Wprowadzenie do aplikacji Internetowych

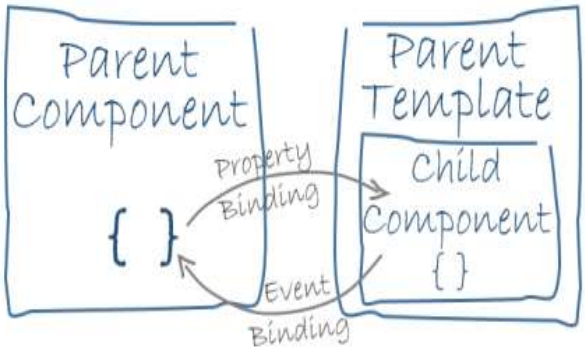
laboratorium 4

Cel zajęć:

Zapoznanie z frameworkiem Angular oraz pojęciem komponentu. Tematem przewodnim dzisiejszego laboratorium jest komponent zarówno pojedynczy jak i komunikacja pomiędzy komponentami.

Wyróżniamy 3 różne przypadki:

- komunikacje rodzic –dziecko
- komunikacje dziecko – rodzic
- komunikacje pomiędzy niepowiązanymi komponentami (realizowane za pomocą serwisów).



Cześć zadań dotyczyć będzie jednego tematu przewodniego rozwijanego przez najbliższe 3 lab. Tematem rozwijanej aplikacji będzie Restauracja Online. Model Biznesowy Restauracji polega na udostępnianiu prze aplikację oferty produktowej (gotowych dań). Użytkownicy będą mogli wstępnie zamawiać a następnie kupować dania dostępne na stronie Restauracji. Aplikacji powinna możliwość przeglądnięcia oferty dań (pod kątem różnego rodzaju kryteriów) dla wszystkich użytkowników. Dodatkowo będzie możliwość przeglądania zawartości i zamawiania posiłków przez aplikacji jako osoba zalogowana – realizacja rejestracji i logowania. Tylko admin będzie miał możliwość dodawania, usunięcia i edycji pozycji menu. Aplikacja powinna pozwalać na ocenę poszczególnych posiłków oraz zostawiania komentarzy pod poszczególnymi potrawami.

Będą istniały warianty realizacji aplikacji:

W wersji łatwiejszej - możliwość oceniania dania przez wszystkich użytkowników

W wersji trudniejszej tylko przez osoby, które kupiły dania + możliwość zastawiania komentarzy.

Jeśli chodzi o architekturę rozwiązania to przewiduje realizacje projektu w dwóch wersjach:

W wersji trudniejszej - możliwość samodzielnego napisania serwera REST API (ExpressJS) w oparciu o materiały z wykładu i ewentualne „lekkie" podpowiedzi.

W wersja łatwiejsza – backend w całości oparty na Firebase.

Aplikacja będzie realizowana etapami:

Lab 4 – przygotowanie funkcjonalności Frontendu dostępnej dla wszystkich

Lab 5 - obsługa backendu – serwer aplikacyjny + persystencja danych w bazie danych

Lab 6 - wprowadzenie autentykacji (logowanie) oraz obostrzenie dostępu do wybranych funkcji systemu tylko dla osób zalogowanych o odpowiednich preferencjach (np. admin).

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

Zadania na rozgrzewkę (bez oceniania).

Zadanie 1. Weryfikacja środowiska pracy

Sprawdź czy masz na swoim komputerze zainstalowane następujące oprogramowanie:

Npm: npm –v

Zapoznaj się z krótkim tutorialem dotyczącym instalacji i używania Angulara

https://angular.io/guide/quickstart. Na jego podstawie sprawdź czy masz zainstalowanego Angulara.

ng version lub ng -v

W przypadku gdyby nie było zainstalowanego angulara proszę o jego instalacje. Zapoznaj się z możliwościami środowiska CLI - https://cli.angular.io/

Zadanie 2. Podstawowym językiem tworzenia oprogramowania w Angularze jest

TypeScript. Przeglądnij tutorial dotyczący TypeScript

https://www.typescriptlang.org/docs/tutorial.html

Zapoznając się z najważniejszymi zasadami i konstrukcjami języka.

Zadanie 3. Stwórz swój pierwszy projekt w Angularze. Zapoznaj się ze strukturą projektu. Stwórz swój pierwszy komponent w Angularze wypisujący informacje o ulubionym Aktorze/Aktorce. Dane aktora: imię, Nazwisko oraz Tytuł ulubionego filmu podaj za pomocą pola/pól typu input.

Wykorzystując interpolacje zweryfikuj wynik działania następujących operacji:

{{2 +2}}

{{ imie.length}}

{{ imie.toUpperCase()}}

{{ a = 2+3}}

{{ windows.location.href}}

Wyciągnij wnioski z otrzymanych wyników i ewentualnie tak zmodyfikuj kod aby zadziałał.

