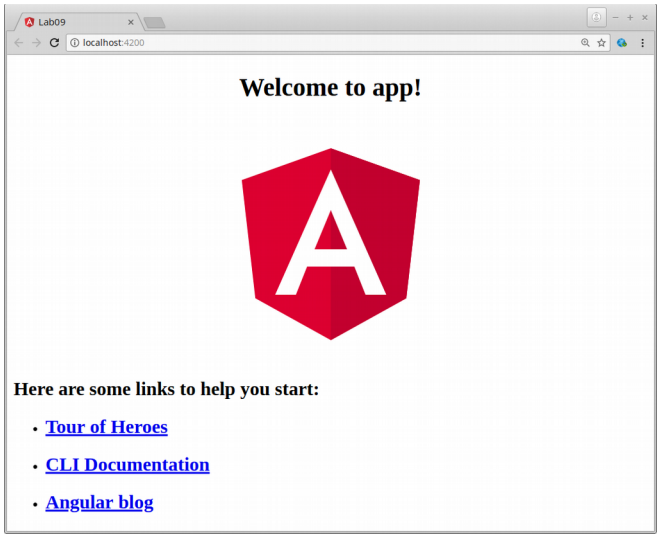
Laboratorium 9 – Angular 7

Angular 7 QuickStart – https://angular.io/guide/quickstart
Angular CLI – https://cli.angular.io/

Zadanie 0 – najnowsze wersje node i npm, instalacja Angular-CLI
Proszę sprawdzić wersje aplikacji node i npm, jeśli będzie potrzeba, to będzie trzeba zaktualizować (node powinien być w wersji 9.3 lub nowszej, npm w wersji 6.0.0 lub nowszej). Instalujemy Angular-CLI (ng) i sprawdzamy: ng version.

Może być potrzeba utworzenia linku symbolicznego do pobranego narzędzia, np.:
sudo ln -s /home/student/.../node-v9.3.0-linux-x64/lib/node_modules/@angular/cli/bin/ng /usr/bin/ng

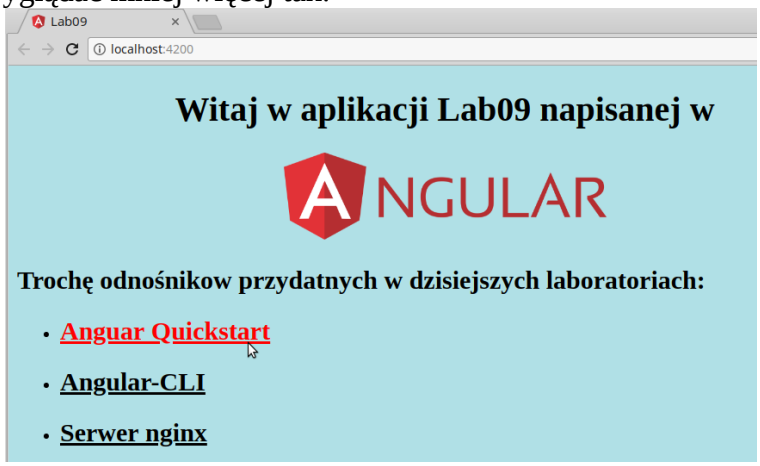
Zadanie 1 – Hello World w Angular 7
Korzysając z Angular-CLI proszę utworzyć projekt o nazwie lab09 w Angular 7 – takie Hello World. Projekt projekt proszę skompilować i uruchomić w przeglądarce, port 4200. Powinniśmy mieć angularowe „Hello World”:



Zadanie 2 – modyfikujemy nasz projekt
W utworzonym projekcie proszę wykonać serię zmian:

- zmiana loga na stronie głównej na http://mherman.org/assets/img/blog/angular-logo.png
- zmiana nazwy aplikacji na na Lab09
- zmiana tła strony na kolor powderblue - styl globalny dla całej aplikacji (styles.css)
- zmiana koloru odnośników na czarny, a po najechaniu na czerwony, w głównym komponencie aplikacji (app.component.css)

Aplikacja powinna wyglądać mniej więcej tak:



