# React.js Workshop – Pets System

Download the web server from the course, install all its dependencies and run the server. It is listening on port 5000.

## Create a React application

Create a React application and prepare the initial project structure. Install Flux and prepare its components.

## Add authentication

Add authentication and make sure the register, login and logout functionalities work correctly. To register a user, you need to send a POST request to the server on ‘/auth/signup’ with ‘name’, ‘email’ and ‘password’ data (sent as JSON). To login a user, you need to send a POST request to the server on ‘/auth/login’ with ‘name’ and ‘password’ data (sent as JSON). You need to save the user token in your application state. Make sure you validate everything on the client application.

## Add creating of pets

Add a form to create pets in the system. Each pet has ‘name’ as string, ‘age’ as number, ‘type’ (Cat, Dog or Other), ‘image’ as string URL and optional ‘breed’ as string. Make sure you validate everything on the client application. The data must be sent as POST request to the server on ‘/pets/create/’. This route is only for authenticated users so you need to send a header with `Authorization` name and value `bearer {*token*}` in order to pass the authentication checks.

## Add listing of pets

Add a page where all pets are listed. Since the data from the server comes in pages of 10 pets, your page must have buttons to navigate between the pages. Try to validate the buttons when the page is no longer valid. You need to make a GET request to ‘/pets/all’ to receive an array of pet data. Optionally, you can pass a query string parameter ‘page’ to request more data, for example ‘/pets/all?page=2’. Link each pet to its details page. Don’t show every piece of information about the pets on this page. Leave something for the details page. You may add this functionality on the home page.

## Add pet details

Add a page where all pet details are shown. You need to make a GET request to ‘/pets/details/{*id*} to retrieve information about the pet with the provided id. This route is only for authenticated users so you need to send a header with `Authorization` name and value `bearer {*token*}` in order to pass the authentication checks. Make sure your React application redirects to the login page, if the user tries to open the pet details page and she’s not logged in.

## Add option to add comments

On the pet details page add an option for the user to comment on the current pet. Each comment has ‘message’ as string data (sent as JSON). You need to make a POST request to ‘/pets/details/{*id*}/comments/create’ in order to create a comment. The id is the pet id the user is commenting on. Make sure you validate everything on the client application. This route is only for authenticated users so you need to send a header with `Authorization` name and value `bearer {*token*}` in order to pass the authentication checks.

## Add listing of all comments

On the pet details page, you need to list all comments for the current pet. You need to make a GET request to ‘/pets/details/{*id*}/comments’ to receive an array of comment data. This route is only for authenticated users so you need to send a header with `Authorization` name and value `bearer {*token*}` in order to pass the authentication checks.