

PŁife

wykonane przez:

Anhelina Mendohralo, Valeria Zlydar, Miraslau Alkhovik, Praskouya Horbach

Spis treści:

1. Cel dokumentacji	3
2. Struktura projektu	4
3. Technologie i narzędzia	5
4. Instalacja i uruchomienie	6
5. Stylowanie	8
6. Interakcja użytkownika	9
7. Zarządzanie stanem i obsługa błędów	18
8. Dokumentacja API	19
9. Utrzymanie i rozwój	19
10. Autorzy i kontakt	20

Cel dokumentacji

Dokumentacja front-endu aplikacji ma na celu zapewnienie jasnych i zrozumiałych informacji na temat struktury, funkcjonalności oraz sposobów utrzymania i rozwoju aplikacji. Jest to narzędzie, które ma pomóc zarówno obecnym, jak i przyszłym deweloperom w efektywnym zarządzaniu projektem.

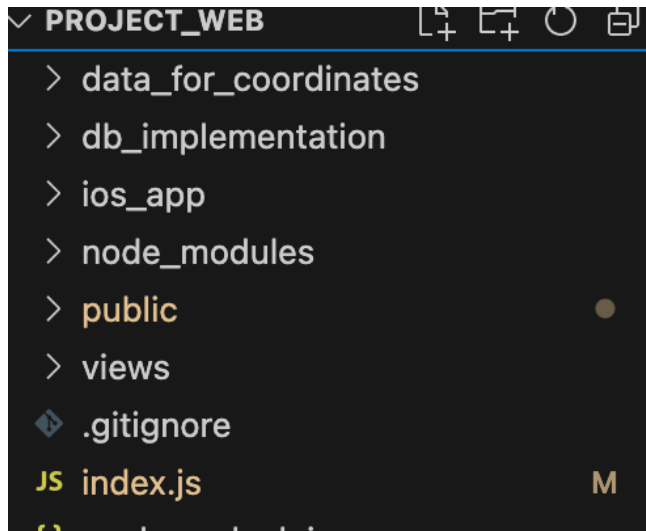
Główne cele:

- **Zrozumienie Struktury Projektu**
 - Opis i organizacja: Dokumentacja dostarcza szczegółowego opisu struktury katalogów i plików, co ułatwia nowym członkom zespołu zrozumienie organizacji projektu.
 - Komponenty: Wskazuje na główne komponenty aplikacji, ich role i sposób interakcji między nimi.
- **Ułatwienie Instalacji i Konfiguracji**
 - Instrukcje instalacji: Krok po kroku opisuje proces instalacji i uruchomienia projektu, co jest kluczowe dla nowych deweloperów lub w przypadku przenoszenia projektu na nowe środowisko.
 - Wymagania systemowe: Podaje wymagania dotyczące oprogramowania i narzędzi niezbędnych do pracy z projektem.
- **Zachowanie Spójności w Kodzie**
 - Standardy kodowania: Opisuje konwencje i najlepsze praktyki dotyczące stylu kodowania, co pomaga w utrzymaniu spójności i czytelności kodu.
 - Stylowanie: Zawiera informacje na temat zasad stylowania, używanych czcionek, kolorystyki i innych aspektów wizualnych.
- **Dokumentacja Funkcjonalności i Interakcji**
 - Opis funkcji: Detalicznie opisuje funkcjonalności aplikacji, w tym interakcji użytkownika, takich jak mikroanimacje, różnorodne metody wejścia (klawiatura, klikanie, przeciąganie) i responsywność interfejsu.
 - Zarządzanie stanem: Wyjaśnia, jak stan aplikacji jest zarządzany, co jest kluczowe dla zrozumienia przepływu danych w aplikacji.
- **Obsługa Błędów i Testowanie**
 - Obsługa błędów: Opisuje mechanizmy obsługi błędów, co pomaga w diagnozowaniu i naprawianiu problemów.
 - Testowanie: Zawiera instrukcje dotyczące testowania aplikacji, co jest kluczowe dla utrzymania jakości i stabilności projektu.
- **Wsparcie i Dokumentacja API**
 - API: Opisuje, jak front-end komunikuje się z backendem, w tym szczegóły dotyczące endpointów i przykłady zapytań oraz odpowiedzi.
 - Pomoc i dokumentacja: Dostarcza zasobów, które pomagają użytkownikom i deweloperom w pełnym wykorzystaniu możliwości aplikacji.
- **Utrzymanie i Rozwój**
 - Proces wprowadzania zmian: Opisuje zasady i procesy dotyczące wprowadzania nowych funkcji i poprawek, co jest kluczowe dla ciągłego rozwoju projektu.
 - Najlepsze praktyki: Zawiera wskazówki dotyczące utrzymania wysokiej jakości kodu i efektywnej współpracy w zespole.

- **Informacje Kontaktowe**

- Autorzy: Przedstawia twórców aplikacji i ich role.
- Kontakt: Podaje informacje, jak skontaktować się z zespołem w razie pytań lub problemów.

Struktura projektu



Główne Katalogi i Pliki:

- **data_for_coordinates**
 - Katalog zawiera pliki i dane niezbędne do obsługi bazy danych, związane głównie z przechowywaniem współrzędnych geograficznych.
 - Pliki z danymi, procedurami SQL oraz innymi skryptami związanymi z bazą danych.
- **db_implementation**
 - Katalog ten jest przeznaczony do implementacji baz danych. Zawiera skrypty SQL oraz procedury niezbędne do zarządzania i manipulacji danymi w bazie.
 - Procedury, schematy bazy danych, skrypty inicjalizacyjne.
- **ios_app**
 - Katalog zawiera szkielet aplikacji mobilnej na platformę iOS. Zawiera pliki związane z widokiem oraz podstawową funkcjonalnością aplikacji mobilnej.
 - Pliki widoków, kontrolery, zasoby multimedialne oraz inne zasoby specyficzne dla iOS.
- **node_modules**
 - Katalog zawierający wszystkie zależności Node.js potrzebne do uruchomienia aplikacji.
 - Biblioteki i moduły zainstalowane za pomocą npm.
- **public**
 - Katalog przeznaczony na publicznie dostępne zasoby aplikacji internetowej.
 - Pliki CSS, obrazy, skrypty JavaScript oraz inne zasoby statyczne.