Dla chętnych (szczerze powiedziawszy, nie wiem czy ma to sens) można sobie zmienić:

- nazwę głównego komponentu aplikacji z AppComponent na MainComponent
- nazwę czterech plików komponentu z ‘app.*’ na ‘main.*’
- nazwę głównego modułu z AppModule na MainModule
- nazwę pliku głównego modułu z ‘app.*’ na ‘main.*’
- nazwę selektora z <app-root> na <main-root>

Należy oczywiście dokonać jeszcze wszystkich potrzebnych zmian, aby aplikacja działała dla komponentu, modułu i selektora przy zmienionych nazwach.

Zadanie 3 – dodatkowy komponent wewnątrz głównego komponentu

Korzystając z Angular-CLI dodajemy do aplikacji komponent LinksComponent, do którego przenosimy informacje o odnośnikach, łącznie ze stylem kolorów. Przy tworzeniu wystarczy podać słowo ‘links’ z małej litery, ‘Component’ jest dodawane automatycznie. Komponent umieszczamy wewnątrz szablonu komponentu AppComponent za pomocą selektora <app-links>.
W odnośnikach umieszczamy 4 adresy URL z nazwami jak w obiekcie poniżej:

```
let links = [
  { "url": "https://angular.io/guide/quickstart", "name": "Anguar Quickstart"},
  { "url": "https://cli.angular.io", "name": "Angular-CLI"},
  { "url": "https://httpd.apache.org/", "name": "Server-Apache"},
  { "url": "https://nginx.org", "name": "Serwer nginx"}
];
```

Może to wyglądać mniej więcej tak:

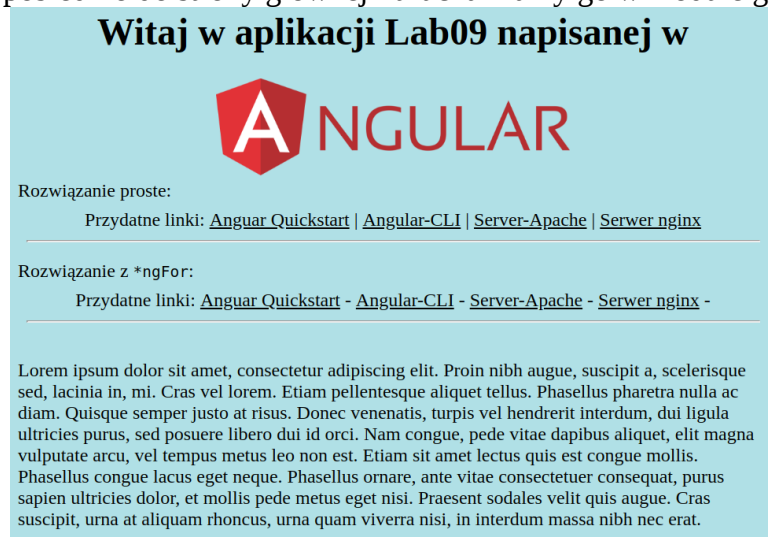
Rozwiązanie proste:

```
<a ... href="{{links[0].url}}">{{links[0].name}}</a> |
...
```

Rozwiązanie z *ngFor (tablica jako pole w klasie komponentu, pętla w szablonie):

Zmiena klasy:	Kod szablonu:	Wynik:
tab = ["Ala", "Ela", "Ola"];	<span *ngFor="let element of tab"> {{element}} ### 	Ala ### Ela ### Ola ### <span _ngcontent-c0=""> Ala ### <span _ngcontent-c0=""> Ela ### <span _ngcontent-c0=""> Ola ###

Zadanie 4 - dodatkowy komponent bezpośrednio w module głównym
Korzystając z Angular-CLI dodajemy do aplikacji komponent ContentComponent, który będzie zawierał informacje o głównej treści strony, np.: „Lorem ipsum...". Dodajemy selektor komponentu <app-content> bezpośrednio do strony głównej i uruchamiamy go w module głównym. Wynik:

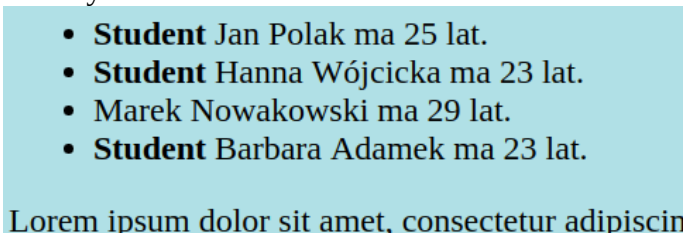


Generalnie prawie wszystko zrobi za nas Angular-CLI. Zadanie na … 60 s :D

Zadanie 5 – ngIf z listą studentów
Dodajemy komponent StudentsComponent. Korzystając z *ngFor wyświetlamy w nim listę osób:

```
const studentsTab = [
  {name: "Jan", "surname": "Polak", "age":25 },
  {name: "Hanna", "surname": "Wójcicka", "age":21 },
  {name: "Marek", "surname": "Nowakowski", "age":29 },
  {name: "Barbara", "surname": "Adamek", "age":23 }
];
```

Dodatkowo wykorzystujemy dyrektywę *ngIf do dobrania odpowiedniego opisu, tj. osoby starsze niż 26 nie uznajemy za studentów. Komponent podłączamy do ContentComponent za pomocą selektora <app-students>. Wynik:



Zadanie 6 – aplikacja Angular na statycznym serwerze plików Apache (albo np. nginx)

a) Instalacja Apache … albo czegoś innego

Przed instalacją może być potrzeba instalacji przynajmniej pakietu mintupdate (całkowita aktualizacja systemu nie jest chyba potrzebna) i aktualizacja dostępnych pakietów:

```
sudo apt-get update
```

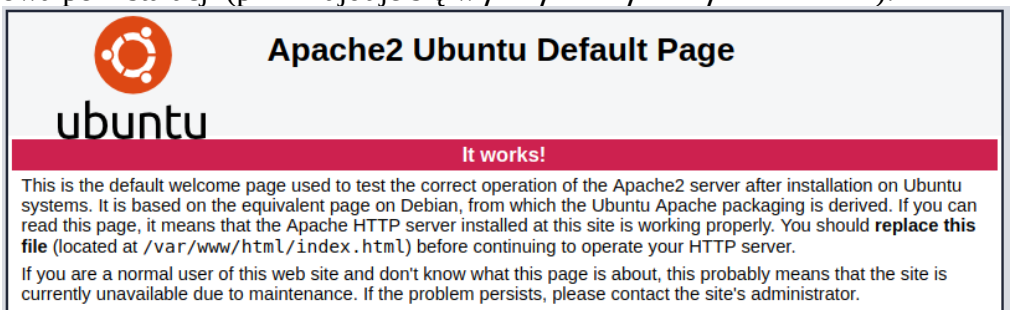
Instalujemy sobie jakiś klasyczny serwer plików, np. Apache:

```
sudo apt-get install apache2
```

Po instalacji serwer już powinien działać pod adresem http://localhost. W razie potrzeb może być potrzeba jego restartu, zatrzymania czy uruchomienia:

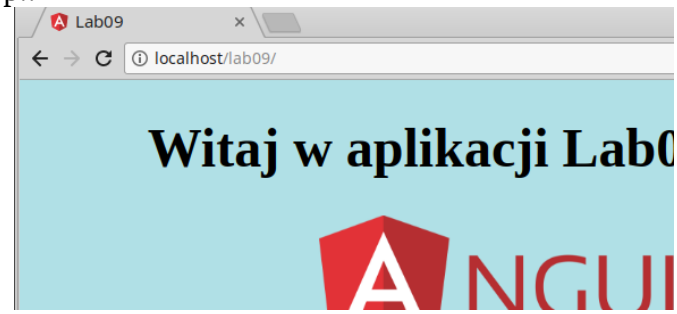
```
sudo systemctl restart apache2
sudo systemctl stop apache2
sudo systemctl start apache2
```

