  Dokumentacja projektu

MedicaBARP

Spis treści

[1. Opis projektu 2](#_Toc155992324)

[2. Autorzy 2](#_Toc155992325)

[3. Założenia projektowe 3](#_Toc155992326)

[4. Ograniczenia i ryzyka projektowe 3](#_Toc155992327)

[5. Opis funkcjonalności 3](#_Toc155992328)

[6. Diagram związków encji 6](#_Toc155992329)

[7. Diagram przypadków użycia 7](#_Toc155992330)

[8. Diagramy życia encji 7](#_Toc155992331)

[9. Diagram przepływu danych 8](#_Toc155992332)

[10. Diagram sekwencji 8](#_Toc155992333)

# Opis projektu

Projekt MedicaBARP oferuje prosty i skuteczny sposób na zarządzanie wizytami, historią medyczną i komunikacją między pacjentami, a placówką medyczną bezkontaktowo. To rozwiązanie sprzyja szybkiemu dostępowi do opieki zdrowotnej, redukując jednocześnie czas oczekiwania na wizytę.

Projekt placówki medycznej online, opiera się na zaawansowanych technologiach, które zapewniają sprawną i bezpieczną obsługę pacjentów. Backend systemu jest rozwijany przy użyciu języka programowania Python, i platformy phpMyAdmin, co umożliwia efektywne zarządzanie bazą danych oraz logiką biznesową platformy.

Dane pacjentów są przechowywane i zarządzane w bazie danych przy użyciu języka SQL, co zapewnia sprawną analizę informacji.

Frontend aplikacji został zrealizowany przy użyciu technologii HTML i CSS, co pozwala na intuicyjne i responsywne interfejsy użytkownika, dostosowane do różnych urządzeń.

Cały projekt jest kontrolowany za pomocą systemu kontroli wersji Git, co ułatwia efektywną współpracę programistów oraz umożliwia śledzenie zmian w kodzie. Ponadto, repozytorium projektu jest hostowane na platformie GitHub, co ułatwia zarządzanie kodem, udostępnianie aktualizacji oraz monitorowanie postępu prac.

W ramach projektu placówki medycznej online, zespół korzysta z metodyki Scrum w celu efektywnego planowania i przydzielania zadań. Do realizacji tego celu wykorzystywany jest specjalnie dostosowany Scrum board udostępniany na platformie Microsoft Teams.

Dzięki zastosowaniu tych nowoczesnych technologii, placówka medyczna online jest w stanie zapewnić pacjentom wygodny dostęp do usług zdrowotnych, jednocześnie dbając o bezpieczeństwo danych i efektywność obsługi.

# Autorzy

**Alicja Kurczyk** – odpowiedzialna za strukturę i wygląd strony. Zajęła się tworzeniem semantycznych i dostępnych struktur HTML, a także układu CSS, który zapewnia atrakcyjny wygląd i intuicyjny layout strony, dostosowany do potrzeb pacjentów. Opracowała również grafiki, ikony i inne elementy wizualne, które nadają stronie profesjonalny wygląd.

**Bartosz Rabenda** – odpowiedzialny za architekturę systemu bazodanowego, utworzonego w PHP za pomocą SQL, umożliwiającego przechowywanie i pobieranie danych związanych z placówką medyczną. Dzięki temu systemowi możliwe jest efektywne śledzenie historii pacjentów, planowanie wizyt oraz obsługa bieżących operacji administracyjnych w przychodni medycznej. A także możliwość przechowywania dodania nowych danych na temat nowych pacjentów.

**Piotr Ostrowski** – odpowiedzialny za implementację funkcjonalności interaktywnych za pomocą HTML, a także CSS oraz zapewnieniu, aby strona działała sprawnie i efektywnie. Dzięki temu strona jest zgodna z najnowszymi standardami przeglądarek.

**Remigiusz Zając** – odpowiedzialny za backend w projekcie strony placówki medycznej, skoncentrował się na tworzeniu solidnej infrastruktury serwerowej. Jego głównym zadaniem było łączenie strony internetowej z bazą danych, umożliwiając efektywne zarządzanie danymi dotyczącymi lekarzy, pacjentów i innych istotnych informacji. Obsługując funkcje interaktywne zapewnił płynne operacje strony poprzez zastosowanie PHP oraz Pythona. Dzięki jego zaangażowaniu strona działa sprawnie, dostarczając kompleksowe wsparcie dla placówki medycznej online.

