Dokumentacja aplikacji Flask do zarządzania rezerwacjami biurek

Spis treści

1. Wprowadzenie
2. Instalacja
3. Konfiguracja
4. Struktura bazy danych
5. Endpoints
6. Uruchomienie aplikacji

Wprowadzenie

Ta aplikacja internetowa, zbudowana przy użyciu frameworku Flask, umożliwia użytkownikom rezerwowanie biurek. Użytkownicy mogą tworzyć konta, przeglądać dostępne biurka, dokonywać rezerwacji, przeglądać swoje rezerwacje oraz anulować istniejące rezerwacje.

Instalacja

1. **Klonowanie repozytorium**

   Skopiuj kod źródłowy aplikacji na swój komputer:

   ```
   git clone <URL_repozytorium>
   cd <nazwa_repozytorium>
   ```

2. **Instalacja zależności**

   Użyj `pip`, aby zainstalować wymagane pakiety:

   ```
   pip install -r requirements.txt
   ```

3. **Konfiguracja bazy danych**

   Aplikacja używa SQLite jako bazy danych. Utworzenie bazy danych następuje automatycznie podczas uruchamiania aplikacji.

Konfiguracja

W pliku głównym aplikacji (`app.py`) znajdują się podstawowe ustawienia aplikacji:

- `app.secret_key = 'your_secret_key'` - klucz sesji używany do zabezpieczenia danych sesji.
- `app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///users.db'` - konfiguracja połączenia z bazą danych SQLite.

Struktura bazy danych

Aplikacja używa SQLAlchemy do zarządzania bazą danych. Zdefiniowane są dwie tabele:

1. **User** - tabela przechowująca informacje o użytkownikach:
   - `id` - unikalny identyfikator użytkownika.
   - `username` - nazwa użytkownika.
   - `email` - email użytkownika.
2. **Reservation** - tabela przechowująca informacje o rezerwacjach:
   - `id` - unikalny identyfikator rezerwacji.
   - `desk_id` - identyfikator biurka.
   - `user_id` - identyfikator użytkownika.
   - `start_time` - czas rozpoczęcia rezerwacji.
   - `end_time` - czas zakończenia rezerwacji.

Endpoints

1. Strona główna

- **Endpoint:** `/`
- **Metody:** `GET`, `POST`
- **Opis:**
  - `GET` - wyświetla listę użytkowników.
  - `POST` - rejestruje nowego użytkownika lub loguje istniejącego.

2. Strona rezerwacji

- **Endpoint:** `/booking`
- **Metody:** `GET`
- **Opis:** Wyświetla stronę rezerwacji dla zalogowanego użytkownika.

3. Wylogowanie

- **Endpoint:** `/logoff`
- **Metody:** `GET`
- **Opis:** Wylogowuje użytkownika i przekierowuje na stronę główną.

4. Lista biurek

- **Endpoint:** `/desks`
- **Metody:** `GET`
- **Opis:** Zwraca listę dostępnych biurek w formacie JSON.

5. Rezerwacje dla biurka

- **Endpoint:** `/reservations/<int:desk_id>`
- **Metody:** `GET`
- **Opis:** Zwraca listę rezerwacji dla określonego biurka.

6. Rezerwacje dla biurka w określonym miesiącu

- **Endpoint:** `/reservations/<int:desk_id>/month/<int:year>/<int:month>`
- **Metody:** `GET`
- **Opis:** Zwraca status rezerwacji dla każdego dnia w określonym miesiącu dla danego biurka.

7. Rezerwacja biurka

- **Endpoint:** `/reserve_desk`
- **Metody:** `POST`
- **Opis:** Dokonuje rezerwacji biurka dla zalogowanego użytkownika. Generuje kod QR z informacjami o rezerwacji.

8. Informacje o rezerwacji

- **Endpoint:** `/reservation_info`
- **Metody:** `GET`
- **Opis:** Wyświetla informacje o ostatniej dokonanej rezerwacji, w tym kod QR.

9. Rezerwacje użytkownika

- **Endpoint:** `/user_reservations`
- **Metody:** `GET`
- **Opis:** Wyświetla listę rezerwacji dla zalogowanego użytkownika.

10. Anulowanie rezerwacji

- **Endpoint:** `/unreserve_desk`
- **Metody:** `POST`
- **Opis:** Anuluje rezerwację biurka dla zalogowanego użytkownika.

Uruchomienie aplikacji

1. Ustaw poziom logowania:

   ```
   logging.basicConfig(level=logging.DEBUG)
   ```

2. Utwórz tabele w bazie danych (wykonuje się automatycznie przy uruchomieniu):

   ```
   with app.app_context():
       db.create_all()
   ```

3. Uruchom serwer aplikacji:

   ```
   app.run(debug=True)
   ```

Po uruchomieniu aplikacji, dostęp do niej można uzyskać w przeglądarce internetowej pod adresem `http://127.0.0.1:5000/`.