Strona startowa po instalacji (plik znajduje się w /var/www/html/index.html):



b) Kompilacja i wrzucenie aplikacji na serwer

Kompilujemy naszą aplikację lab09 do wersji produkcyjnej (ng build --prod) i wgrywamy całą zawartość katalogu dist/ na serwer czyli do (/var/www/html). Sprawdzamy w przeglądarce czy nasza aplikacja działa, np.:



c) Dopracowanie konfiguracji

Można sobie dopracować konfigurację dopisując do pliku /etc/apache2/apache2.conf domenę albo IP serwera: ServerName 127.0.0.1

Na potrzeby programowania i testów warto by też przenieść główny katalog serwera do katalogu domowego, np. do /home/student/DevHost. Tworzymy odpowiedni katalog w /home/student:

```
mkdir /home/student/DevHost
```

Ustawiamy odpowiednią konfigurację w /etc/apache2/apache2.conf:

```
# <Directory /var/www/>  ## linia do zastąpienia
<Directory /home/student/DevHost/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
</Directory>
```

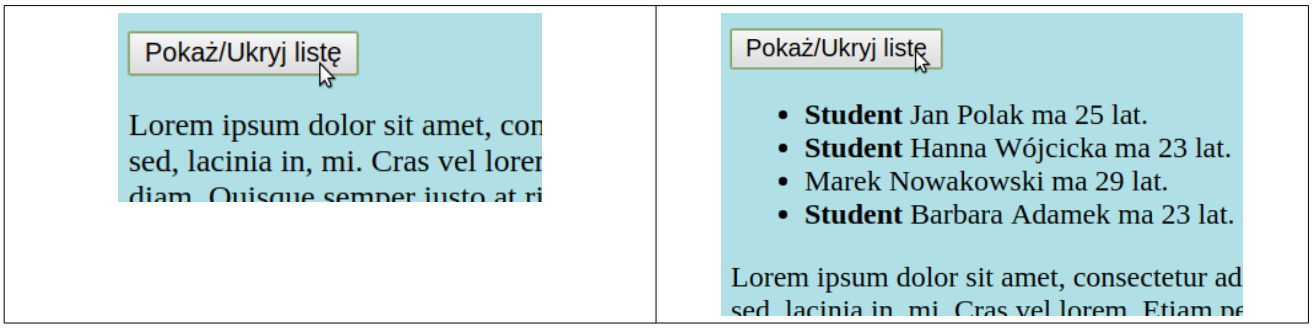
i w pliku /etc/apache2/sites-enabled/000-default.conf zamieniamy:

```
# DocumentRoot /var/www/html ## linia do zastąpnia
DocumentRoot /home/student/DevHost
```

Po takiej zmianie plik /home/student/DevHost/index.html z dowolną zawartością po restarcie serwera powinien być wyświetlany pod http://localhost. Nie trzeba mieć uprawnień root'a do kopiowania plików na serwer.

Zadanie XYZ – coś więcej dla chętnych

Można sobie do komponentu StudentsComponent dodać przycisk, który będzie wyświetlał i chował listę studentów po każdym naciśnięciu. Zmienna logiczna w klasie, np. showComponent = true, metoda w klasie zmieniająca true na false i odwrotnie, np. onClickMe() i odpowiedni wpis w szablonie, zob. https://angular.io/guide/user-input. Wynik:



Aktualizacja node i npm

Aktualizacja node i npm po pobraniu najnowszej wersji NodeJs (np. https://nodejs.org/dist/v11.14.0/node-v11.14.0-linux-x64.tar.xz) i rozpakowaniu jej do katalogu /home/student/Programy:

```
sudo mv /usr/bin/node /usr/bin/node2
## albo
## sudo unlink /usr/bin/node
sudo ln -s /home/student/Programy/node-v11.14.0-linux-x64/bin/node /usr/bin/node
node -v
npm -v
whereis npm
sudo mv /usr/bin/npm /usr/bin/npm2
## albo
## sudo unlink /usr/bin/npm
sudo ln -s /home/student/Programy/node-v11.14.0-linux-x64/bin/npm /usr/bin/npm
npm -v
sudo npm i -g @angular/cli
```