# Założenia projektowe

1. Interfejs strony: Stworzenie przyjemnego w użytkowaniu i nowoczesnego interfejsu serwisu.
2. Rejestracja i Logowanie: Implementacja systemu rejestracji oraz logowania dla pacjentów, z uwzględnieniem odpowiednich poziomów dostępu dla użytkowników niezalogowanych.
3. Kalendarz i Rezerwacje Online: Stworzenie interaktywnego kalendarza, umożliwiającego pacjentom rezerwację w pełni funkcjonujących wizyt online.
4. Historia Pacjenta: Opracowanie systemu przechowującego dane, historię pacjenta, umożliwiającego dostęp do informacji o przyszłych i przeszłych wizytach.
5. Baza Danych Medycznych: Stworzenie bezpiecznej bazy danych przechowującej informacje medyczne pacjentów, z zachowaniem zasad ochrony prywatności.

# Ograniczenia i ryzyka projektowe

Ograniczenia:

1. Złożoność Systemu: Ograniczenie związane z złożonością samego systemu, co może sprawić, że kod stanie się trudny do zrozumienia i utrzymania.
2. Brak Jasnich Wymagań: Ograniczenie wynikające z braku jasnych i kompletnych wymagań, co może prowadzić do konieczności wprowadzania zmian w późniejszych fazach projektu.
3. Zależności Technologiczne: Ograniczenia wynikające z zależności od konkretnych technologii, które mogą ograniczać elastyczność systemu oraz możliwość przyszłych rozbudów.
4. Różnorodność Platform: Ograniczenia związane z koniecznością dostosowania kodu do różnych platform (przeglądarki internetowe, urządzenia mobilne), co może wymagać specjalistycznej wiedzy i doświadczenia.

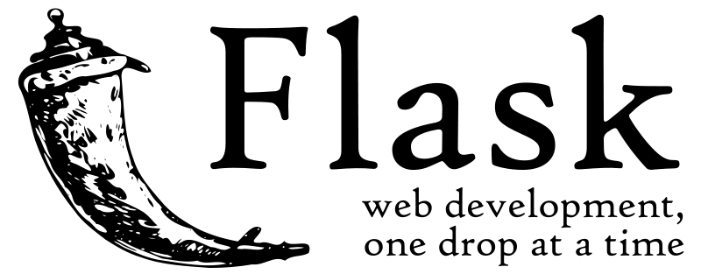
Ryzyka:

1. Błędy Logiki Biznesowej: Ryzyko popełnienia błędów w implementacji logiki biznesowej, co może prowadzić do nieprawidłowego funkcjonowania systemu oraz problemów w obszarze przetwarzania danych.
2. Błędy Bezpieczeństwa: Ryzyko wystąpienia luk w zabezpieczeniach, takich jak niewłaściwa obsługa sesji, niesanitarny input, czy niewłaściwe zarządzanie dostępem, co może prowadzić do zagrożeń dla bezpieczeństwa danych medycznych.
3. Problemy z Wydajnością: Ryzyko związane z niewłaściwą optymalizacją kodu, co może prowadzić do problemów z wydajnością, zwłaszcza gdy system będzie obsługiwał dużą liczbę użytkowników.

# Opis funkcjonalności

Placówka medyczna działa przy pomocy lokalnego serwera, biblioteki Flask używanej przez język Python. Oprócz serwera lokalnie działa również baza danych (o nazwie przychodnia) utworzona dzięki software’owi phpMyAdmin, do którego możemy uzyskać dostęp przy pomocy programu Xampp

