

Programowanie Aplikacji Webowych

laboratorium 10

Cel zajęć:

Celem laboratorium jest rozpoczęcie prac nad częścią frontendową mini-projektu będącego podsumowaniem całego semestru pracy nad technologiami webowymi. Projekt będzie sukcesywnie rozwijany przez najbliższe 3 lab.

Tematem rozwijanej aplikacji będzie **aplikacja do zarządzania terminami konsultacji lekarskich**.

Aplikacja będzie miała dwa typy użytkowników: lekarzy oraz pacjentów. Lekarz będzie miał możliwość definiowania harmonogramu planowanych terminów konsultacji on-line. Możliwe jest stworzenie planu cyklicznych (powtarzalnych slotów czasowych) lub planów jednorazowych. W oparciu o upublicznione terminy konsultacji pacjenci samodzielnie rezerwują sobie dogodnie dla nich terminy konsultacji. Harmonogramy konsultacji mają być wyświetlane w postaci macierzowej z możliwością zarówno wertykalnego (sprawdzanie całego dnia) jak i horyzontalnego (sprawdzanie poszczególnych dni tygodnia) przeglądania planu lekarza.

Pacjent będzie miał możliwość przeglądania planów lekarzy wybranej specjalizacji i wyboru interesującego go terminu konsultacji. Pacjent po rezerwacji interesujących go terminów konsultacji musi jeszcze zatwierdzić wybrane rezerwacje. Dokonuje tego na ekranie finalizującym etap rezerwacji konsultacji (symulacja płatności oraz potwierdzania wyboru).

Dzisiejsze lab to rozpoczęcie prac nad funkcjonalnością frontendową. Jako zewnętrzne źródło danych proszę używać lokalnego pliku JSON. W kolejnym lab – lab 11 przejdziemy do implementacji części backendowej. Kolejne lab - lab 12 to implementacji mechanizmów autentykacji i autoryzacji. Backend i autentykacja oparte będą na wykorzystaniu Firebase lub poprzez samodzielnie implementowanym serwerze.

Rozwój aplikacji będzie sterowany kolejnymi zadaniami, których celem jest jej ewolucyjny rozwój.

Zadanie 1. Kalendarz lekarza (3 pkt)

Prace rozpoczniemy od stworzenia komponentów pozwalających na wyświetlanie i przeglądanie kalendarza dostępności lekarza. Wzorować się tutaj można na kalendarzach Googla lub temu podobnych. Domyślnym widokiem kalendarza jest widok tygodniowy. Pojedynczy slot czasowy niech będzie 0.5 h – oznacza to że minimalny czas na konsultację wynosi 0.5 h. Oczywiście będzie możliwe rezerwowanie większej ilości slotów czasowych przez

pacjenta. W widoku głównym jest możliwość przeglądania kalendarza zarówno w przyszłości jak i przeszłość. Nagłówek każdej kolumny wyświetla informacje o dniu tygodnia, dacie oraz ilości konsultacji zarezerwowanych danego dnia. Nagłówek wiersza wyświetla informacje o godzinie. Oczywiście jest że nie możliwym od razu wyświetlić 24h – niech widok domyślnie wyświetla harmonogram konsultacji dla max 6h. Sloty czasowe zarezerwowane przez klientów na harmonogramie głównym wyświetlane są w postaci bloków oznaczonych kolorami. Kolor oznaczać będzie typ konsultacji. Po najechniu na dany blok wyświetla się informacja o szczegółach wizyty. Wizyty które już się odbyły wyświetlane są jako wyszarzane. Aktualny dzień (cała kolumna) powinna być w jakiś sposób wyróżniona. Podobnie aktualny czas – powinien być symulowany przez znacznik wskazujący w którym slotcie czasowym aktualnie jesteśmy.

W kalendarzu lekarza widoczne będą terminy wolnych konsultacji oraz konsultacje już zarezerwowane. Po najechniu na konsultację zarezerwowaną przez pacjenta lekarz widzi szczegóły konsultacji (dane pacjenta, typ wizyty oraz ewentualne dokumenty dodane do podczas rezerwacji wizyty)

Zadanie 2. Definiowanie dostępności (2pkt)

Zaimplementuj opcje definiowania przez lekarza swoje dostępności. Możliwe jest definiowania cykliczne – gdzie podawany jest przedział czasowy (od kiedy do kiedy i w jakich godzinach konsultuje). Lub jednorazowe dla konkretnego dnia. Przykład dla definiowania cyklicznego - definiuje od 1.02.2025 do 12. 03. 2025 . Potem maskę dni konsultacji np. (poniedziałki, wtorki, czwartki i soboty) a następnie czasy konsultacji (i tutaj podaje się maskę konsultacji np. od 8 do 12.30 i od 16 do 21.30).

Zadanie 3. Dodawanie absencji (2 pkt)

Lekarz może również zdefiniować terminy nieobecności (np. terminy planowych wakacji lub inne planowane nieobecności). Nieobecności są całodienne, Proszę o uwzględnienie podanych planowanych absencji do identyfikacji konfliktów z już zaplanowanymi konsultacjami lub jako element walidacyjny z zadaniem 2 (przy definiowaniu dostępności). Dni nieobecności powinny być zaznaczane na czerwono w widoku głównym harmonogramu pacjenta. W przypadku identyfikacji konfliktu z zarezerwowanymi konsultacjami zmienić status konsultacji na odwołane i powiadomić o tym pacjenta

Zadanie 4. Rezerwacja konsultacji (2 pkt)

W aplikacji istnieje możliwość zarezerwowania konsultacji. W tym celu po wybraniu opcji rezerwacja konsultacji na ekranie harmonogramu zaznaczymy wolny slot czasowy. Powinien wyświetlić się formularz do definiowania szczegółów konsultacji. Potrzebne dane:

- Długość konsultacji (pamiętaj że wolne sloty czasowe muszą być bezpośrednio obok siebie), w przypadku detekcji konfliktu z kolejną konsultacją informacja o problemie oraz brak możliwości rezerwacji konsultacji.

- typ konsultacji (np. pierwsza wizyta, wizyta kontrolna, choroba przewlekła, recepta itp.).
- imię i nazwisko pacjenta
- płeć pacjenta
- wiek pacjenta
- informacje dla lekarza
- możliwość dołączania dokumentów zewnętrznych (np. wyniki badań).

Zadanie 5 Koszyk (1 pkt)

Komponent wyświetlający listę zarezerwowanych konsultacji. Zawiera funkcjonalność symulującą proces płatności.

Zadanie 6. Odwołanie rezerwacji konsultacji (1pkt)

Powinna być możliwość odwołania rezerwacji. Po zaznaczeniu wybranej konsultacji na harmonogramie powinna być możliwość odwołania konsultacji. W jej wyniku zreferowane jako konsultacje sloty czasowe stają się ponownie wolne.

Zadanie 7. Implementacja lokalnego pliku json (2pkt)

Dane w tym lab powinny być trwale przechowywane w lokalnym pliku json. Zaprojektuj jego strukturę tak aby można było go użyć jako lokalny magazyn danych dla funkcjonalności lab 10. Obsługę operacji na pliku zaimplementuj w dedykowanym temu pliku typu np. consultationsServices.ts