- **views**
 - Katalog zawierający pliki widoków EJS (Embedded JavaScript).
 - Pliki .ejs używane do renderowania dynamicznych widoków na stronie internetowej.
- **.gitignore**
 - Plik konfiguracji Git, który określa, które pliki i katalogi mają być ignorowane przez system kontroli wersji.
 - Reguły ignorowania plików.
- **index.js**
 - Główny plik JavaScript uruchamiający aplikację.
 - Logika serwera, konfiguracja aplikacji, połączenia z bazą danych. Szczegółowa Struktura.

Technologie i narzędzia

Front-End

EJS (Embedded JavaScript)

- EJS jest silnikiem szablonów, który pozwala na generowanie HTML z kodem JavaScript. Wykorzystujemy EJS do dynamicznego renderowania widoków dla każdej funkcjonalności naszej strony internetowej.
- Każda strona aplikacji (takie jak strona główna, kalendarz, posty, mapa) jest renderowana za pomocą plików EJS, co umożliwia łatwe zarządzanie dynamicznymi treściami i logiką wyświetlania.

CSS

- Kaskadowe Arkusze Stylów (CSS) są używane do stylizacji naszej strony internetowej.
- CSS jest używany do nadawania stylów wszystkim elementom strony, w tym układom, kolorom, typografii i animacjom. Przykłady zastosowania CSS obejmują stylizację przycisków, tła, formularzy oraz interfejsu użytkownika.

JavaScript

- JavaScript jest językiem programowania używanym do dodawania interaktywności do naszej strony internetowej.
- JavaScript jest wykorzystywany do obsługi dynamicznych elementów strony, takich jak animowane przyciski, walidacja formularzy, interakcje użytkownika oraz integracja

z backendem. Skrypty JavaScript umożliwiają także ładowanie treści bez przeładowania strony oraz obsługę zdarzeń użytkownika.

Back-End

T-SQL (Transact-SQL)

- T-SQL jest rozszerzeniem języka SQL używanym do zarządzania i manipulacji danymi w bazie danych Microsoft SQL Server.
- T-SQL jest wykorzystywany do tworzenia i zarządzania bazą danych naszej aplikacji, w tym tabelami, procedurami składowanymi oraz zapytaniami. Dane są przechowywane i pobierane z bazy danych za pomocą T-SQL.

SSMS (SQL Server Management Studio)

- SSMS to zintegrowane środowisko zarządzania bazami danych SQL Server.
- SSMS jest używane do projektowania, konfigurowania i zarządzania bazą danych. Deweloperzy używają SSMS do pisania, testowania i uruchamiania skryptów T-SQL.

Node.js

- Node.js to środowisko uruchomieniowe JavaScript, które pozwala na uruchamianie JavaScript poza przeglądarką.
- Node.js jest używane do tworzenia serwera backendowego, który obsługuje żądania HTTP, łączy się z bazą danych i przesyła dane do i z aplikacji front-endowej. Skrypty w Node.js umożliwiają obsługę logiki serwera oraz komunikację z bazą danych SQL Server.

Aplikacja Mobilna

Swift

- Swift jest nowoczesnym językiem programowania stworzonym przez Apple do tworzenia aplikacji na iOS, macOS, watchOS i tvOS.
- Aplikacja mobilna naszej aplikacji została napisana w Swift. Swift umożliwia tworzenie wydajnych, bezpiecznych i interaktywnych aplikacji mobilnych. Widoki, kontrolery oraz logika aplikacji mobilnej są implementowane w Swift.

Instalacja i uruchomienie

1. Klonowanie Repozytorium

Aby rozpocząć pracę z projektem, najpierw musisz sklonować repozytorium na swoje lokalne środowisko. Wykonaj następujące kroki:

Otwórz terminal. Wykonaj polecenie:

```
bash: git clone [URL_do_repozytorium]
```

1. Zastąp [URL_do_repozytorium] aktualnym adresem URL do repozytorium projektu.

2. Instalacja Zależności

Przejdź do katalogu projektu i zainstaluj wszystkie zależności Node.js:

W terminalu, przejdź do katalogu projektu:

```
bash: cd PROJECT_WEB
```

Zainstaluj zależności:

```
bash: npm install
```

3. Konfiguracja Bazy Danych

Aby skonfigurować bazę danych, musisz załadować skrypty SQL do SQL Server Management Studio (SSMS).

1. Otwórz SQL Server Management Studio (SSMS).
2. Połącz się z lokalnym lub zdalnym serwerem SQL Server.
3. Utwórz nową bazę danych lub użyj istniejącej.
4. Załaduj skrypty SQL z katalogu `db_implementation` oraz `data_for_coordinates`:
 - **Przykład:**
 - Otwórz plik `schema.sql` i uruchom go w SSMS, aby utworzyć struktury tabel.
 - Otwórz plik `procedures.sql` i uruchom go, aby dodać procedury składowane.
 - Otwórz plik `coordinates_data.sql` i uruchom go, aby załadować dane geograficzne.

4. Konfiguracja Klucza API Google Maps

Aby aplikacja mogła korzystać z Google Maps, musisz uzyskać własny klucz API i wstawić go do kodu.

1. Przejdź na stronę Google Cloud Console.
2. Utwórz nowy projekt lub wybierz istniejący.
3. W sekcji "API & Services" wygeneruj nowy klucz API.
4. Skopiuj wygenerowany klucz.
5. Otwórz plik `views/map.ejs` i przejdź do linii 195.

Wstaw swój klucz API w odpowiednim miejscu w kodzie.

Przykład:

html

```
<script
```

```
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callba  
ck=initMap" async defer></script>
```

6. Zamień **YOUR_API_KEY** na twój rzeczywisty klucz API.

5. Uruchomienie Aplikacji

Aby uruchomić aplikację, wykonaj następujące kroki:

Otwórz terminal. Przejdź do katalogu projektu, jeśli jeszcze tego nie zrobiłeś:

```
bash cd PROJECT_WEB
```

Uruchom serwer Node.js:

```
bash: node index.js
```

Aplikacja powinna teraz działać na lokalnym serwerze. Możesz otworzyć przeglądarkę i przejść do <http://localhost:3000>, aby zobaczyć działającą stronę internetową.

