

# Specyfikacja projektu: System Wyszukiwania Lotów

## Opis projektu:

Projekt jest systemem umożliwiającym wyszukiwanie dostępnych połączeń lotniczych na podstawie wybranych lotnisk wylotu i przylotu oraz daty. Wyszukiwanie odbywa się w oparciu o bazę danych zawierającą szczegóły lotów, takie jak linie lotnicze, numery lotów, lotniska wylotu i przylotu, godziny wylotu i przylotu, oraz dzień tygodnia. System oblicza także przewidywany czas lotu na podstawie współrzędnych geograficznych lotnisk.

## Zakres projektu:

### 1. Wyszukiwarka lotów:

2. Użytkownicy mogą wyszukiwać dostępne połączenia lotnicze, podając:

- a. Lotnisko wylotu
- b. Lotnisko przylotu
- c. Datę wylotu

### 3. Panel administratora:

Panel administracyjny umożliwia:

- a. Dodawanie nowych lotów do bazy danych (z linią lotniczą, numerem lotu, lotniskami, godzinami, dniem tygodnia)
- b. Usuwanie istniejących lotów na podstawie numeru lotu
- c. Zabezpieczenie hasłem dla dostępu do panelu administracyjnego

### 4. Baza danych:

Baza danych zawiera dane o lotach, takie jak:

- a. Linia lotnicza
- b. Numer lotu
- c. Lotnisko wylotu i przylotu
- d. Dzień tygodnia (numer dnia)
- e. Godzina wylotu
- f. Czas trwania lotu

### 5. Algorytm obliczania odległości i czasu lotu:

System korzysta z geolokalizacji, aby obliczyć odległość między lotniskami na podstawie ich współrzędnych geograficznych. Na tej podstawie system wylicza szacunkowy czas trwania lotu.

### ***Technologie użyte w projekcie:***

#### **1. Flask (Python):**

Flask jest lekkim frameworkiem webowym w języku Python, który obsługuje backend aplikacji. Używamy go do obsługi zapytań HTTP, renderowania stron HTML i zarządzania logiką aplikacji.

#### **2. SQLite:**

Jako system zarządzania bazą danych używamy SQLite, który przechowuje dane dotyczące lotów. Jest to lekka, relacyjna baza danych, dobrze pasująca do aplikacji o mniejszej skali, jak nasz projekt.

#### **3. HTML/CSS (Bootstrap):**

Frontend aplikacji został zbudowany z użyciem HTML i CSS, gdzie w celu zwiększenia estetyki strony, zastosowano bibliotekę Bootstrap do tworzenia responsywnych i atrakcyjnych interfejsów użytkownika.

#### **4. OpenCage Geocoding API:**

W celu obliczenia odległości pomiędzy lotniskami, projekt korzysta z API OpenCage Geocoding, które zapewnia możliwość przekształcania nazw lotnisk na współrzędne geograficzne (szerokość i długość geograficzną). Na podstawie tych współrzędnych obliczamy odległość między dwoma punktami na Ziemi za pomocą wzoru Haversine.

#### **5. Wzór Haversine:**

Do obliczenia odległości pomiędzy dwoma punktami na Ziemi, zastosowano wzór Haversine, który uwzględnia krzywiznę Ziemi i zapewnia dokładność obliczeń dla małych i średnich odległości.

#### **6. Flash Messages (Flask):**

Do wyświetlania komunikatów o statusie operacji (np. dodanie lotu, usunięcie lotu), wykorzystano mechanizm **flash messages** w Flasku, który pozwala na przekazywanie informacji między stronami.

### ***Zakładane dane do bazy:***

Na potrzeby projektu, baza danych zawiera ogromną liczbę fałszywych (fejkowych) danych lotów, które zostały wygenerowane w sposób losowy. Wygenerowano:

- **X linii lotniczych** – lista dodanych linii lotniczych.
- **Y lotnisk** – lista lotnisk, położonych w różnych częściach świata.

Każdy lot w bazie danych zawiera następujące informacje:

- Nazwę linii lotniczej
- Numer lotu
- Lotnisko wylotu
- Lotnisko przylotu
- Dzień tygodnia (1-7)
- Godzinę wylotu

### ***Opis funkcjonalności:***

#### **1. Wyszukiwanie lotów:**

Użytkownicy mogą wpisać lotniska wylotu i przylotu oraz datę, aby zobaczyć dostępne połączenia. System przeszukuje bazę danych, filtrując wyniki na podstawie lotnisk oraz dnia tygodnia. Obliczany jest także czas trwania lotu, uwzględniając odległość geograficzną pomiędzy lotniskami.

#### **2. Panel Administratora:**

Panel admina zapewnia dostęp do operacji związanych z zarządzaniem danymi o lotach, w tym:

- a. **Dodawanie lotów:** Administrator może dodać nowe połączenie lotnicze do bazy danych, podając wszystkie wymagane informacje.
- b. **Usuwanie lotów:** Administrator może usunąć lot z bazy danych, podając numer lotu.

#### **3. Bezpieczeństwo:**

W celu zapewnienia bezpieczeństwa panelu administratora, dostęp do niego jest chroniony hasłem (domyślnie "qwerty"). Administrator musi wprowadzić poprawne hasło, aby móc zarządzać lotami w bazie.

### ***Przykład działania systemu:***

#### **1. Wyszukiwanie lotów:**

Użytkownik wchodzi na stronę główną, wybiera opcję „Wyszukiwarka lotów”, wpisuje lotnisko wylotu, lotnisko przylotu i datę wylotu, a system zwraca dostępne połączenia lotnicze.

#### **2. Panel administracyjny:**

Po kliknięciu „Strefa Admina”, użytkownik zostaje poproszony o podanie hasła. Po pomyślnym zalogowaniu, administrator ma możliwość dodawania nowych połączeń oraz usuwania istniejących.

### ***Podsumowanie:***

Projekt wykorzystuje Flask jako framework backendowy, SQLite jako bazę danych, OpenCage API do obliczania odległości i czasu lotu, a także umożliwia pełną administrację danymi lotów. Dzięki zastosowaniu nowoczesnych technologii, system jest łatwy w użyciu, a jego funkcjonalności są intuicyjne zarówno dla użytkowników, jak i administratorów. Na potrzeby projektu wygenerowano dużą ilość fałszywych danych, aby symulować działanie systemu w realistycznym środowisku.