

Sprawozdanie z przedmiotu
Projektowanie i programowanie systemów
internetowych II

Adrian Matys
Kacper Walenga
Vladyslav Kutsyn
Denys Lutsenko

Semestr zimowy 2024/2025

Spis treści

1	Cel projektu	3
2	Opis funkcjonalny systemu	3
2.1	System	3
2.2	Administrator	3
2.3	Pracownik	3
2.4	Klient	4
3	Opis technologiczny	4
4	Podział obowiązków w zespole	4
5	Instrukcja lokalnego i zdalnego uruchomienia systemu	5
6	Wnioski projektowe	5

1 Cel projektu

Celem wykonywanego przez nas projektu, było stworzenie aplikacji ułatwiającej zamawianie pizzy w restauracji. Aplikacja ma działać na tablecie w restauracji, wspomagając pracę kelnera. W aplikacji użytkownik może dodać do koszyka pizzę już dodaną do menu lub stworzyć własną z podanych składników. Pracownicy mogą zarządzać zamówieniami. Administrator może zarządzać zamówieniami i pracownikami, ma też wgląd do logów aplikacji oraz zarządza danymi restauracji.

2 Opis funkcjonalny systemu

2.1 System

- Pozwala zamówić pizzę dostępną w menu
- Umożliwia logowanie nowych urządzeń jako użytkowników do systemu przez administratora.
- Umożliwia zmianę i resetowanie hasła urządzenia w przypadku jego utraty przez administratora.
- Ma części przeznaczoną dla klienta i dla zalogowanych pracownika i administratora
- Wysyła o brakach magazynowych zbiorczo raz dziennie
- Wszystkie akcje powinny być logowane w systemie i widoczne dla administratora
- Zapisuje akcje wykonane w formacie: nazwa pracownika, kategoria, typ, data oraz detale np: admin@collegiumpizzerona.online Auth Logged in 2025-01-18 22:39:45 Przycisk(Show details)

2.2 Administrator

- Może zarządzać danymi pizzerii (nazwa, ulica, miasto, numer telefonu, czy dostępna dostawa, ile maksymalnie kilometrów do dostawy)
- Może zarządzać pracownikami oraz ich danymi
- Widzi dzienne, tygodniowe i miesięczne zużycie danych produktów oraz liczbe zamówień konkretnych pozycji z menu
- Widzi logi z aplikacji

2.3 Pracownik

- Może zarządzać stanem magazynowym np. może wpisać, że na stanie posiada 10kg sera mozzarella
- Może układać menu złożone z różnych pizz składających się z uprzednio zdefiniowanych składników np. "pizza inferno" może składać się z sosu pomidorowego, sera mozzarelli, sera gorgonzoli, salami picante, nduji i cebuli
- Otrzymuje powiadomienia, gdy stan danego produktu zbliża się do zera

2.4 Klient

- Może zamówić dowolną liczbę pizz.
- Jest możliwość wyboru dodatków do każdej pizzy, którą zamawia.
- Widzi tylko te pizze, które są aktualnie możliwe do zrobienia przez kucharza ze względu na stany magazynowe.
- Przychodząc do pizzerii może na stronie internetowej obejrzeć dostępne menu.
- Może sprawdzić koszt zamówienia.

3 Opis technologiczny

- **Backend:** Projekt został zbudowany w języku **PHP** w oparciu o framework **Laravel** w wersji 11, który oferuje zaawansowane narzędzia do tworzenia aplikacji webowych, takie jak routing, kontrolery oraz migracje.
- **Frontend:** Interfejs użytkownika został opracowany przy użyciu **Blade** oraz **HTML**, **CSS** i **JavaScript**. W projekcie wykorzystano również **Bootstrap**.
- **Baza danych:** Projekt korzysta z systemu zarządzania bazą danych **PostgreSQL** w wersji 16.1.
- **Cache:** W celu zwiększenia wydajności aplikacji używany jest system cache oparty na **Redis**, który wspiera przechowywanie sesji, wyników zapytań oraz innych dynamicznych danych.
- **Środowisko:** Projekt rozwijany i testowany jest w środowisku **Docker**, co pozwala na łatwą konfigurację i uruchamianie aplikacji w kontenerach. Do wdrożenia aplikacji użyto serwera **Nginx**.

4 Podział obowiązków w zespole

Obowiązki

- **Adrian Matys:** Frontend, Lider.
- **Kacper Walenga:** Backend, devops.
- **Vladyslav Kutsyn:** Frontend.
- **Denys Lutsenko:** Testy.