Podsumowując, aby zainstalować i uruchomić aplikację, należy wykonać następujące kroki:

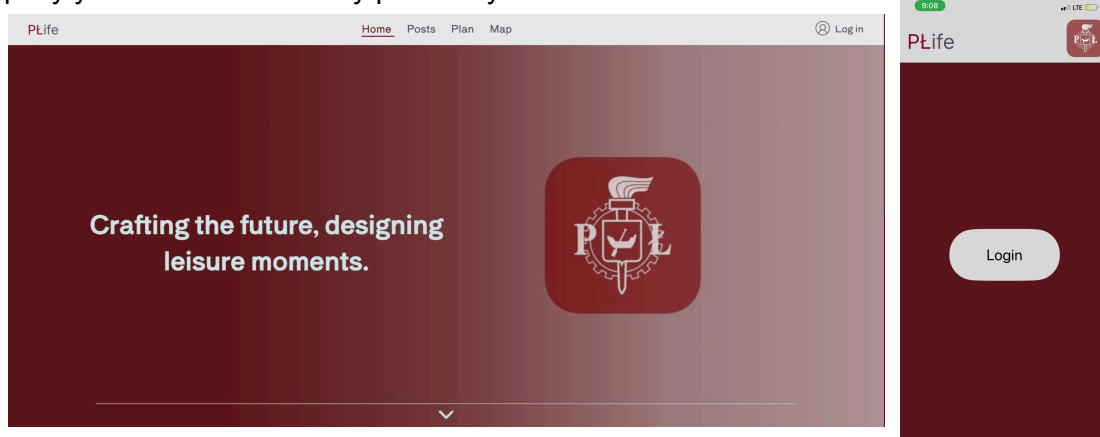
1. Klonowanie repozytorium.
2. Instalacja zależności Node.js.
3. Konfiguracja i załadowanie bazy danych w SSMS.
4. Konfiguracja klucza API Google Maps.
5. Uruchomienie serwera Node.js.

Po wykonaniu tych kroków, aplikacja powinna być gotowa do użycia.

Stylowanie

Czcionka

W naszej stronie internetowej i aplikacji mobilnej używamy czcionki "Shippori Antique", która jest czcionką bezszeryfową. Dzięki temu tekst jest czytelny i estetyczny, co wpływa pozytywnie na odbiór strony przez użytkowników.



Kolory

Paleta kolorów na naszej stronie i aplikacji jest spójna i harmonijna. Używamy następujących kolorów:

- **Kolor tła (background-color):** głęboki odcień czerwieni (rgb(94, 23, 26)).
- **Kolor tła nagłówka (header-background):** jasnoszary (#E9E9E9).
- **Kolor tekstu nagłówka (header-text):** ciemny szary (#4D4C5C).
- **Kolor paska przewijania (scrollbar):** intensywny odcień fioletu (rgb(134, 0, 40)).
- **Kolor głównego tekstu (main-text):** jasny odcień błękitu (rgb(204, 230, 233)).

Te kolory są zdefiniowane jako zmienne CSS, co ułatwia ich zarządzanie i ewentualne modyfikacje.

Zasady Typografii

W naszej typografii stosujemy zasadę ograniczenia liczby używanych czcionek do maksymalnie dwóch. W rzeczywistości, używamy tylko jednej czcionki "Shippori Antique", co zapewnia spójność wizualną. Teksty są stylizowane w jednolity sposób, co ułatwia ich czytanie i poprawia estetykę strony.

- **Nagłówki (headers):** Stylizowane przy użyciu tej samej czcionki i kolorów, co zapewnia spójność i łatwość w identyfikacji.
- **Paragrafy (paragraphs):** Utrzymane w jednolitej kolorystyce, z odpowiednim odstępem między liniami tekstu, co poprawia czytelność.
- **Linki (links):** Stylizowane tak, aby były wyraźne i łatwe do zauważenia, z efektem podkreślenia po najechaniu kursorem.

Stylowanie naszej strony internetowej jest proste, spójne i estetyczne. Dzięki zastosowaniu jednej czcionki oraz określonej palety kolorów, strona jest łatwa w nawigacji i przyjemna w odbiorze. Zastosowanie zmiennych CSS ułatwia zarządzanie kolorami i stylami, co pozwala na szybkie wprowadzanie ewentualnych zmian i modyfikacji.

Interakcje użytkownika

Widoki strony internetowej

Widok Główny

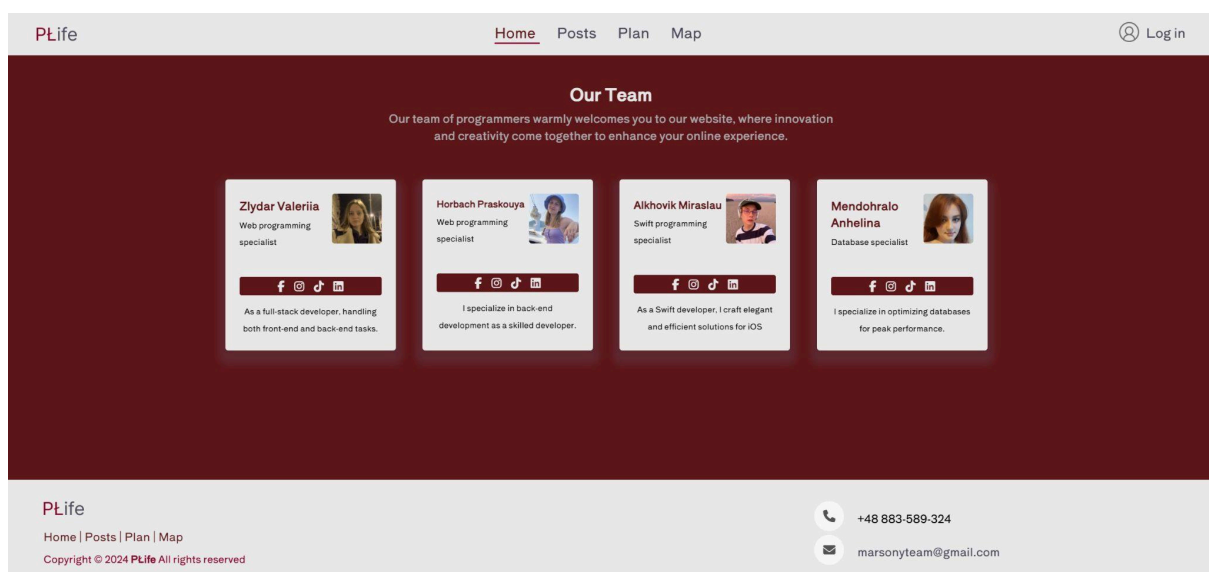
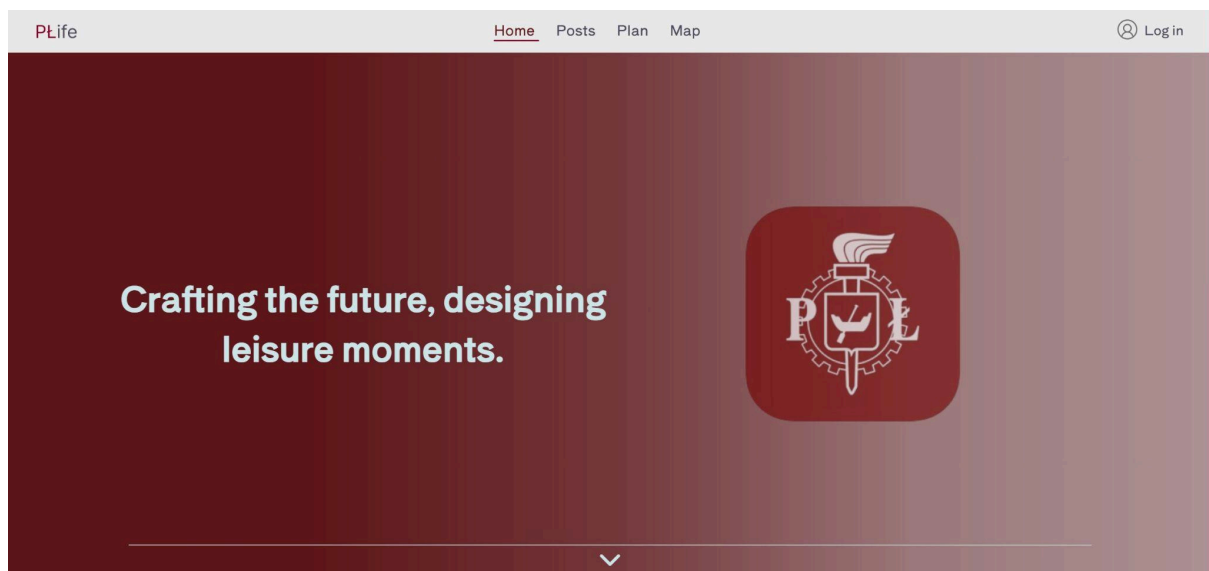
1. Strzałka w Dół i Scrollowanie:

- Po naciśnięciu strzałki w dół (zawierającej mikroanimację) lub za pomocą scrollowania, użytkownik może przewinąć stronę w dół, aby zobaczyć informacje dotyczące twórców strony.

- Bloki ze zdjęciami i informacjami twórców strony mają animacje typu hover, co oznacza, że po najechaniu kursorem zmieniają się w jakiś sposób (np. zmieniają kolor, powiększają się).
- Aktywne linki w tych blokach również mają efekty hover, podkreślając interaktywność.
- Przewijając dalej, użytkownik może zobaczyć stopkę (footer) strony.

2. Przycisk Login:

- W prawym górnym rogu strony znajduje się przycisk "Login".
- Po jego naciśnięciu użytkownik jest przekierowany do formularza logowania/rejestracji.
- Podczas rejestracji użytkownik musi zaakceptować RODO, aby zakończyć proces tworzenia konta.



The image shows the registration form of the PLife application. The form is titled "Registration" and is located in the center of the screen. It contains the following fields: "University ID", "First Name", "Last Name", and "Password". Each field has a lock icon indicating it is required. Below the "Password" field, there is a checkbox labeled "I accept" and a link "Rules of Privacy". At the bottom of the form, there is a "Register" button and a link "Already have account? Login". The background of the application shows a dark red header with the "PLife" logo and navigation links "Home", "Posts", "Plan", and "Map". The main content area has a dark red background with the text "Crafting the future, d... leisure moment" and a large "P L" logo on the right.

The image shows the login form of the PLife application. The form is titled "Login" and is located in the center of the screen. It contains the following fields: "Username" and "Password". Each field has an icon indicating its type (envelope for username, lock for password). Below the "Password" field, there is a checkbox labeled "Remember me" and a link "Forgot Password?". At the bottom of the form, there is a "Login" button and a link "Don't have an account? Register". The background of the application shows a dark red header with the "PLife" logo and navigation links "Home", "Posts", "Plan", and "Map". The main content area has a dark red background with the text "Crafting the future, d... leisure moment" and a large "P L" logo on the right.

