# Exercise: Modular Applications

Exercise problem for the [“JavaScript Applications” course @ SoftUni](https://softuni.bg/courses/javascript-applications).

## Team Manager

Create a JS application for managing teams.

### Libraries

Use the **page** and **lit-html** libraries for routing and templating. Use npm install to install them, as they are included in **package.json**.

### Data

The users are stored standard like in the previous tasks in this course. In this problem by registration you will need one more field, where the **username** should be writen from the user (**email**, **username**, **password**, **rePass**).

Use the following endpoints:

* Teams:

GET **/data/teams**

* List of teams, created by the user:

GET **/data/teams?**where=\_ownerId%3D%22{userId}%22

* Teams, where the user is a member (returns compound object with property team):

GET **/data**/members?where=\_ownerId%3D%22{userId}%22%20AND%20status%3D%22member%22&load=team%3DteamId%3Ateams

* Member count:

GET **/data**/members?where=teamId%3D%22{teamId}%22%20AND%20status%3D%22member%22&count

* List of members (returns compound object with property user):

GET **/data**/members?where=teamId%3D%22{teamId}%22%20AND%20status%3D%22member%22 &load=user%3DuserId%3Ausers

* Create team (body contains name, imageUrl, description):

POST **/data/teams**

* Edit team:

PUT **/data/teams/:id**

* List of pending membership requests (returns compound object with property user):

GET **/data**/members?where=teamId%3D%22{teamId}%22%20AND%20status%3D%22pending%22 &load=user%3DuserId%3Ausers

* Request to join a team (body contains teamId):

POST /data**/members**

* Reject membership request / leave team / remove member:

DELETE **/data/members/:requestId**

* Accept membership request (body contains status set to **"member"**):

PUT /data/members/**:requestId**

Each user can be creator and member of multiple teams.

For the catalogue you don't need to implement pagination, instead the user can scroll down infinitely.

### HTML & CSS

As usual you are provided with HTML and CSS. You may change the HTML by adding attributes, adding URL's where needed and so on as you see fit.

### Views:

#### Navbar

The application has a navigation bar, with **links** **to other pages**. You need to **implement** the **visibility** of the **buttons** in the header, depending on whether there is a **logged-in user** or a **guest**. This is how the header should look like for a **guest** user:

Screenshot_4.png

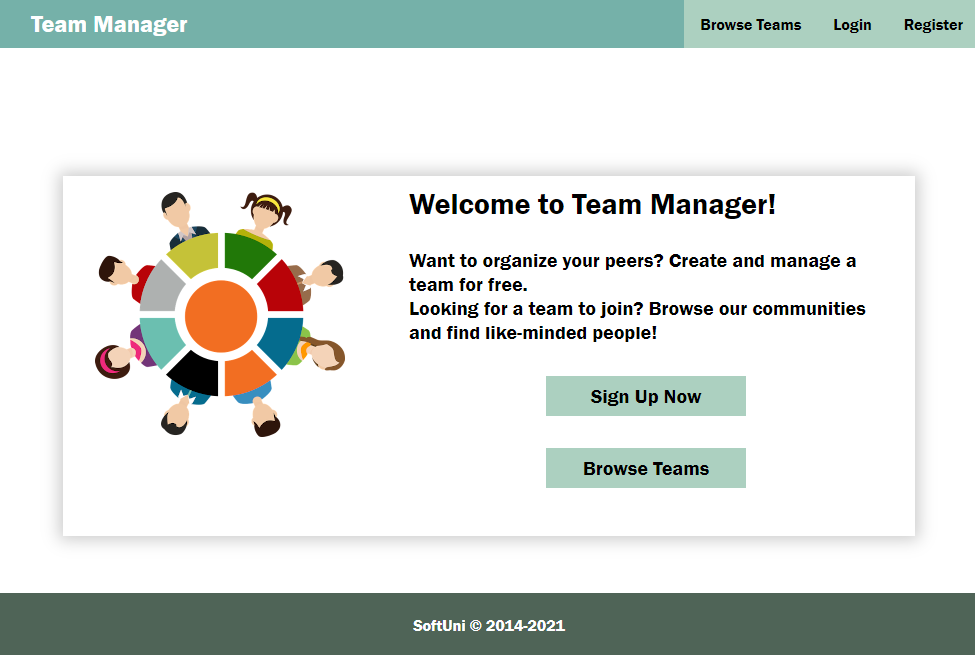
When there is a **logged-in** user, the header should look like this:

Screenshot_3.png

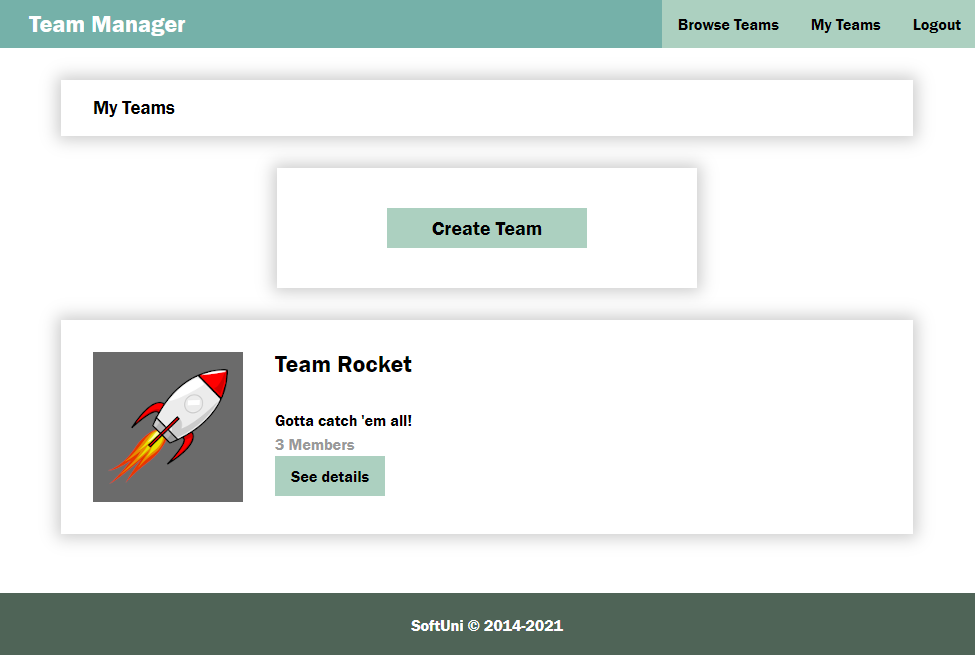
The "**Browse Teams**" button is **always** **visible**. The "**Team Manager**" is link to **home page**.

#### Home

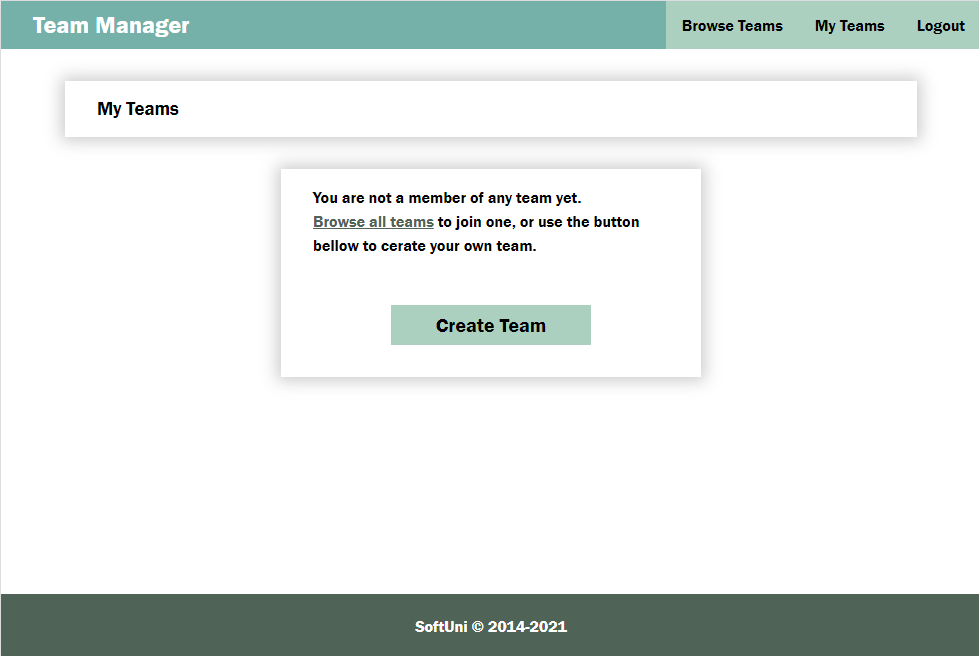
* The **guest** can see the "**Sign Up Now**" button and also have acess to the "**Browse Teams**" through the navbar and the button on home page.



