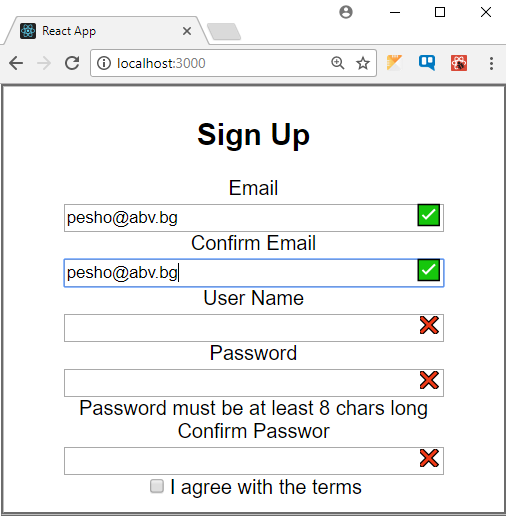
# Exercise: Events and Forms

**Create pokemon roster**

1. Implement registration logic (this part is already done)
2. Create login form
3. Implement login logic
4. Create form for adding new pokemons, it should be visible only for logged users
5. Create component that will visualize all added pokemons, it should be visible only for logged users
6. Add roster, that visualize all pokemons in the db

## Registration Form

After installing all dependencies and running both provided servers, you should land on the following page, with registration form, which is already completed.



## Login Form

### 2.1 Form Creation

By examine the registration form, create login form, that should send POST request to the following path

http://localhost:5000/auth/login

In its **body** the POST request should send the **following object**

let payload = {

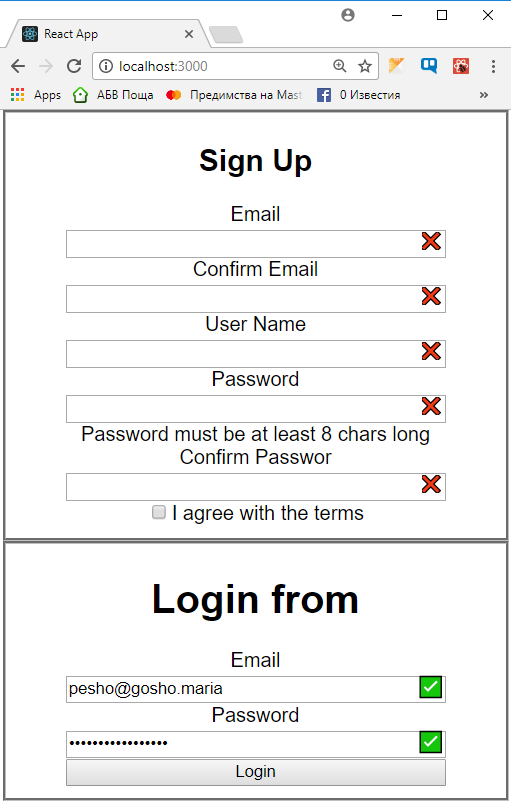
email: [Input\_From\_User],

password: [Input\_From\_User]

}

Try to add also some kind of validation, that will prevent the user from submitting form with empty fields

Upon finishing the task, you should have a view like the following image.



### 2.2 Login Logic

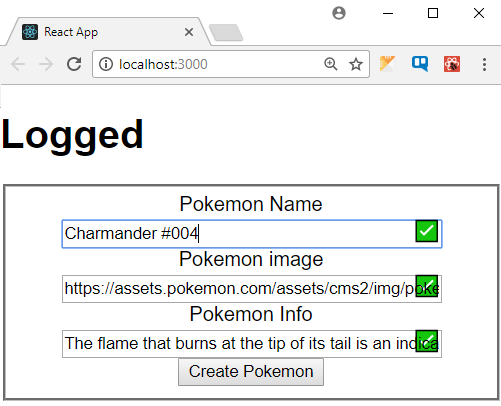
Upon receiving login request, from a user with valid credentials, the server will respond with request status, authentication token and some personal user data.

Save the token at appropriate place.

Hint: Use Local Storage to save the token, so even upon page refresh you still be logged in

## Pokemon Form

Create a form for generating new pokemons, which might look like the following image.



Upon creating new pokemon you should send new POST request to the server

<http://localhost:5000/pokedex/create>

In the post body you should have the following object

let payload = {

pokemonName: [Current\_Pokemon\_Name],

pokemonImg:[Current\_Img\_Url],

pokemonInfo: [Current\_Pokemon\_Info]

}

Upon adding new Pokemon, it should be listed below the form.

