

Admin Panel

Aplikacja "Admin Panel" stanowi kompleksowe narzędzie do zarządzania wieloma witrynami internetowymi skupionymi pod marką DMD. Została stworzona w oparciu o framework Flask, co pozwala na łatwe dostosowywanie i obsługę. Aplikacja oferuje intuicyjny interfejs użytkownika, umożliwiający administratorom efektywne zarządzanie różnymi aspektami stron. Oto kluczowe funkcje dostępne w panelu:

1. Strona Główna:

- Monitorowanie podstawowych statystyk dotyczących witryny.
- Dostęp do aktualności i informacji związanych z panelem administracyjnym.

2. Zarządzanie Użytkownikami (kadra):

- Dodawanie, edycja i usuwanie kont użytkowników (pracowników).
- Zarządzanie poziomami uprawnień i rolami użytkowników.

3. Zarządzanie Blogiem:

- Dodawanie nowych wpisów bloga.
- Edycja i usuwanie istniejących wpisów.
- Moderowanie komentarzy.

4. Zarządzanie Newsletterem:

- Zarządzanie wysyłką i harmonogramem newsletterów.

5. Zarządzanie Subskrybentami:

- Przeglądanie i zarządzanie listą subskrybentów.
- Analiza danych subskrybentów i ich preferencji.
- Moderowanie komentarzy subskrybentów.

6. Ustawienia:

- Dostosowywanie ustawień aplikacji, takich jak konfiguracja SMTP, powiadomienia e-mail, itp.
- Zarządzanie ustawieniami dostępu i bezpieczeństwa.

Dodatkowo, aplikacja posiada responsywny interfejs, umożliwiający łatwy dostęp do funkcji zarządzania treścią z różnych urządzeń. Struktura katalogów aplikacji jest przejrzysta, co ułatwia utrzymanie kodu i rozwijanie funkcjonalności. Pliki i katalogi są logicznie podzielone, co ułatwia zrozumienie struktury projektu.

Ważnym elementem są również pliki w katalogu `app`, takie jak `models.py`, które zawierają definicje modeli danych, oraz moduły w katalogu `routes`, które obsługują różne części funkcjonalności aplikacji. Dodatkowe szablony HTML i pliki statyczne zostały rozmieszczone w odpowiednich katalogach, co przyczynia się do uporządkowania projektu.

Całość aplikacji "Admin Panel" została zaprojektowana w sposób, który ułatwia zarówno administrację witrynami internetowymi, jak i rozwijanie dodatkowych funkcji zgodnie z potrzebami użytkowników.

Struktura katalogów aplikacji.

`admin_panel/`

```
-- static/
| |-- css/
| | |-- style.css
| |-- js/
| | |-- script.js
-- templates/
| |-- home.html
| |-- user_management.html
| |-- blog_management.html
| |-- newsletter_management.html
| |-- subscriber_management.html
| |-- settings.html
-- app/
| |-- __init__.py
| |-- models.py
| |-- routes/
| | |-- __init__.py
| | |-- home_routes.py
| | |-- user_routes.py
| | |-- blog_routes.py
| | |-- newsletter_routes.py
| | |-- subscriber_routes.py
| | |-- settings_routes.py
| |-- templates/
| | |-- email_templates/
| | | |-- newsletter_template.html
| | | |-- subscriber_comment_template.html
| | | |-- admin_notification_template.html
| |-- static/
| | |-- admin_panel_logo.png
| |-- utils/
| | |-- __init__.py
| | |-- authentication.py
| | |-- email.py
| | |-- permissions.py
-- config.py
-- requirements.txt
-- run.py
```

Opis plików i katalogów:

- `static/`: Katalog zawierający pliki statyczne, takie jak arkusze stylów CSS (`css/`) i skrypty JavaScript

(js/).

- `templates/`: Katalog zawierający szablony HTML dla różnych stron aplikacji.
- `app/`: Katalog zawierający logikę aplikacji.

- `__init__.py`: Plik inicjalizacyjny dla modułu.
- `models.py`: Definicje modeli danych, takich jak modele użytkowników, wpisów bloga itp.
- `routes/`: Katalog zawierający moduły obsługujące różne części funkcjonalności.

- `__init__.py`: Plik inicjalizacyjny dla modułu tras.
- `home_routes.py`: Trasy związane z główną stroną.
- `user_routes.py`: Trasy związane z zarządzaniem użytkownikami.
- `blog_routes.py`: Trasy związane z zarządzaniem blogiem.
- `newsletter_routes.py`: Trasy związane z zarządzaniem newsletterem.
- `subscriber_routes.py`: Trasy związane z zarządzaniem subskrybentami.
- `settings_routes.py`: Trasy związane z ustawieniami aplikacji.

- `templates/`: Katalog zawierający dodatkowe szablony HTML specyficzne dla aplikacji.

- `email_templates/`: Katalog zawierający szablony e-mail dla newslettera i powiadomień.

- `static/`: Katalog zawierający dodatkowe pliki statyczne specyficzne dla aplikacji.

- `utils/`: Katalog zawierający pomocnicze funkcje i moduły.

- `__init__.py`: Plik inicjalizacyjny dla modułu utils.
- `authentication.py`: Funkcje związane z uwierzytelnianiem i autoryzacją.
- `email.py`: Funkcje związane z wysyłaniem e-maili.
- `permissions.py`: Funkcje związane z kontrolą uprawnień użytkowników.

- `config.py`: Plik zawierający konfigurację aplikacji, taką jak klucze tajne, ustawienia bazy danych itp.
- `requirements.txt`: Plik zawierający zależności aplikacji, które można zainstalować przy użyciu `pip install -r requirements.txt`.
- `run.py`: Plik do uruchamiania aplikacji.