* Logged-in user is redirected to the **My Teams** page, where should be shown **all the teams** the user is **member** of and the option to "**Create Team**".



* If the logged-in user **is NOT a member** or **creator** of team, you should show this view with the button "**Create Team**":

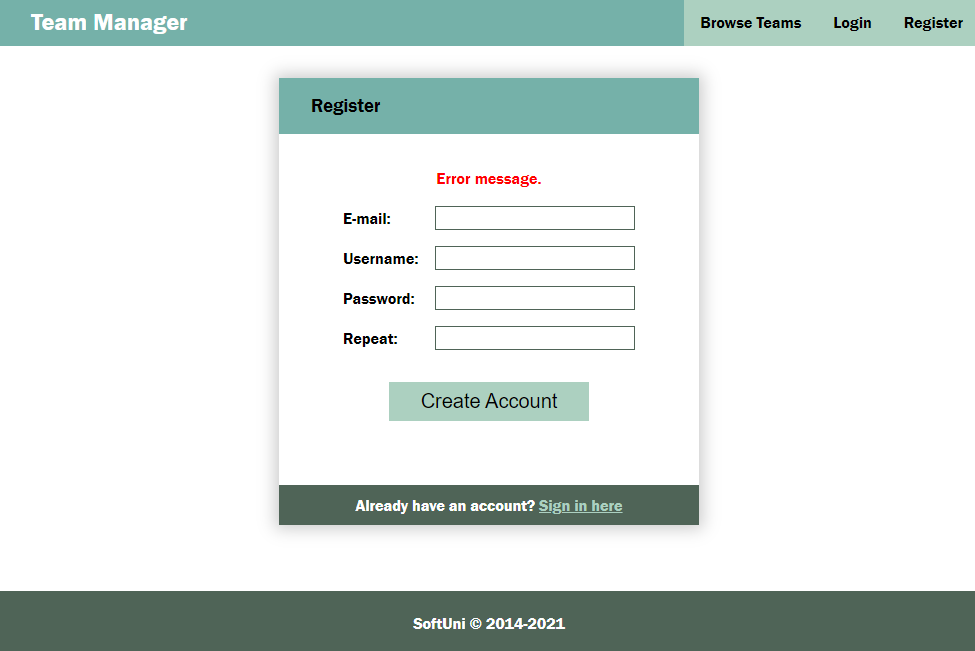


#### Register

You need to **write** the **functionality** for **registration** of new user. By **clicking** the "**Register**" button you have to load the **registration form**. When the "**Create Account**" button of the **form** is clicked you need to **send a post request** to register the new user. The fields by registration have validations as follows:

* **e-mail**: required, valid e-mail,
* **username**: required, at least 4 characters,
* **password**: required, at least 4 characters/digits
* **rePass**: required, match the password

