

Projekt “Life is short”

Opis aplikacji:

Projekt “Life is short” to aplikacja internetowa służąca do wsparcia zarządzania danymi przychodni lekarskiej. Back-end aplikacji został napisany w języku python wykorzystując obficie framework django wraz systemem baz danych sqlite. Do stworzenia warstwy wizualnej użyto biblioteki bootstrap.

Wymagania funkcjonalne:

Aplikacja powinna posiadać trzy różne poziomy dostępów opisane poniżej, każdy charakteryzuje się innym zakresem funkcjonalności. Ponadto aplikacja musi dostarczać menu logowania oraz ograniczać dostęp danych tylko do uprawnionych użytkowników

1. Administrator:
 - Dodawanie i usuwanie użytkowników aplikacji.
 - Zarządzanie hasłami innych użytkowników aplikacji.
 - Operowanie na danych dotyczących pracowników zatrudnionych w przychodni.
 - Operowanie na danych dotyczących zabiegów dostępnych w ofercie przychodni.
2. Recepcjonista:
 - Dodawanie i usuwanie pacjentów do systemu.
 - Operowanie na danych dotyczących pacjentów.
 - Umawianie pacjentów na zabiegi, operowanie na danych dotyczących zabiegów.
 - Podgląd do wyników zabiegów, możliwość usunięcia ich z bazy danych.
3. Lekarz:
 - Dostęp do informacji na temat przypisanych do niego wizyt.
 - Zapisywanie wyników przeprowadzonych zabiegów.

Uruchomienie na systemie linux:

Cały kod aplikacji razem z zależnymi bibliotekami znajduje się na repozytorium. Aby uruchomić stronę startową aplikacji należy wykonać poniższe instrukcje:

1. Pobierz projekt: **git clone <https://github.com/wojweb/clinic.git>**.
2. Wejdź w pobrany folder: **cd clinic/**.
3. Uruchom środowisko wirtualne: **source .venv/bin/activate**.
4. Uruchom lokalny serwer: **python mysite/manage.py runserver**.
5. Następnie w przeglądarce wejdź na adres: <http://127.0.0.1:8000/clinic/>.

Co się udało zrealizować:

Po uruchomieniu wita nas panel logowania do aplikacji dla pracowników. Warto sprawdzić, że po podaniu nieprawidłowych danych wyświetla informacje o błędzie. Zaczniemy od zalogowania się jako administrator. Aby to zrobić wpisujemy: login “admin” i hasło “admin”.

W panelu administratora widzimy tablice zawierającą wszystkich pracowników przychodni/ użytkowników aplikacji. Możemy tutaj dodać nowego pracownika do systemu, albo wyświetlić szczegółowe dane już istniejących pracowników. Aby to zrobić klikamy myszką na odpowiednim wierszu tabeli. Wyświetlą się wtedy podstawowe dane kontaktowe pracownika jak również zajmowane stanowisko. W widoku danych o pracownikach są również dostępne trzy przyciski. Pierwszy aby zaktualizować dane, drugi aby usunąć pracownika z systemu i trzeci aby zmienić hasło.

Każdy z nich działa intuicyjnie. Jeżeli chcemy wrócić do tabeli pracowników klikamy przycisk "Employees" na górze strony. W widoku administratora możemy również zarządzać zabiegami oferowanymi w przychodni. Po kliknięciu w przycisk "Treatments" na górze strony wyświetla się analogiczna tabela z zabiegami. Podobnie możemy tutaj dodać zabieg do systemu lub wyświetlić szczegóły zabiegów w systemie poprzez kliknięcie odpowiedniego wiersza tabeli. Wylogowujemy się z panelu administratora klikając przycisk "Log out" na górze strony.

Wejdźmy teraz w panel recepcjonistki, logując się na konto naszej jedynej recepcjonistki Basi używając danych: login "bacha", hasło "b". Po zalogowaniu wyświetli się panel pozwalający znaleźć pacjenta po peselu, albo jeżeli wyszukiwaniu zwróci informacje o braku pacjenta w bazie, dodać go jako nowego pacjenta. Poszukajmy pacjenta o moim peselu "97011604334". Powinien wyświetlić się ekran zawierający szczegółowe dane na temat pacjenta jak również informacje na temat przyszłych jak i przeszłych zabiegów. Możemy zaktualizować dane pacjenta, usunąć go z bazy danych jak również umówić pacjenta na nowy zabieg. Warto zauważyć, że przy umawianiu nowego zabiegu system sprawdza dostępność lekarza do którego próbujemy się umówić. Podobnie jak w panelu administratora przez wciśnięcie wiersza tabeli możemy zobaczyć szczegóły umówionej wizyty. Wychodzimy z panelu pacjenta klikając przycisk "Patients", lub wylogowując recepcjonistkę. Zachęcam do dodawania nowego pacjenta do bazy danych, a następnie umówienie go na zabieg w dniu dzisiejszym do doktor Kasi.

Ostatni panel w aplikacji to panel lekarza. Zalogujmy się na konto doktor Kasi za pomocą danych: login "kaska", hasło "k". Po zalogowaniu wyświetli się lista zabiegów doktora umówionych na dzień dzisiejszy. To co musi doktor zrobić w pracy to podczas zabiegu kliknąć odpowiedni wiersz listy, a następnie uzupełnić dane o wyniku przeprowadzonego zabiegu. Po zakończeniu pracy doktor może się wylogować. Jeżeli był zaplanowany zabieg na dzień dzisiejszy to warto uzupełnić jego wynik, a następnie zalogować się ponownie jako recepcjonisty i zobaczyć jak zabieg przeszedł z listy zaplanowanych do listy wykonanych.

Kwestie bezpieczeństwa:

- Na wstępie trzeba zastrzec, że uruchomiona aplikacja korzysta z jawnego protokołu http. Aby korzystać z zaszyfrowanego https trzeba wykupić domenę wraz z certyfikatem bezpieczeństwa czym się po prostu nie zajmowałem.
- Kwestia logowania i przechowywania haseł użytkowników jest rozwiązana przy użyciu systemu autoryzacji django, który domyślnie korzysta z funkcji PBKDF2 do generowania 256 bitowych hashy. Oczywiście użytkownicy w systemie mają ustawione tak proste hasła tylko na potrzeby prezentacji.
- Wszystkie operacje na bazach danych są wykonywane za pomocą funkcji systemu django uniemożliwiających ataki typu sql injection.
- W celu poprawienia ochrony danych zostało zaimplementowane rozwiązanie, które dla niezalogowanego użytkownika wyświetla strony do których nie ma dostępu, w ten sam sposób co strony nieistniejące.

Diagram bazy danych:

