

Angular - zadanie

Zadanie polega na utworzeniu strony logowania.

Wszystkie niezbędne dane, w celu działania aplikacji, powinny być umieszczone na sztywno w aplikacji.

Punkty do zrealizowania:

1. Wykorzystać [routing](#) angulara do nawigacji po aplikacji
2. Na ścieżce /login powinien znajdować się formularze logowania
 - Formularz musi się składać z:
 - Pola Email
 - Pole Hasło
 - Przycisk Logowania
 - Formularz wykonać w [reactive forms](#)
 - Formularz powinien walidować wprowadzane dane
 - Sprawdzając czy pola wymagane są uzupełnione
 - Sprawdzając czy email jest emailem
 - Sprawdzając poprawność danych
 - Błędy powinny być pokazywane użytkownikom w czytelny sposób
3. Proces logowania, powinien być opisany w [serwisie](#)
 - Gdy proces logowania zakończy się powodzeniem powinien przekierować na stronę domową (/home)
4. Strona domowa powinna być chroniona przez ręcznym przejściem do strony
 - Jeśli użytkownik nie zalogował się to strona powinna przekierować użytkownika z powrotem do strony logowania.
5. Na stronie domowej ma znajdować się tylko przycisk do wylogowania

Dodatkowe punkty za:

- Responsywność strony (czytelna i funkcjonalna na urządzeniach mobilnych)
- Układ komponentów na stronie
- Dobrze zarządzana struktura plików w projekcie
- Wykorzystanie [lazy loading](#) przy ładowaniu strony domowej
- Wykorzystanie [guard](#) do ochrony strony domowej
- Podzielenie aplikacji na moduły
- Dowolny adres nie będący /login, /home powinien być przekierowany na stronę /login

Można wykorzystywać różne gotowe komponenty np. [Material Design](#)

Java/Kotlin - zadanie

Zadanie polega na utworzeniu, przy pomocy technologii [Spring Boot](#), prostego rest Api. Rest służyć będzie do zarządzania użytkownikami. Serwis powinien przechowywać dane w bazie danych (np SQLite).

Lista requestów do zaimplementowania:

Path: /user

Method: GET

Response body:

```
{
  "id": Long,
  "name": String,
  "surname": String
},
{
  "id": Long,
  "name": String,
  "surname": String
}, ...]
```

Request powinien czytać z bazy danych listę użytkowników.

Path: /user

Method: POST

Request body:

```
{
  "name": String,
  "surname": String
}
```

Request służy do dodawania nowego użytkownika, który ma zostać zapisany w bazie danych.

Path: /user/:id

Method: PUT

Request body:

```
{
  "name": String,
  "surname": String
}
```

Request służy do aktualizacji danych użytkownika, i podmianie jego danych w bazie danych.

Path: /user/:id

Method: DELETE

Request służy do usunięcia danych użytkownika i usunięcia go z bazy danych.