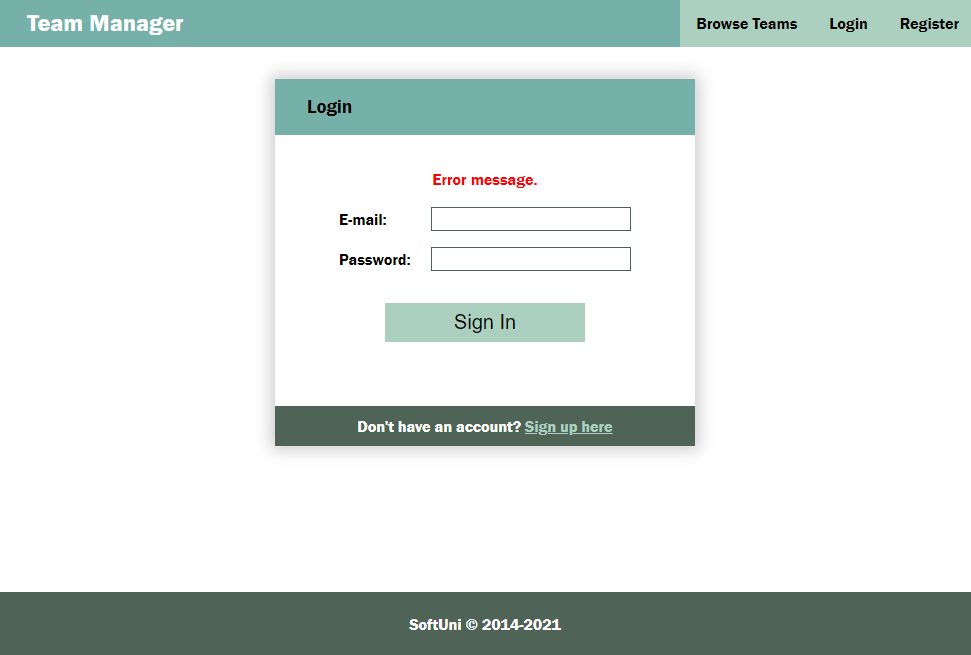
If there is an **error by validation** you need to show the **div with class "error"** to visualize the **message**, otherwise the **div is hidden**.



If the registration is **successful** you can redirect to **My Teams page**.

#### Login

If the user **has already registration**, the **user can login** by using the **login form**. If there is an error, it should be shown the same way as described for the registration.



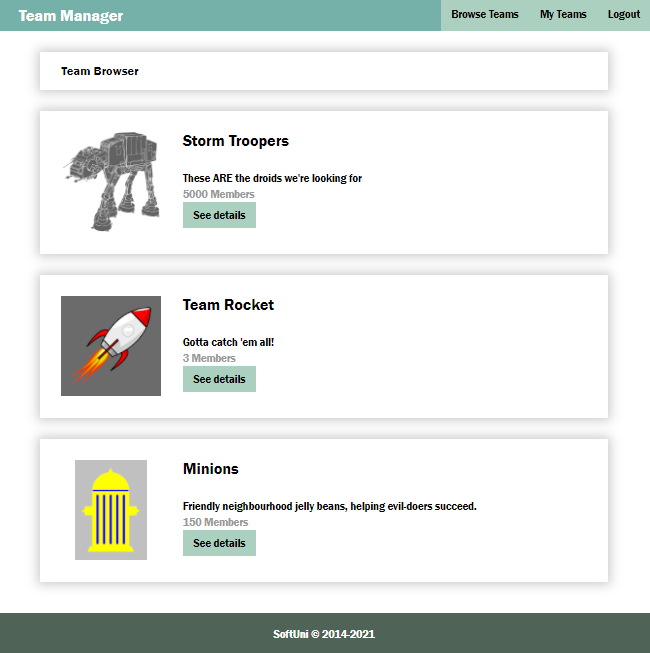
After **successful** login the user should be **redirected** to the **My Teams page**.

#### Logout

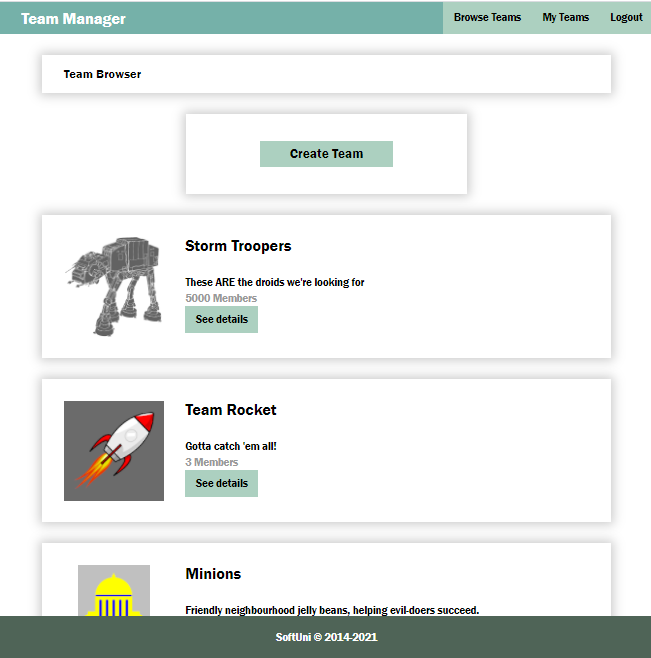
The **logged in** user **can be logged out** by clicking the **logout button**. Write the functionality for this action. After logout **redirect** ot **Home page**.

#### Browse Teams

**Guest user** can see a list of all teams. The button "Create Team" is invisible for guest users.



**Logged in user** can see also the button **"Create Team"**



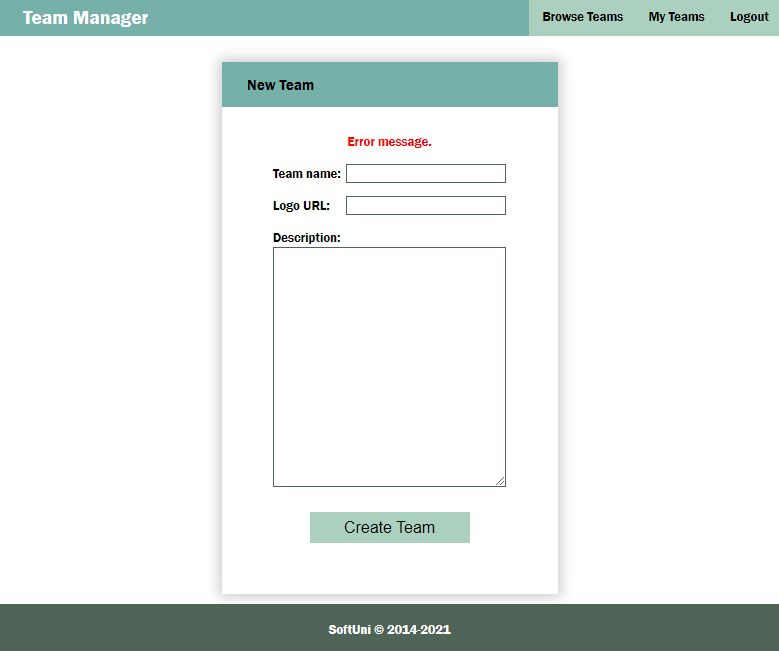
#### New Team

The Create new team page is a form with the following fields and validations:

* **name** (Team name): required, at least 4 characters,
* **logoUrl**: required,
* **description**: required, at least 10 characters

If there is an error by the creation of new team, display the error div with the corresponding message.

