. Ma on znacznie większą liczbę nowych funkcji w odróżnieniu od MySQL'a co czyni ją lepszą pod względem wydajności.

Docker umożliwia nam łatwe pakowanie, dostarczanie oraz uruchamianie aplikacji jako lekkie, przenośne i samowystarczalne kontenery, które mogą działać praktycznie wszędzie.

Webpack Encore, część framework'u Symfony, posłużył nam do szybkiego i wygodnego utworzenia layout'u, a także zapewnienia, że nasza strona będzie miała w pełni działające RWD. Co jest niezbędne w świecie gdzie smartfony stają się coraz bardziej powszechne.

PHPStorm jest lekkim i inteligentnym środowiskiem programistycznym (IDE), z którego korzystaliśmy podczas pisania kodu naszej webowej aplikacji. Szybki, intuicyjny i wygodny edytor kodu pozwoliły nam na szybkie debugowanie.

Github, czyli system kontroli wersji, pozwalający wielu użytkownikom pracować na różnych komputerach bez zaburzania przepływu pracy. Oczywiście to na nim utworzyliśmy nasze repozytorium.

Trello - tablica pozwalająca na wygodną organizację pracy, poprzez przypisywanie zadań do konkretnych osób. Możemy też nadać konkretnym zadaniom wagę czyniąc to bardzo czytelnym backlogiem zadań

Discord - komunikator głosowo-tekstowy pozwalający na szybką i wygodną komunikację pośród zespołu.

3 Instrukcję lokalnego i zdalnego uruchomienia systemu

3.1 Postawienie systemu lokalnie

Wymagane oprogramowanie:

PHPStorm lub dowolny inny IDE

Docker Desktop

Terminal Windows (nie jest potrzebny osobno jeśli jest wbudowany w IDE jak w PHPStorm)

Github Desktop – aby móc wprowadzać zmiany, bądź pobierać aktualizacje jeśli są potrzebne. (nie jest wymagany jeśli jest wbudowany w IDE, bądź jeśli ktoś posiada zainstalowany pakiet GIT do użytku poprzez terminal)

Jak postawić środowisko testowe na dockerze?

[Instrukcja dla Windowsa](#)

Instrukcja dla Linuxa

Jeżeli wszystko zainstalowałeś, przejdź do instrukcji poniżej:

1. Pobranie projektu z repozytorium oraz przejście do jego folderu:

```
git clone https://github.com/MichalMyskow/PPSI2-Wedding-Planner.git
cd PPSI2-Wedding-Planner
```

2. Konfiguracja pliku .env

```
cp .env .env.local
```

3. Uruchomienie kontenerów dockerowych:

```
docker-compose up -d --build
```

4. Pobranie zależności Composer:

```
docker-compose exec php composer install
```

5. Pobranie zależności npm:

```
docker-compose exec php npm install
```

6. Budowanie Assetów:

```
docker-compose exec php npm run dev
```

Migracje:

1. Utworzenie migracji:

```
php bin/console doctrine:migrations:diff
php bin/console doctrine:migrations:migrate
```

Bez migracji:

1. Utworzenie schematu bazy danych:

```
php bin/console doctrine:schema:update --force
```

2. Załadowanie zdefiniowanych wcześniej danych do bazy danych:

```
php bin/console doctrine:fixtures:load
```

Usługi:

- www <http://localhost:8080>
- phpmyadmin <http://localhost:8081>

Dane logowania do phpMyAdmin:

- Serwer: database
- Użytkownik: user
- Hasło: 123qwe
- Nazwa bazy danych: weddingplannerdb

Przydatne Komendy:

1. Docker - uruchomienie kontenerów:

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

2. Docker - zatrzymanie kontenerów:

```
docker-compose stop
```

3. Docker - zatrzymanie i usunięcie kontenerów

```
docker-compose down
```

3.2 Postawienie systemu zdalnie

W naszym wypadku użyliśmy programu PuTTY, aby podłączyć się z serwerem Debianowym. Może do tego posłużyć dowolny klient SSH.

Po tym musieliśmy zainstalować następujące pakiety:

- nginx 1.18
- PHP 8.0.14
- mariadb 10.5.12
- phpMyAdmin 5.0.4
- Node.js 16.13.2

Pliki naszej strony są automatycznie pobierane z repozytorium git.

1. Pobieranie zależności Composer'a:

```
composer install
```

2. Pobranie zależności npm:

```
npm install
```

3. Budowanie Assetów:

```
yarn build
```

4. Utworzenie schematu bazy danych:

```
php bin/console doctrine:schema:update --force
```

5. Załadowanie zdefiniowanych wcześniej danych do bazy danych:

```
php bin/console doctrine:fixtures:load
```

4 Testy

4.1 Instrukcja uruchomienia testów

1. Uruchomienie testów:

```
php ./vendor/bin/phpunit
```

4.2 Opis testowanych funkcjonalności

Do przeprowadzenia naszych testów używamy paczek, "phpunit", "phpunit symfony" oraz "test-pack".

Głównym celem testów jest sprawdzenie czy na każdej podstronie w pełni działa autoryzacja. Jeśli mamy dbać o dane klientów, jest to podstawowa funkcjonalność.

Poza tym sprawdzany jest cały system logowania, czy nie pojawiają się w nim błędy.

Sprawdza się co się dzieje po zalogowaniu. Sprawdza to też czy stworzone jest wesele. W wypadku gdy wesele nie jest utworzone użytkownik jest przenoszony do ścieżki /create-wedding.

5 Dokumentacja

Link do pełnej dokumentacji na naszym repozytorium: https://github.com/MichalMyskow/PPSI2_Wedding_Planner/tree/develop/documentation

6 Wnioski projektowe

Projekt ten nie był aż tak prostym przedsięwzięciem jak zakładaliśmy przy wybraniu tematu. Jednak ucząc się po drodze udało się nam go ukończyć. Nigdy wcześniej nie zajmowaliśmy się niczym związanym z tematyką weselną. Musieliśmy więc przeprowadzić trochę researchu, aby poprawnie utworzyć naszą aplikację.

Poza techniczną stroną projektu musieliśmy się też zmierzyć po raz pierwszy w ramach uczelni z podziałem zadań na sprinty oraz estymatą.

Niektórzy członkowie naszego zespołu mieli już z tym styczność w życiu zawodowym, natomiast niektórzy musieli zainteresować się tym tematem po raz pierwszy.

Nie mieliśmy też osoby, która personalnie byłaby zainteresowana frontendem. Musiała więc to być praca wspólna, aby osiągnąć nasz cel końcowy.

Możemy więc wynieść następujące wnioski:

- Przygotowanie aplikacji wymaga wiedzy poza czystą wiedzą techniczną, jak w naszym wypadku wiedzę o weselach
- Warto posiadać członków zespołu, którzy posiadają głęboką wiedzę w danym zakresie, np. Dev-ops
- Wykupienie domeny jest bardzo proste
- Organizacja grupy może być wymagająca, gdy niektórzy członkowie są osobami pracującymi

- Flow pracy może łatwo być przerywany w wypadku chorób i różnych prywatnych spraw w zespole
- Nie należy być zbyt optymistycznym w przygotowywaniu estymaty