5 Instrukcja lokalnego i zdalnego uruchomienia systemu

1. Zainstaluj Dockera na swoim systemie.
2. Sklonuj repozytorium: `git clone https://github.com/AdrianMatys/ProjektPizzeria.git`.
3. Przejdź do folderu z projektem: `cd ProjektPizzeria`.
4. Stwórz plik `.env` i dostosuj jego konfigurację: `cp .env.example .env`.
5. Zbuduj obrazy Dockera: `make init`
6. Uruchom projekt: `make dev`

Projekt będzie dostępny pod adresem: <http://localhost/>.

6 Wnioski projektowe

Adrian Matys

Podczas pracy nad projektem, w którym pełniłem rolę lidera zespołu oraz frontend developera, miałem okazję rozwijać swoje umiejętności w zakresie pracy zespołowej, organizacji czasu oraz efektywnego zarządzania zadaniami. Szczególną uwagę poświęciłem poprawie komunikacji wewnątrz zespołu, co obejmowało nie tylko lepsze przekazywanie informacji, ale również dogłębne zrozumienie zadań realizowanych przez innych członków zespołu. Te doświadczenia pozwoliły mi spojrzeć na projekt z szerszej perspektywy. W trakcie realizacji projektu nauczyłem się korzystania z technologii takich jak PHP oraz Blade. Poznałem również techniki tworzenia i implementacji wielokrotnie wykorzystywanych komponentów, takich jak nagłówki stron (**header**), co znacznie zwiększyło wydajność pracy i modularność kodu. Początkowo miałem trudności z wdrożeniem narzędzi takich jak **GitHub**, które były dla mnie nowością. Dodatkowo, problemy z instalacją **WSL 2.0** uniemożliwiły mi korzystanie z **PhpStorm**, co było pewnym wyzwaniem. Na szczęście, dzięki wsparciu jednego z członków zespołu, Kacpra, opanowałem efektywne korzystanie z **GitHub Desktop**, co pozwoliło mi lepiej zarządzać wersjonowaniem kodu. Gdybym mógł rozpocząć projekt ponownie, zdecydowanie włączyłbym **GitHub** od samego początku jako integralną część procesu pracy. Dodatkowo, w projekcie korzystaliśmy z narzędzi wspomagających organizację i zarządzanie zadaniami, takich jak **Figma** oraz **Mural**. **Figma** odegrała kluczową rolę w projektowaniu interfejsów użytkownika, umożliwiając łatwe dzielenie się projektami graficznymi z Vladyslavem, natomiast **Mural** pomógł w skutecznym rozdzielaniu zadań i monitorowaniu postępów w realizacji pracy. Te narzędzia znacząco przyczyniły się do uporządkowania procesu pracy oraz poprawy efektywności współpracy zespołowej.

Vladyslav Kutsyn

W projekcie pełniłem rolę frontend developera, współpracując z testerem, backend developerem oraz liderem frontendowym. Moim głównym zadaniem było tworzenie i implementacja elementów interfejsu użytkownika oraz integracja frontendu z backendem. Dzięki pracy nad tym projektem nauczyłem się wielu nowych rzeczy, które znacząco wpłynęły na mój rozwój. Przede wszystkim opanowałem korzystanie z **GitHuba** – od podstawowych funkcji, takich jak klonowanie repozytoriów i tworzenie commitów, po bardziej zaawansowane działania, takie jak rozwiązywanie konfliktów i praca z pull requestami. Rozwijałem również swoje umiejętności w **JavaScript**, ucząc się pisać bardziej wydajny i zoptymalizowany kod. Lider frontendowy był dla mnie dużym wsparciem w rozwiązywaniu bardziej złożonych problemów technicznych, co pomogło mi lepiej zrozumieć strukturę i organizację pracy frontendowej. Podczas projektu największym wyzwaniem było połączenie frontendu z backendem. Napotkaliśmy problemy techniczne związane z integracją danych, które wymagały dodatkowej analizy i wspólnego rozwiązywania z backend developerem. Dzięki wspólnej pracy udało się jednak przezwyciężyć te trudności, co pozwoliło na pomyślną realizację zadania. Projekt przebiegał sprawnie, głównie dzięki jasnemu podziałowi ról w zespole i regularnej wymianie informacji. Projekt był dla mnie cennym doświadczeniem. Planuję kontynuować naukę, szczególnie w zakresie frameworków takich jak **React** czy **Vue.js**, aby jeszcze bardziej rozwijać swoje umiejętności frontendowe. Chciałbym również zgłębić temat automatyzacji testów i procesów CI/CD, co mogłoby zwiększyć efektywność mojej pracy w przyszłych projektach. Podsumowując, udział w projekcie pozwolił mi nie tylko poszerzyć wiedzę techniczną, ale także lepiej zrozumieć, jak efektywnie współpracować w zespole. Zdobyte doświadczenie na pewno będzie dla mnie solidnym fundamentem w dalszej karierze zawodowej.