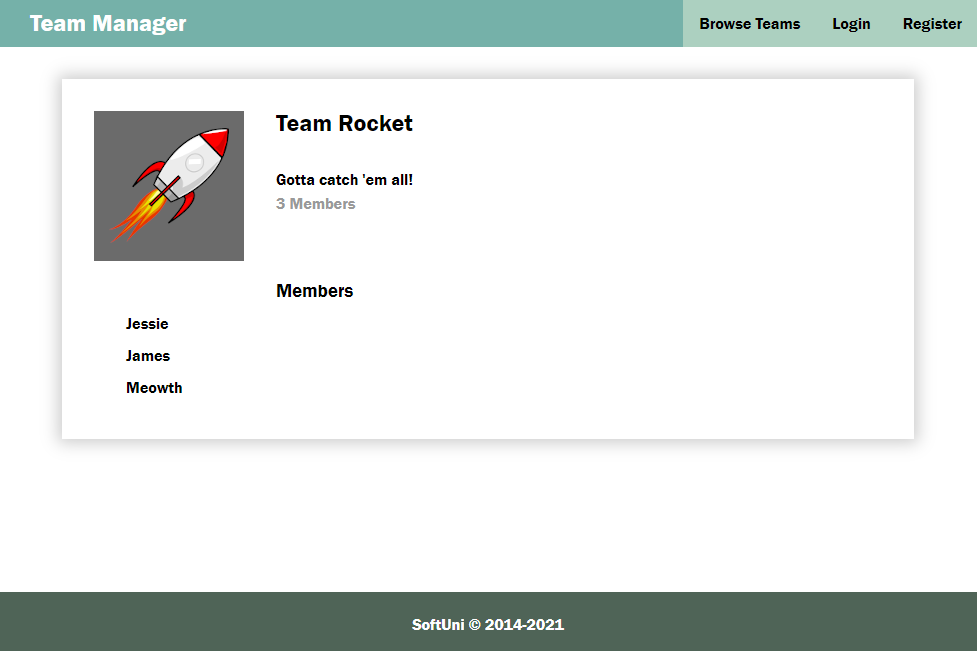
After successful creation of team, **redirect** to "**Team Details" page** of the new created team.



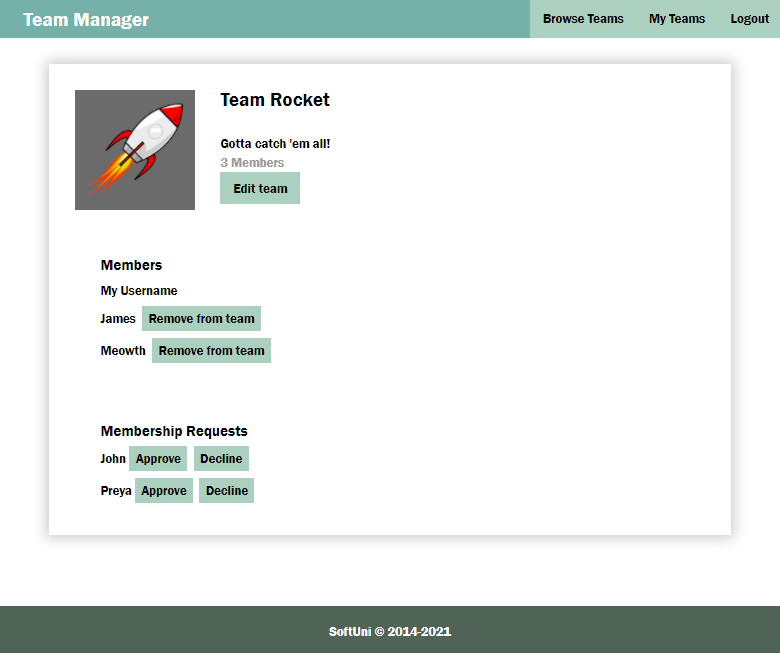
#### Team Details

Another view is the Team Details page. It is accessible for logged-in, but also for guest users. According to the status of the user, there should be visible different things on this page:

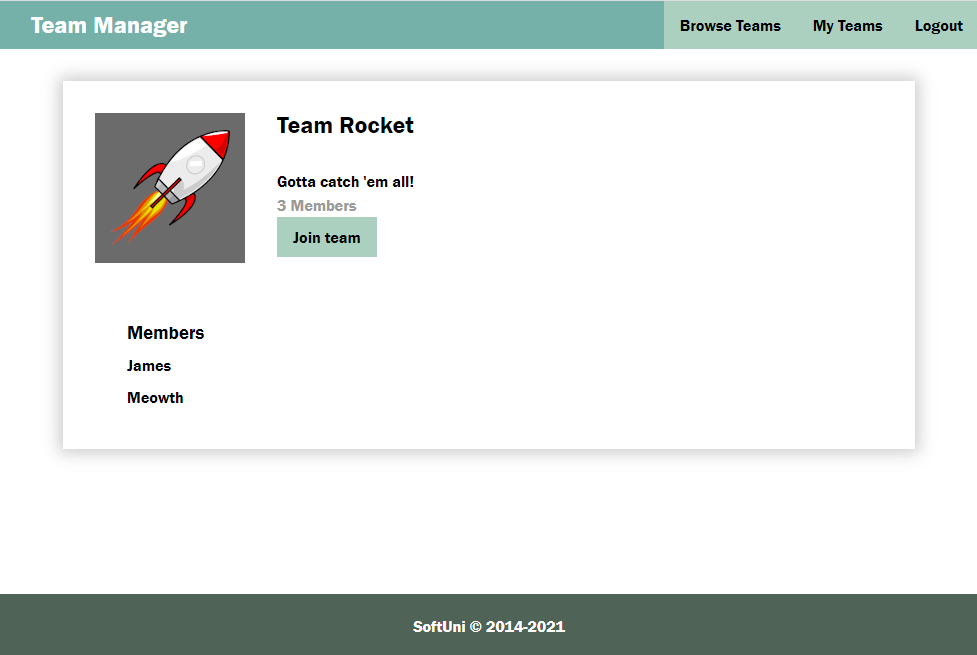
* The **GUEST users** can't see any buttons, but can see the names of the members (the names of the users are the usernames of the registered users).



