## 5.Fisher Game

Use the provided skeleton and the server.



### Login User

The **Login** page contains a form for existing user authentication. By given **username** and **password,** the app should login an existing user.

* After a **successful login** the **home page should be displayed**.
* In case of **error**, an appropriate error **message** should be displayed and the user should be able to fill in the login form again.
* Keep the user data in the browser's **session or locale storage**.
* Get request: **http://localhost:3030/users/login**



If the user is not logged in, all the buttons should be disabled except the "LOAD" button.

### Register User

By given **email** and **password,** the app should register a new user in the system.

* In case of **error** (eg. invalid username/password), an appropriate error **message** should be displayed, and the user should be able to **try** to register again.
* Keep the user data in the browser's **session or local storage**.
* After a **successful registration** the **home page should be displayed**.
* Post request: **http://localhost:3030/users/register**



### Logout

The logout action is available to **logged-in users**. Send the following **request** to perform logout:

* Get: **http://localhost:3030/users/logout**

Required **headers** are described in the documentation. Upon success, the **REST service** will return an **empty response**. Clear any session information you’ve stored in browser storage.

If the logout was successful, **redirect** the user to the **Home** page and change the button in navigation.

### Load catches

By clicking it you have to load all the catches from the server and render them like on the picture:

* Pressing the **[Load]** button should **list all** catches. (For all users)
* Pressing the **[Update]** button should send a **PUT** request, updating the catch in **http://localhost:3030/data/catches/:id**. (**Only for the creator of the catch**)
* Pressing the **[Delete]** button should delete the catch from **http://localhost:3030/data/catches/:id**. (**Only for the creator of the catch**)
* Pressing the [Add] button should submit a new catch with the values of the inputs in the fieldset with id="addFrom". (**Only for logged in users**)
* Button [Add] should be disabled in there are no logged in user.
* Buttons [Update] and [Delete] should be disabled if the currently logged-in user is not the author of the catch.



Each catch should have:

* angler - string representing the name of the person who caught the fish
* weight - floating-point number representing the weight of the fish in kilograms
* species - string representing the name of the fish species
* location - string representing the location where the fish was caught
* bait - string representing the bait used to catch the fish
* captureTime - integer number representing the time needed to catch the fish in minutes

Use the following requests to access your data:

* **List All Catches**
  + Endpoint - **http://localhost:3030/data/catches**
  + Method: GET
* **Create a New Catch**
  + Endpoint: **http://localhost:3030/data/catches**
  + Method: POST
  + Request body (JSON): {"angler":"…", "weight":…, "species":"…", "location":"…", "bait":"…", "captureTime":…}
* **Update a Catch**
  + Endpoint: **http://localhost:3030/data/catches/:catchId**
  + Method: PUT
  + Request body (JSON): {"angler":"…", "weight":…, "species":"…", "location":"…", "bait":"…", "captureTime":…}
* **Delete a Catch**
  + Endpoint: **http://localhost:3030/data/catches /:catchId**
  + Method: DELETE