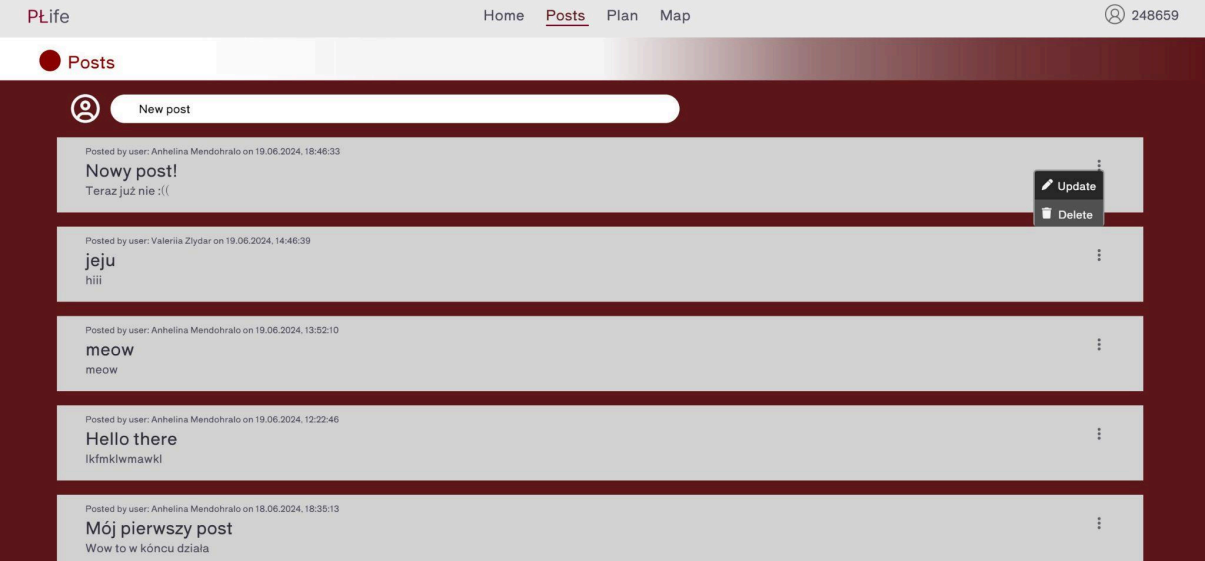
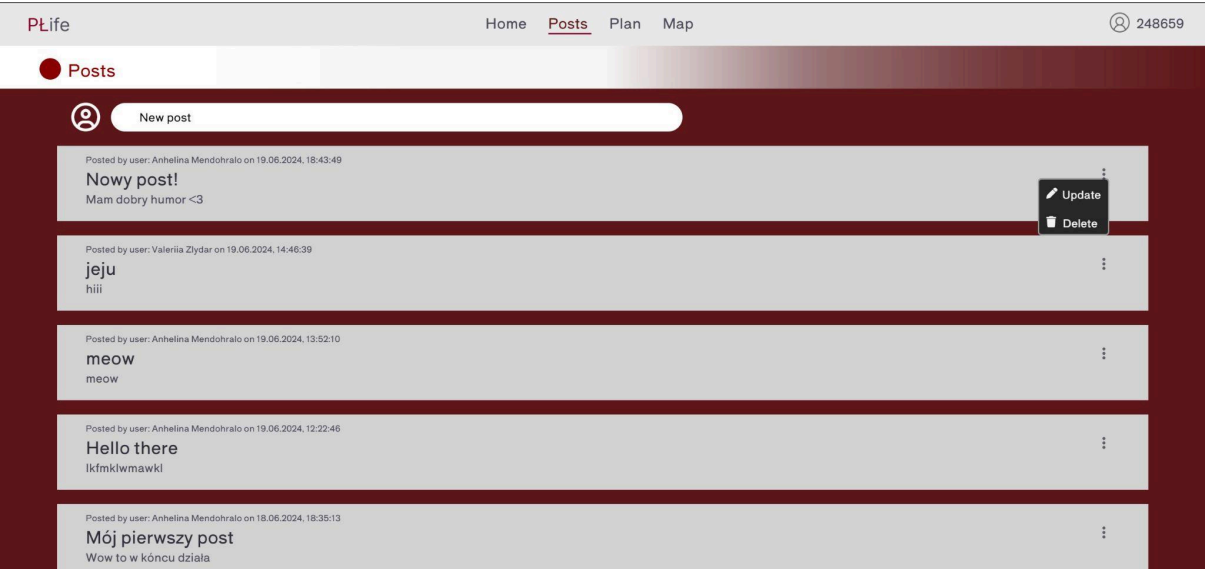
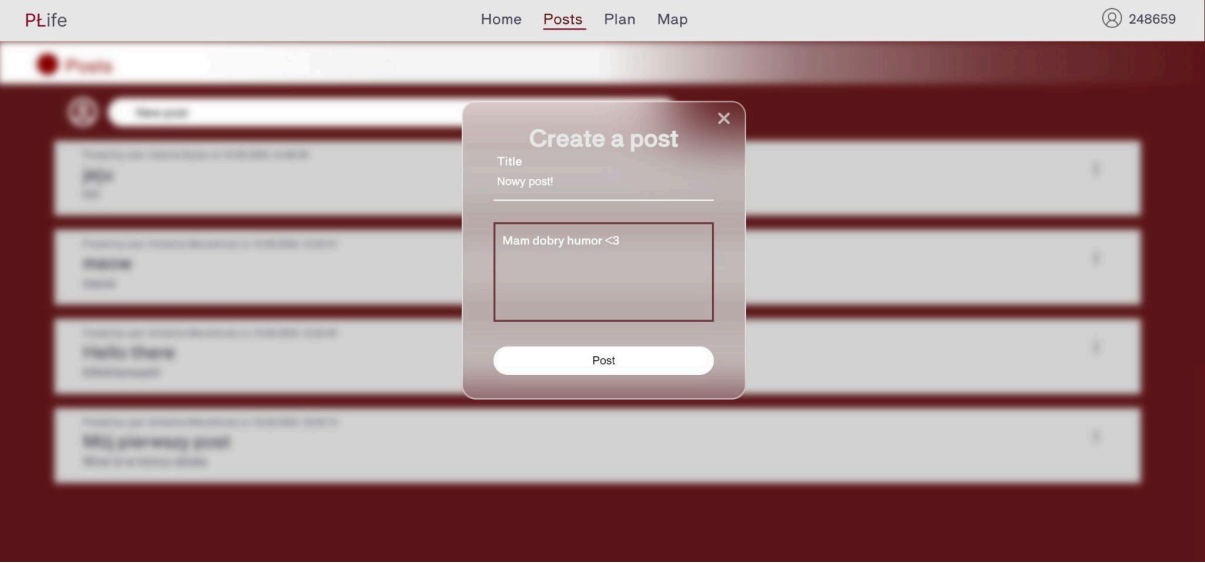
Widok Postów

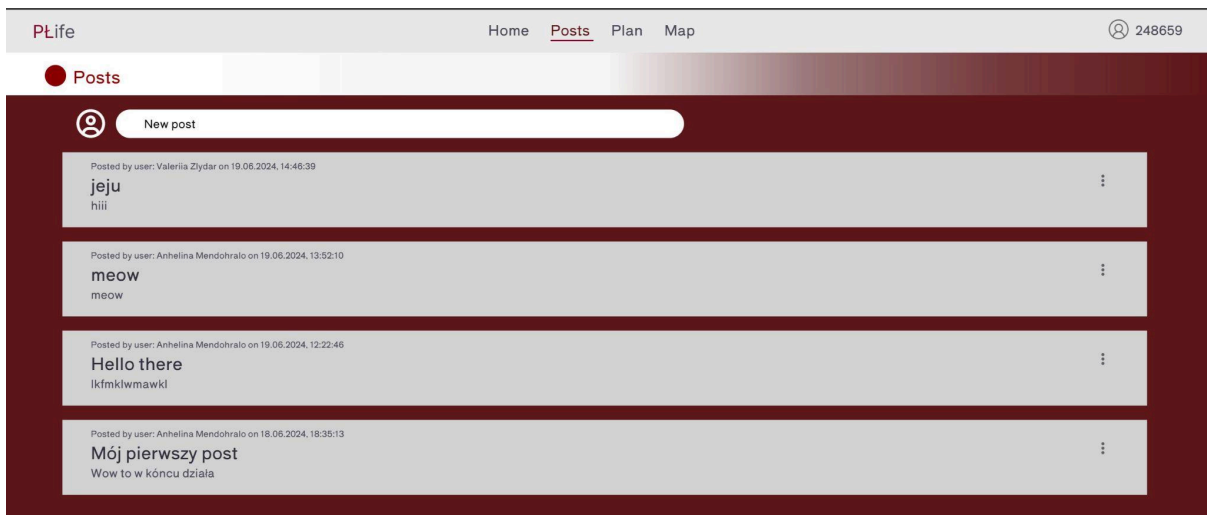
1. Dodawanie Postów:

- Pod nagłówkiem strony znajduje się przycisk do dodawania postów.
- Po naciśnięciu tego przycisku, użytkownik może wprowadzić nazwę i tytuł postu.

