

Dawid Kruczek	299704	Group 2
Internet Engineering	16.06.2020	Applied Computer Science

## 1. Introduction

The library system app is built to be used in library. It provides system to manage books orders, adding authors, new books, new users, new categories etc. It based on my last term's backend in Java. The frontend is made in Angular. Communication using rest service to HTTP request.

## 2. Application

First you have to login in your account. User has to put personal data into the form ([img. 1](#)) and the system will check it in database. ([img. 2](#))

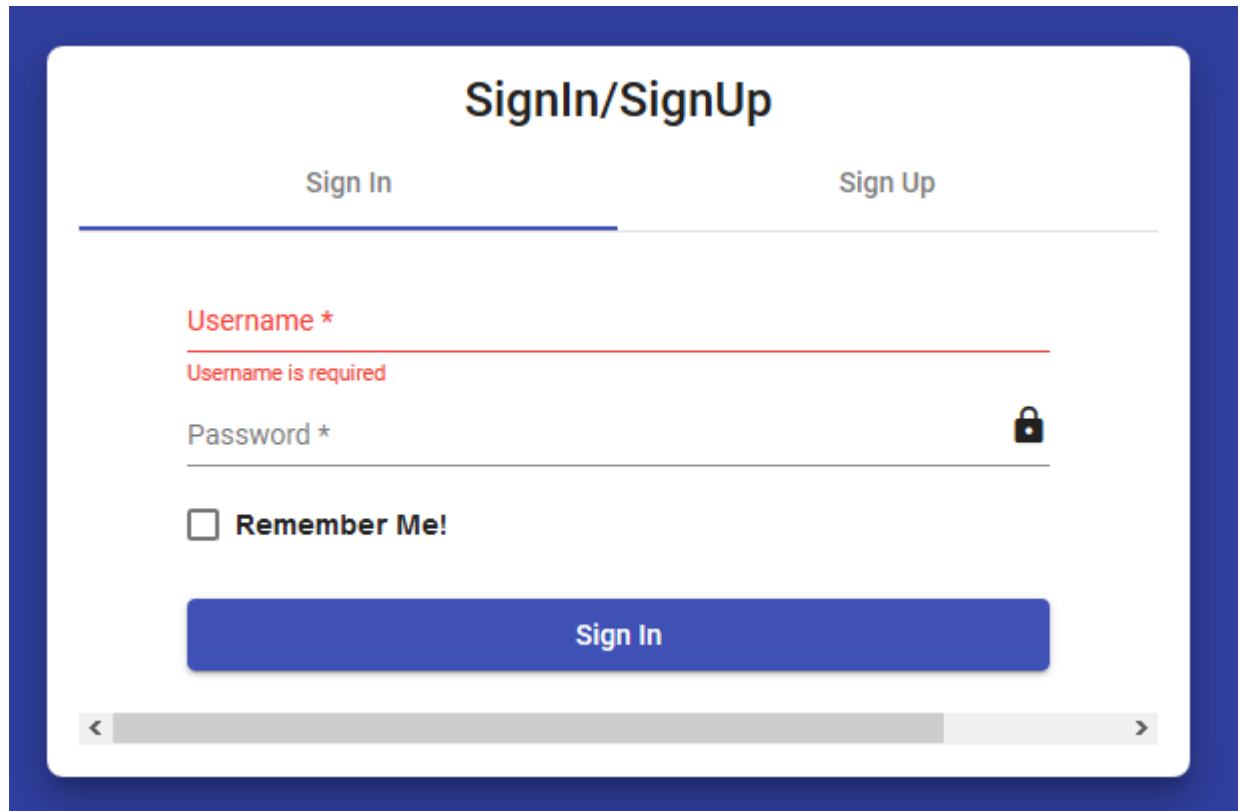


Image 1 Login form

```
login(username: string, password: string) {
  return this.http.post<User>(this.AUTHENTICATE_URL, { body: { username, password } })
    .pipe(map( project: user => {
      console.log(user);
      if (user && user.jwt) {
        localStorage.setItem('currentUser', JSON.stringify(user));
        this.currentUserSubject.next(user);
      }
      return user;
    }));
}
```

Image 2 Code to login and authentication

After that the server return JSON Web Token which is saved in the local storage. When the user logs in the site shows welcome panel ([img. 3](#)). Next user can choose one of an option on the menu.

Welcome admin!

Image 3 Welcome panel for admin

IE Project				Authors	Books	Categories	Orders	Publishers	Logout
First Name	Last Name	Date of birth	Actions						
Henryk	Sienkiewicz	05.05.1846	<button>Edit</button> <button>Delete</button>						
Olga	Tokarczuk	29.01.1962	<button>Edit</button> <button>Delete</button>						
Adam	Mickiewicz	24.12.1798	<button>Edit</button> <button>Delete</button>						
<button>Add new Author</button>									

Image 4 Authors panel

When the client click authors, he can add, edit and delete authors.

24.12.1798

Edit Author

first name

Henryk

×

last name

Sienkiewicz

×

birthday

05.05.1846

×

Save

Discard

Image 5 Popup to editing

24.12.1798

## Add Author

first name  
Test ×

last name  
Test ×

birthday  
10.06.2020 × ×

Save

Discard

Image 6 Popup to adding

Editing, deleting, adding is very similar for every menu option. For example I will show authors. Every request is done in authors service. Adding and updating request has to be JWT added to authorize the operation.

```
addAuthor(firstName: string, lastName: string, birthDate: string) {
  const httpOptions = {
    headers: new HttpHeaders({ headers: {
      'Content-Type': 'application/json',
      Authorization: '',
    }})
  };

  httpOptions.headers = httpOptions.headers.set('Authorization', localStorage.getItem( key: 'currentUser'));
  const body = {
    firstName: firstName,
    lastName: lastName,
    dateOfBirth: birthDate
  }

  console.log(httpOptions.headers);
  return this.http.post<Author>(this.ADD_AUTHOR, body, httpOptions);
}
```

Image 7 Add method

```
deleteAuthor(authorId) {
  return this.http.delete( url: this.DELETE_AUTHOR + authorId);
}
```

Image 8 Delete method

```

updateAuthor(author: Author) {
  const httpOptions = {
    headers: new HttpHeaders({ headers: {
      'Content-Type': 'application/json',
      Authorization: '',
    })
  };

  httpOptions.headers = httpOptions.headers.set('Authorization', localStorage.getItem('key: 'currentUser'));

  const body = {
    id: author.id,
    firstName: author.firstName,
    lastName: author.lastName,
    dateOfBirth: author.dateOfBirth
  }

  console.log(httpOptions.headers);
  return this.http.put<Author>({ url: this.UPDATE_AUTHOR + author.id, body, httpOptions});
}

```

Image 9 Update method