1. Logo Flask

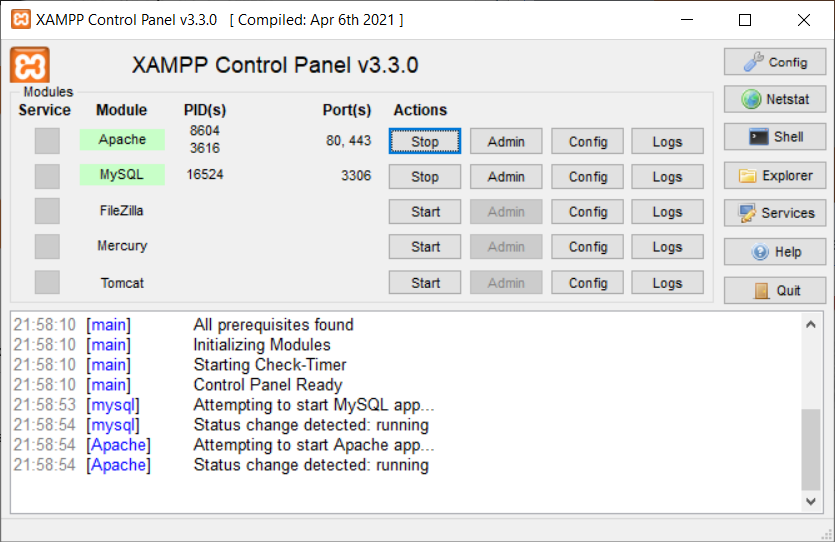


1. Logo Xampp



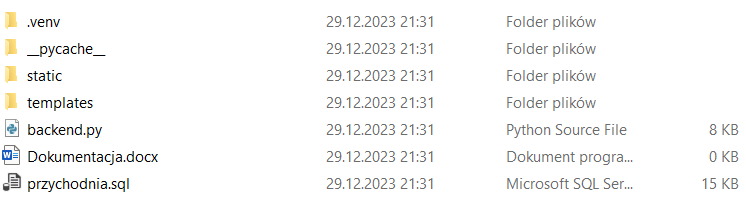
Proces uruchamiania aplikacji:

* Uruchamiamy program Xampp z wcześniejszym przygotowaniem bazy danych. Aby baza danych była dostępna, należy z dostępnych usług uruchomić moduły **Apache** oraz **MySQL**



1. XAMPP Control Panel

* W przypadku chęci zobaczenia bazy danych, możemy wejść do panelu phpMyAdmin poprzez kliknięcie przycisku Admin dla modułu MySQL lub w przeglądarce wpisać adres: [**localhost/phpmyadmin/**](http://localhost/phpmyadmin/)
* Kolejnym krokiem jest uruchomienie serwera Flask, dzięki któremu cała aplikacja będzie działać. Do tego potrzebne są pliki z aplikacji (github.com/RZajacUE/MedicaBARP/) oraz Wiersz Polecenia. W folderze MedicaBARP znajdziemy podfoldery oraz inne pliki



* .venv – wirtualne środowisko, na nim zainstalowane są wszystkie biblioteki dla języka Python, z których korzysta plik backend.py: Flask, mysql-connector-python, colorama oraz inne konieczne do poprawnego działania wyżej wymienionych
* \_\_pycache\_\_ - folder ze skompilowanymi plikami .py
* Static – pliki statyczne, arkusze stylów CSS oraz grafiki
* Templates – pliki HTML przedstawiające strony aplikacji
* Backend.py – backend aplikacji napisany w języku Python
* Dokumentacja.docx – obecny plik
* Przychodnia.sql – plik z bazą danych przychodnia
* Uruchamiając Wiersz Poleceń w folderze MedicaBARP należy wpisać polecenia: **.venv\Scripts\activate.bat** (aktywacja środowiska wirtualnego)  
  **python -m flask --app backend run** (uruchomienie serwera)
* W przeglądarce należy wpisać adres: **127.0.0.1:5000**

**Serwis internetowy został podzielony na zakładki:**

**Strona główna**

Całość aplikacji, strona początkowa z odnośnikami do innych stron oraz podstawowymi informacjami.

**Logowanie**

Logowanie polega na wpisaniu odpowiedniego numeru PESEL oraz wpisaniu odpowiadającego hasła. Występuje tutaj obsługa błędnego logowania, czyli w przypadku podania nieprawidłowego numeru PESEL lub nieprawidłowego hasła otrzymujemy odpowiedni komunikat. Aplikacja wyposażona jest w funkcjonalność utrzymywania sesji użytkownika. Aby zakończyć sesję należy kliknąć w przycisk „Wyloguj się” .

**Rejestracja**

Jeżeli użytkownik nie posiada swojego konta, może je utworzyć. Należy w formularzu wypełnić wszystkie dane. Również zastosowano obsługę błędnej rejestracji. Sprawdzane są pola (czy każde wymagane jest uzupełnione), poprawność hasła, numeru PESEL, adresu e-mail, numeru telefonu przy pomocy wyrażeń regularnych, poprawność powtórzonego hasła, czy któryś z użytkowników posiada ten sam numer PESEL oraz czy zostało zaakceptowane RODO.