2. Modyfikacja/Usunięcie Postów:

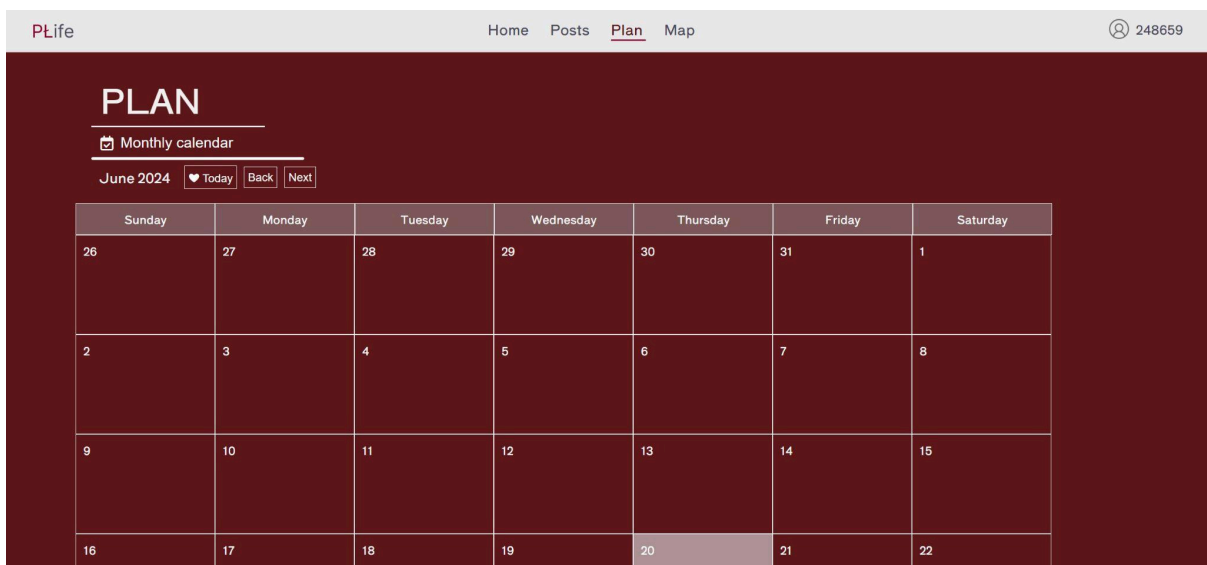
- Po prawej stronie każdego postu znajdują się trzy kropki.
- Po naciśnięciu tych kropek, użytkownik ma możliwość modyfikacji lub usunięcia postu.





Widok Kalendarza

- Przemieszczanie się po Miesiącach:**
 - Użytkownik może używać przycisków "Back" i "Next", aby przemieszczać się po miesiącach w kalendarzu.
- Powrót do Dnia Dzisiejszego:**
 - Po naciśnięciu przycisku "Today", użytkownik wraca do dzisiejszej daty i miesiąca.
- Tworzenie Wydarzeń:**
 - Po kliknięciu na konkretną datę w kalendarzu, użytkownik może stworzyć nowe wydarzenie.
 - Jeśli jest za dużo wydarzeń na dany dzień, kliknięcie na tę datę otwiera okienko z listą wszystkich wydarzeń przypisanych do tego dnia.



PLife

HomePostsPlanMap

248659

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	26	27	28	29	30	31	1
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29

Events for 6/21/2024

ywe

DetailsDelete

whru

DetailsDelete

Add EventClose

PLife

HomePostsPlanMap

248659

	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6

New Event

nowy event!

08:00

09:00

plwot!