Zadanie 4. Napisać funkcję tabliczka() przyjmującą dwie tablice (typu string i number), która wyświetli tabliczkę „złożenia" z zawartości dwóch tablic. Przykładowo dla

tablic: ['Ala', 'ma', 'kota'] oraz [0, 1, 2, 3] otrzymamy:

Ala0 Ala1 Ala2 Ala3

ma0 ma1 ma2 ma3

kota0 kota1 kota2 kota3

Wykorzystać dwa różne rodzaje pętli z TypeScript:

https://www.typescriptlang.org/docs/handbook/iterators-and-generators.html

Zadania punktowane. (20 pkt)

Zadanie 5. (2 pkt)

Wybór samochodu. Model danych składa się z kolekcji samochodów. Każdy samochód to marka, model oraz lista dostępnych kolorów. W kolekcji może być kilka modeli tej samej marki. Stwórz aplikacje pozwalająca na wybór pojazdu. Najpierw wybierasz z listy markę, która Cię interesuje. Na tej podstawie druga lista rozwijana wyświetla tylko listę modeli, które są dostępne dla wybranej marki. Wybierz teraz z tej listy model. Dla wybranego pojazdu niech wyświetli się lista dostępnych kolorów – w postaci np. kwadratów o odpowiednim kolorze. Wybór któregokolwiek powinien wyświetlić pełną informacje o wybranym samochodzie tzn. Model, marka, kolor.

Zadanie 6. (1pkt)

Stwórz dwa komponenty. Niech jeden będzie rodzicem drugiego. Komponent dziecko zawiera 2 przyciski typu buton: click oraz reset. Niech komponent rodzic wyświetla informacje o ilości kliknięć na przycisku clik. Gdy licznik pokaże wartość 10 nich rodzic zablokuje możliwość klikania w ten przycisk przez dziecko. Naciśniecie przycisku reset w sytuacji gdy clik jest nieaktywny resetuje wartość licznika i aktywuje przycisk clik (powrót do stanu pierwotnego).

Zadanie 7. (1pkt)

Stwórz aplikacje o funkcjonalności prostego kalkulatora. Kalkulator.

Zadanie 8. Restauracja - Lista dań (6 pkt)

Stwórz nowy projekt a w nim nowy komponent/komponenty reprezentujące Restauracje a w niej dania oferowane przez Restauracje. Zmodyfikuj tak kod aby nowy komponent był komponentem wyświetlanym na starcie naszej aplikacji. Komponent Dishes powinien wyświetlać listę dań. Lista dań powinna być zdefiniowana w pliku projektu zwierającym tylko fake data lub w zewnętrznym pliku. Pojedynczy obiekt typu Dish powinien zawierać następujące pola: Nazwa dania, typ kuchni do której należy danie (np. włoska, polska, indyjska, międzynarodowa, francuska itp.), typ i kategoria posiłku (wegański, mięsny, zupa, dania główne, sałatki, przystawka, kolacja, śniadanie itp.), lista składników, max ilość dań możliwych do wydania danego dnia, cena dania, krótki opis dania oraz link do poglądowych zdjęć.

Stwórz przynajmniej 12 obiektów i użyj ich w komponencie.

Wyświetl zawartość tablicy obiektów w szablonie komponentu głównego - dyrektywa *ngFor (każda element odpowiednio wystyluj). Na liście dań proszę wyświetlić tylko jedno zdjecie dania. Reszta zdjęć zostanie użyta później w formie np. slajdera zdjeć prezentującego szczegóły dania. Przy każdym produkcie powinny znajdować się 2 przyciski + i - pozwalające na zamówienie dania (+) lub rezygnacje z zamówionego dania (przycisk -).

Jeśli ilość dań znajdujących się w tablicy będzie wynosiła 0 to należy wyświetlić inny komunikat przy daniu niż gdy ilość dostępnych dań jest większa od 0. W przypadku gdy ilość dań spadnie do zera przycisk + powinien zostać ukryty. Nie możemy zamawiać przecież dania, którego restauracja nie jest w stanie wydać – dyrektywy ngStyle lub ngClass.

Podobnie przy rezygnacji z zamówienia – ilość dań nie może przekroczyć max ilości dań zaplanowanych przez restauracje.

Gdy ilość możliwych do wydania dań będzie zbliżała się do 0 (np. od 3 w dół) należy zaznaczyć to w sposób graficzny np. inne tło, kolor czcionki, wielkość fontów lub inny wizualny sposób. Podobnie należy rozróżnić dnia najdroższe oraz najtańsze – za pomocą dodatkowego obramowania obejmującego dany produkt - zielone – najdroższy, czerwone najtańszy.

Wypisz nazwę dania oraz nazwę kuchni kraj dużymi literami -> skorzystaj odpowiedniego typu pipe.

Ze względu na szalejącą inflacje szefowie restauracji zdecydowali że wszystkie ceny mają być podawane tylko w USD lub Euro. Wyświetl cenę dania wraz z nazwa (lub znakiem płatniczym) skojarzonym z walutą np. USD - $.