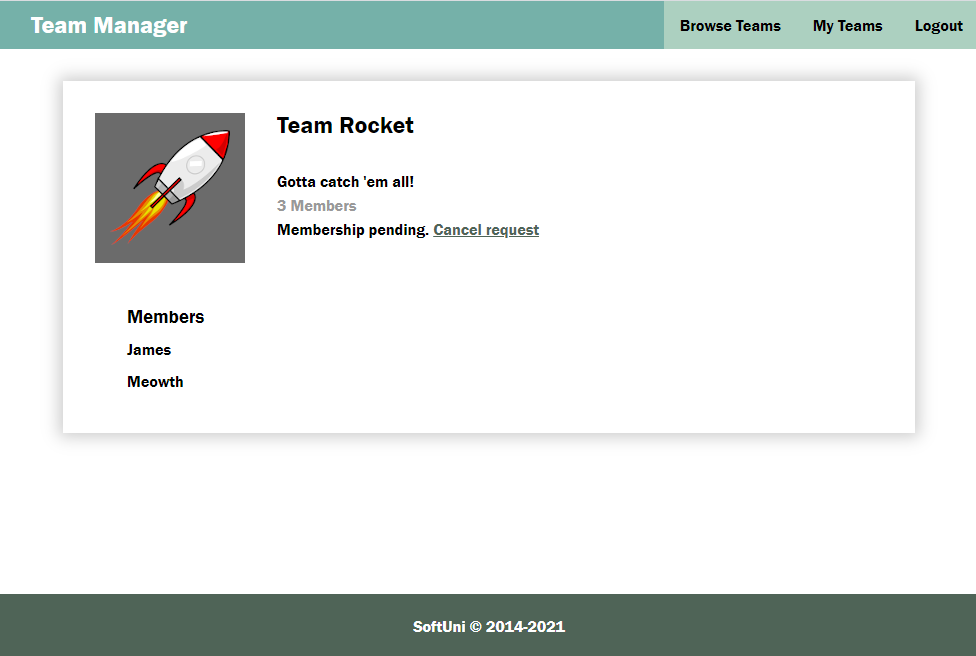
* The **CREATOR** of the team can see the "Edit" button, the "Remove from team" button (for all users, except for himself) and the list with the pending requests for joining the team (also the "Approve" and "Decline" buttons)