SaveCancel

PLife

HomePostsPlanMap

248659

	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
	30	1	2	3	4	5	6

Event Details

ywe

6/21/202403:59 - 09:03

uksefh

Close

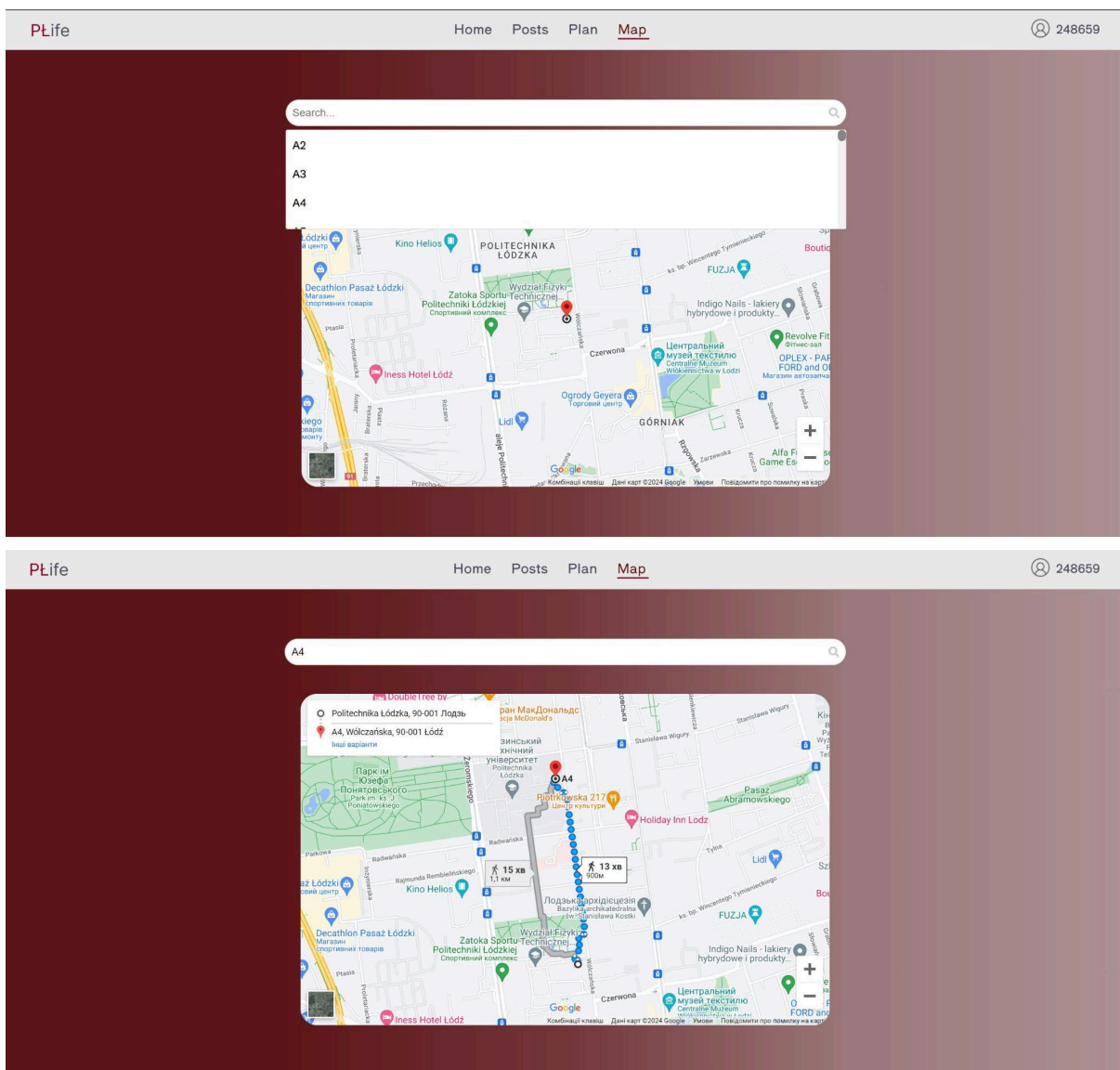
Widok Mapy

1. Wyszukiwanie Budynków:

- Po naciśnięciu na pasek wyszukiwania (search bar), użytkownik może wyszukać budynek lub wybrać go z rozwijanego menu (dropdown menu).
- Po wyborze budynku na mapie pojawi się ścieżka piesza prowadząca od aktualnej lokalizacji użytkownika do wejścia wybranego budynku.

2. Interakcje z Mapą:

- Użytkownik może przeciągać widok mapy, aby zobaczyć różne obszary.
- Jest również możliwość zmiany widoku mapy na tryb satelitarny.



Widoki aplikacji mobilnej

Widok Główny

1. Przycisk Logowania:

- Na głównym ekranie aplikacji znajduje się przycisk logowania.
- Po naciśnięciu przycisku logowania pojawia się okno pop-up (slide-in window), w którym użytkownik może wprowadzić swój login (username) i hasło.

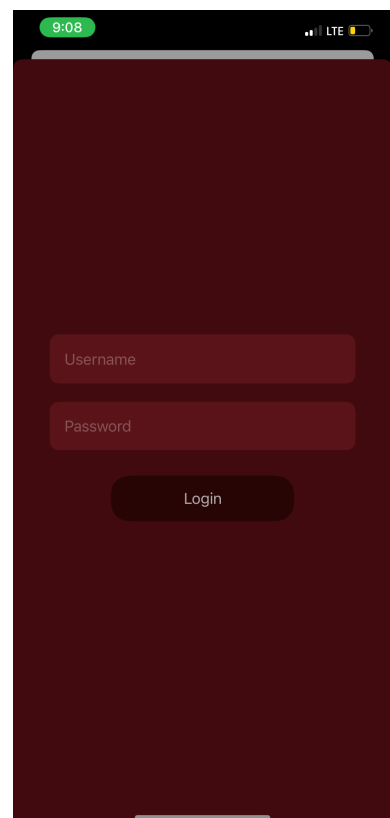
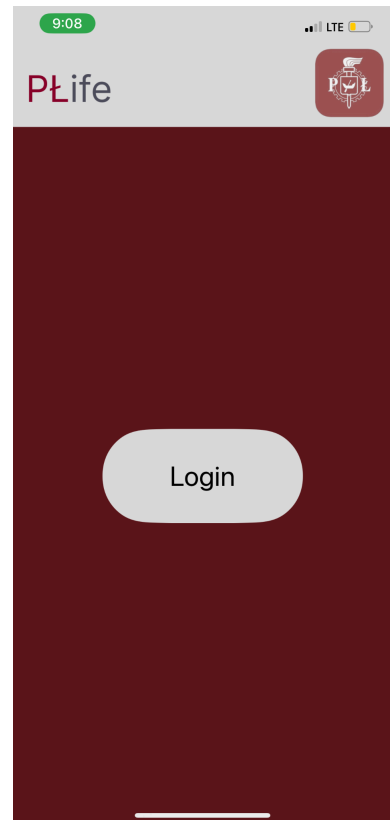
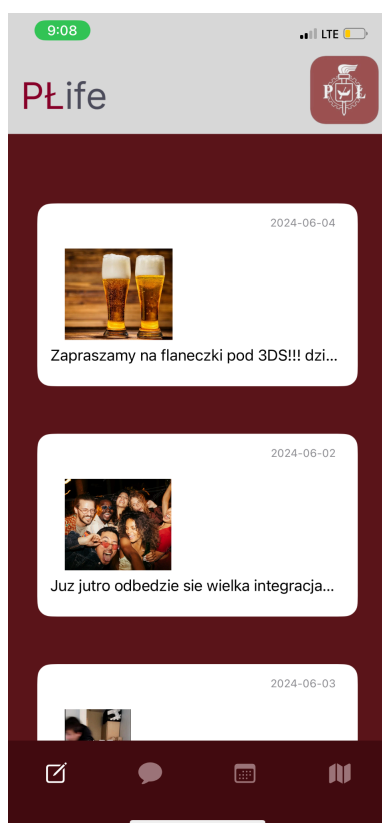
2. Po Zalogowaniu:

- Po pomyślnym zalogowaniu się użytkownik zostaje automatycznie przeniesiony do widoku postów, który jest ustawiony jako domyślny.
- Na dole ekranu pojawia się pasek nawigacyjny (navigation bar) z domyślnymi i zrozumiałymi ikonami umożliwiającymi nawigację między widokami.

Widok Postów (Posts)

• Domyślny Widok Po Zalogowaniu:

- Użytkownik zostaje automatycznie przeniesiony do widoku postów po zalogowaniu.
- W tym widoku można przeglądać wszystkie posty dodane przez użytkowników.