Wyświetl również sumaryczną ilość aktualnie zarezerwowanych dań - jeśli wynosi on więcej niż 10 ma być wyświetlana na niebieskim tle, jeśli poniżej 10 na pomarańczowym tle.

Uwaga!! Oceniać będę również zaproponowaną architekturę rozwiązania. Powinna być zwinna i elastyczna – pamiętajmy o zasadzie SOLID.

Zadanie 9. Usuwanie dania (1pkt)

Rozszerz funkcjonalność aplikacji o możliwość usuwania dania. Zrealizuj tą funkcjonalność poprzez dołożenie przycisku Usun obok dania. Naciśniecie tego przycisku powinno skutkować usunięciem dania z listy dań.

Zadanie 10. Dodawanie dania (2pkt)

Skoro jest możliwość usuwanie dania z listy, niech będzie także dostępna możliwość dodawania nowego dania. Dodawania odbywa się za pomocą formularza – sugeruje zastosowanie formularza typu Model Driven Forms. Na razie podobnie jak z usuwaniem dania dostęp do tej funkcjonalności będą mieli wszyscy użytkownicy – potem tylko z odpowiednimi uprawnieniami. Zastosuj mechanizm walidacji.

Zadanie 11. Ocena dania (1+1pkt)

Rozszerzmy funkcjonalność pojedynczego dania o możliwość oceniania atrakcyjności dania przez klienta (np. wybór ilość gwizdek lub jakaś inna interesująca forma oceniania). Na razie oceniać dania będzie mógł każdy klient. Później po wprowadzeniu autoryzacji tylko osoba która kupiła danie. Docelowo funkcjonalność ta będzie dostępna tylko z poziomu karty poszczególnego dania. Preferowane samodzielna realizacja oceny bez korzystania z gotowych bibliotek. (1pkt)

Zastanów się w jaki sposób zrealizujesz ocenę (oddzielny komponent? a może tylko atrybut komponentu Dish?)

Zadanie 12. – filtrowanie listy dań

Tworzymy dodatkowy komponent służący do filtrowania/ wyszukiwania dań pod katem wybranych kryteriów. Kryteriami po których możemy filtrować są: typ kuchni, cena (zakres), ocena, kategoria dania. Proponuje do realizacji tej funkcjonalności zastosowanie samodzielnie zdefiniowanych potoków. Sposób realizacji opisany w sekcji poniżej **(1 pkt)**

Wersja rozszerzona

Możliwość wyboru kilku wartości dla danego kryterium np. wybór kilku typów kuchni z listy dostępnych, wybór kilku kategorii produktu (np. sałatki, dania mięsne, kolacje) lub np. oceny 4 i 5 gwiazdek **(1 pkt)**

Filtry zwierają tylko wartości dostępne w liście dań - dotyczy np. zakresu cen. Nie od 0 do 100 tylko od dostępna cena minimalna do cena maksymalna – wartości powiązane z aktualnymi wynikami filtrowania. **(1 pkt)**

Kryteria filtrowania można łączyć tzw. przykładowe kryteria filtrowania: interesują mnie dania o ocenie 3 i 4 gwiazdki należące do kuchni meksykańskiej i będące daniami

wegańskimi. Wyniki filtrowania powinny być dostępne online już podczas wyboru wartości w filtrze**. (1 pkt)**

Zadanie 13 Koszyk (1 pkt)

Stwórz nowy komponent, niepowiązany z pozostałymi zawierający informacje o wybranych daniach, ich ilości oraz sumie całego zamówienia. Jej zawartość będzie powiązana z lista dań. Jeśli dodaje dania do koszyka pozycje w koszyku rosną, jeśli usuwam maleją. Usuniecie dania z listy powinno skutkować usunięciem dania z koszyka.

---------- Potoki własne – implementacja przykładowa ----------

Angular pozwala tworzyć własne potoki . Wymagane jest aby:

- uzyć dekorator @Pipe z metadanymi potoku, wśród których jest własność name. Ta wartość zostanie wykorzystana do wywołania potoku
- Implementować metodę transformacji interfejsu PipeTransform. Ta metoda pobiera z potoku wartość oraz zmienną liczbę argumentów dowolnego typu i zwracają wartość przekształconą (piped).

```
// Przykładowa implementacja potoku typu wyszukaj po podanym tekście
@Pipe({ name: 'searchPipe' })
export class SearchPipe implements PipeTransform {
    transform(courses: Course[], searchText: string): Course[] {
        if (!courses)
            return [];
        if (!searchText)
            return courses;
        searchText = searchText.toLowerCase();
          return courses.filter(course => {
              return course.name.toLowerCase().includes(searchText);
          });
    }
```

Użycie zdefiniowanego potoku w szablonie komponentu. Każdy parametr rozdzielany dwukropkami w szablonie odwzorowuje jeden argument metody w tej samej kolejności

```
<div>
<div *ngFor="let p of (getCourses() | searchPipe : search )">
        ……..
```

```
</div>
</div>
```