Kacper Walenga

Podczas realizacji tego projektu miałem okazję znacząco rozwijać swoje umiejętności. Praca w **Laravelu** pozwoliła mi lepiej zrozumieć jego możliwości. Dzięki temu stworzyłem funkcjonalny panel administracyjny umożliwiający zarządzanie zamówieniami, menu pizzerii oraz użytkownikami. W trakcie projektu regularnie korzystałem z **GitHuba**, co umożliwiło mi efektywne zarządzanie kodem. Praca z systemem kontroli wersji, w tym tworzenie branchy, pull requestów oraz rozwiązywanie konfliktów, dała mi praktyczne doświadczenie, które jest kluczowe w pracy zespołowej w dużych projektach. Do pisania kodu używałem **PHPStorma**, co znacznie usprawniło proces programowania. Narzędzie to, dzięki funkcjom takim jak analiza błędów w czasie rzeczywistym oraz wsparcie w uzupełnianiu kodu, znacząco ułatwiło pracę. Dodatkowo integracja **PHPStorma** z **GitHubem** ułatwiła synchronizację kodu oraz zarządzanie repozytorium projektu. Ważnym etapem pracy było wdrożenie aplikacji na serwer. Byłem odpowiedzialny za skonfigurowanie środowiska produkcyjnego i uruchomienie aplikacji na serwerze. Proces ten wymagał ode mnie konfiguracji serwera, plików **.env** oraz migracji bazy danych. W trakcie pracy napotkałem kilka wyzwań, takich jak zarządzanie zamówieniami oraz zabezpieczenie aplikacji przed nieoczekiwanymi działaniami użytkowników. Na szczęście udało mi się rozwiązać te problemy. Projekt pozwolił mi zidentyfikować obszary, w których aplikacja mogłaby być rozwinięta. Przykładowo, wdrożenie systemu powiadomień w czasie rzeczywistym, które byłyby wysyłane do klienta w momencie zmiany statusu zamówienia, lub wiadomości do pracownika w momencie, gdy kończą się składniki w magazynie.

Realizacja tego projektu nie tylko wzbogaciła moje doświadczenie techniczne w zakresie `Laravela`, `GitHuba` oraz `PHPStorma`, ale również pozwoliła mi na rozwinięcie umiejętności zarządzania pełnym cyklem życia aplikacji – od planowania, przez implementację, aż po wdrożenie. Zdobyta wiedza będzie cenna podstawa do realizacji kolejnych projektów w przyszłości.

Denys Lutsenko

Podczas realizacji projektu miałem okazję poszerzyć swoje umiejętności w zakresie testowania. Przeprowadzanie testów aplikacji było jednym z kluczowych elementów, który pozwolił mi na dokładną weryfikację jej funkcjonalności i stabilności. Testowanie wymagało dużego zaangażowania, ale dzięki temu zrozumiałem, jak istotne jest odpowiednie przygotowanie testów oraz ich systematyczne wykonywanie w trakcie rozwoju projektu. Samodzielne tworzenie testów pozwoliło mi lepiej poznać proces testowania aplikacji i nauczyć się, jak wykrywać oraz naprawiać błędy na różnych etapach realizacji. Choć było to czasochłonne, okazało się kluczowe dla zapewnienia jakości produktu końcowego. Przeprowadzenie testów jednostkowych i funkcjonalnych dało mi solidne podstawy do dalszego rozwoju w tym obszarze. Dzięki testom integracyjnym mogłem upewnić się, że poszczególne części aplikacji współpracują ze sobą zgodnie z oczekiwaniami, a testy wydajnościowe pozwoliły na ocenę reakcji systemu pod większym obciążeniem. W przyszłości będę starał się bardziej systematycznie wdrażać testowanie od samego początku projektu, aby zminimalizować ryzyko błędów i poprawić ogólną jakość aplikacji. Wnioski, które wyciągnąłem, to przede wszystkim potrzeba lepszego planowania testów na początku projektu oraz ich systematycznego wykonywania. Testowanie stało się dla mnie równie ważnym elementem jak samo kodowanie, a jego rola w procesie tworzenia aplikacji stała się dla mnie znacznie bardziej doceniana.