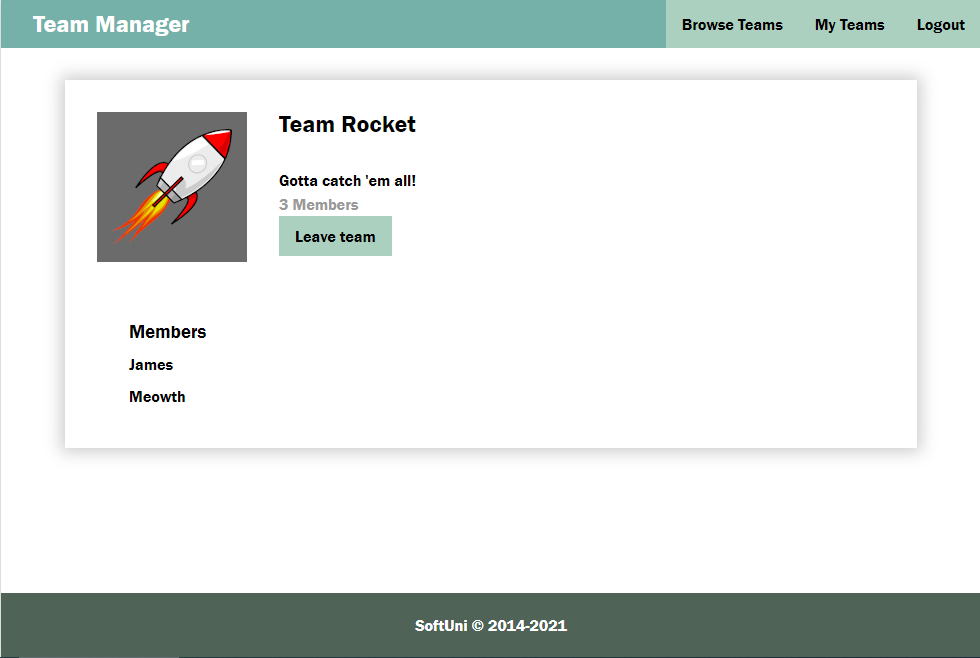
* **Logged user**, who is **NOT member** of the team can see the list with member names and the button "Join Team"



* **Logged** **user** with **PENDING** reques of the team can see the list with member names and Membership pending and "Cancel request" button



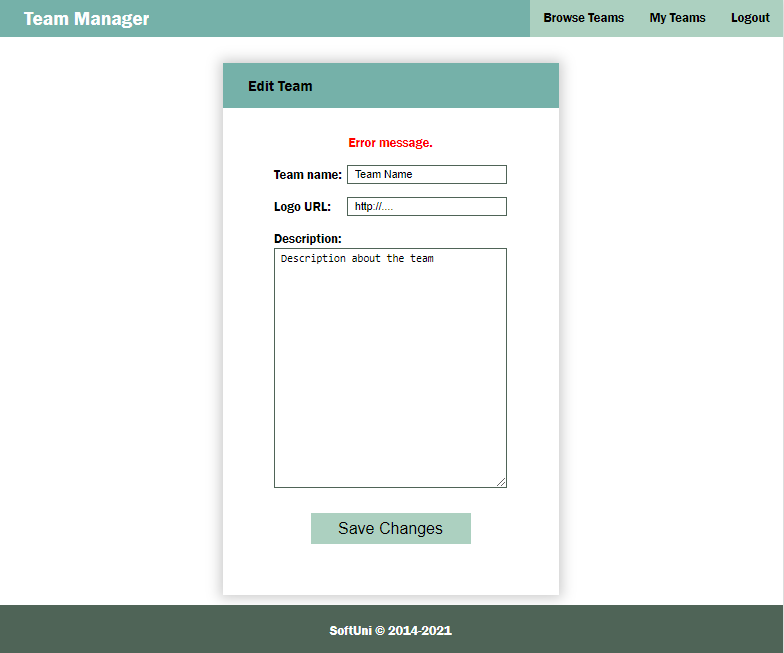
* **Logged** **user**, who is **MEMBER** of the team can see the list with member names of the team and the button "Leave Team"



#### Edit Team

If the **user** is the **creator** of the team, by clicking the "**Edit**" button in the details form, can see the **Edit form**. When loaded the **form fileds should be filled up** with the data of the team. After change and before sending the **PUT request** a **validation** should be made. The validation is the same as by creating a team. If **error** occurs it should be shown in the **div with class "error"**.

After **successful** edit, **redirect to Details page**.



#### Modal

An HTML example is included with **id "overlay"** (initially set to display:none). If there is a need of **confirmation**, **something is wrong** or there is an **error** (not an error in the forms validation) the **modal** should be shown with the corresponding **message** and **buttons**. You can use this modal by your own decission – for successful registration, login, logout and so on.