Widok Powiadomień (Messages)

1. Przycisk Powiadomień:

- Na pasku nawigacyjnym znajduje się przycisk powiadomień (messages).
- Po naciśnięciu tego przycisku użytkownik zostaje przeniesiony do sekcji prywatnych wiadomości (private messages).

2. Przeglądanie Wiadomości:

- W tej sekcji użytkownik może przeglądać wiadomości od znajomych.
- Każda wiadomość wyświetla imię i nazwisko nadawcy, czas wysłania, jego avatar oraz treść wiadomości.



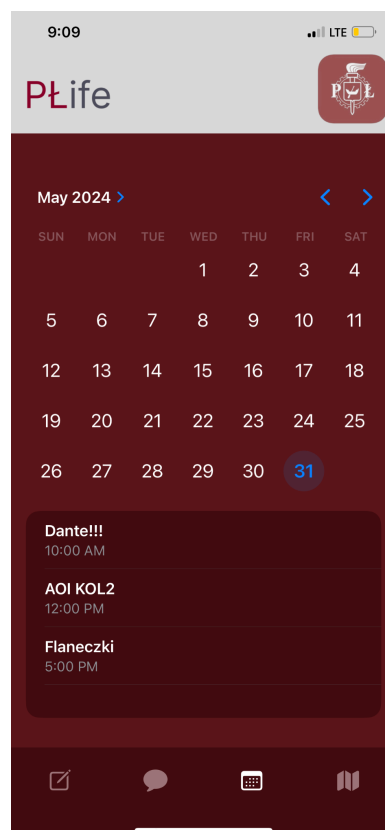
Widok Kalendarza (Calendar)

1. Przycisk Kalendarza:

- Na pasku nawigacyjnym znajduje się przycisk kalendarza.
- Po naciśnięciu tego przycisku użytkownik zostaje przeniesiony do widoku kalendarza.

2. Przeglądanie Kalendarza:

- W widoku kalendarza użytkownik może zobaczyć wszystkie nadchodzące wydarzenia.
- Po kliknięciu na konkretną datę można zobaczyć listę wydarzeń przypisanych do tego dnia.



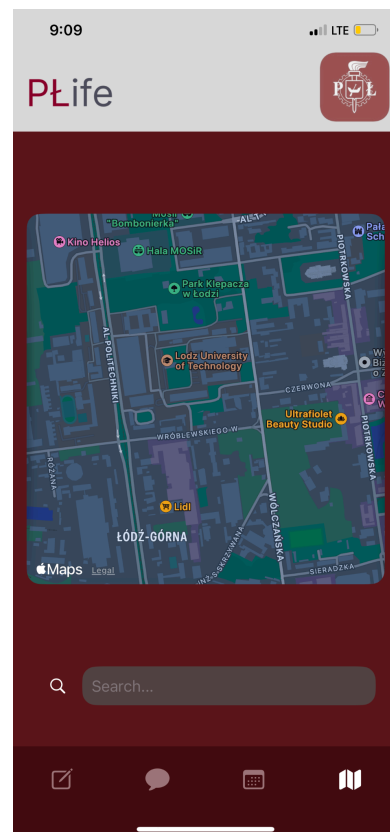
Widok Mapy (Map)

1. Przycisk Mapy:

- Na pasku nawigacyjnym znajduje się przycisk mapy.
- Po naciśnięciu tego przycisku użytkownik zostaje przeniesiony do widoku zintegrowanej mapy.

2. Interakcja z Mapą:

- W widoku mapy użytkownik ma dostęp do zintegrowanej mapy oraz paska wyszukiwania (search bar).
- Użytkownik może wyszukiwać lokalizacje i budynki oraz przeglądać mapę w celu znalezienia potrzebnych informacji.



Zarządzanie stanem i obsługa błędów

- **Stan Logowania:** Po zalogowaniu użytkownika, jego nazwa użytkownika (username) jest wyświetlana w prawym górnym rogu ekranu, co pozwala łatwo zidentyfikować, że jest zalogowany.
- **Komunikaty o Błędach:** W przypadku wprowadzenia nieprawidłowych danych logowania, na ekranie pojawiają się odpowiednie komunikaty błędów, informujące użytkownika o problemie.
- **Obsługa Błędów:** Każdy potencjalny problem jest odpowiednio obsługiwany, a wszystkie błędy są logowane i wyświetlane w konsoli deweloperskiej, co ułatwia diagnostykę i naprawę.

Dokumentacja API

SSMS (SQL Server Management Studio)

- **Baza Danych:** SSMS zawiera bazę danych wraz z procedurami, które zarządzają danymi użytkowników i innymi niezbędnymi informacjami.
- **Odporność na SQL Injections:** Baza danych jest zabezpieczona przed atakami typu SQL injection, co zwiększa jej bezpieczeństwo.
- **Hasła:** Wszystkie hasła są zahashowane, co oznacza, że nawet administrator nie ma dostępu do rzeczywistych haseł użytkowników.

Skrypt w `index.js`

- **Request do Bazy Danych:** Skrypt `index.js` wysyła zapytania do bazy danych w SSMS, gdzie są one przetwarzane.
- **Przetwarzanie i Odpowiedź:** Wyniki zapytań są następnie zwracane do strony, gdzie są wyświetlane użytkownikowi.

Utrzymanie i rozwój

Strona Internetowa

- **Lokalne Działanie:** Strona jest zaprojektowana tak, aby działała lokalnie, co ułatwia jej rozwój i testowanie.
- **Instrukcje Instalacji i Uruchomienia:** Wszystkie niezbędne instrukcje dotyczące instalacji i uruchomienia znajdują się w odpowiednim dziale dokumentacji, co umożliwia szybkie i bezproblemowe wdrożenie strony.
- **Gotowość do Użytku:** Strona jest gotowa do użytku i może być łatwo wdrożona w środowisku produkcyjnym.

Aplikacja Mobilna

- **Wymagania Backendowe:** Aplikacja mobilna wymaga działającego backendu, aby mogła prawidłowo funkcjonować i komunikować się z bazą danych.
- **Rozwój:** Aplikacja mobilna jest zaprojektowana z myślą o przyszłym rozwoju, z elastyczną architekturą, która umożliwia łatwe dodawanie nowych funkcjonalności.

Autorzy i kontakt

Valeria Zlydar: 248677@edu.p.lodz.pl

Anhelina Mendoralo: 248659@edu.p.lodz.pl

Miraslau Alkhovik: 248655@edu.p.lodz.pl

Praskouya Horbach: 248656@edu.p.lodz.pl