**Wizyta**

Miejsce, w którym można umówić wizytę z lekarzem. Każdy lekarz ma dodatkową informację o specjalizacji i numerze pokoju. Wizytę można ustalić w wybrany dzień o pełnej godzinie od 8:00 – 15:00. Tak jak w przypadku rejestracji i logowania obsłużone są możliwe błędy: brak wyboru lekarza bądź terminu, wybór daty która już minęła lub sytuacja w której lekarz w dany termin ma ustaloną inną wizytę. Gdy wizyta zostanie zatwierdzona, następuje przekierowanie na stronę główną.

**Lekarze**

Strona poświęcona informacjom o personelu. Oprócz opisów lekarzy są również ich zdjęcia.

**Oferta**

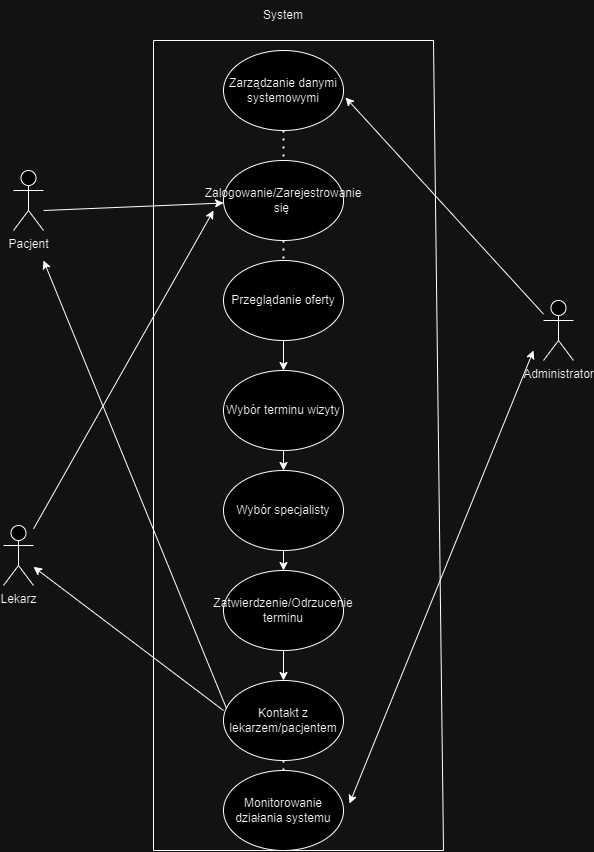
Strona z informacjami o dostępnych usługach w placówce.

**Kontakt**

Dane kontaktowe oraz adres firmy.

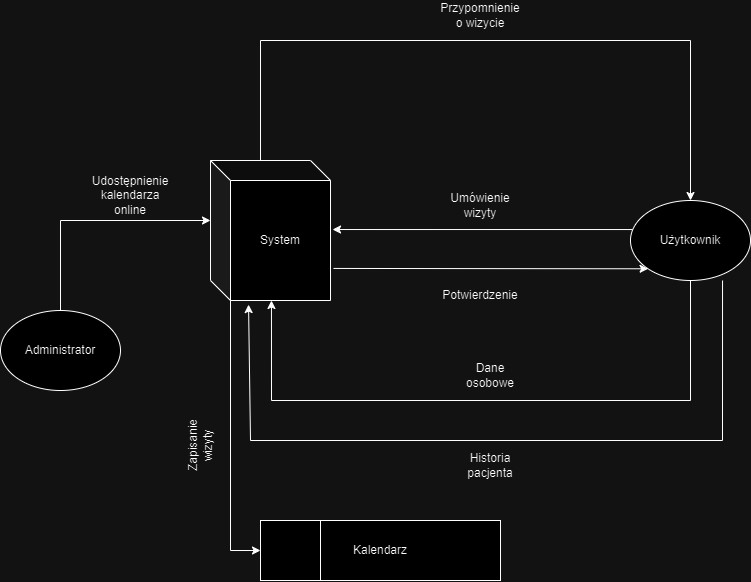
# Diagram związków encji

# Diagram przypadków użycia



# Diagramy życia encji

# Diagram przepływu danych



# Diagram sekwencji

