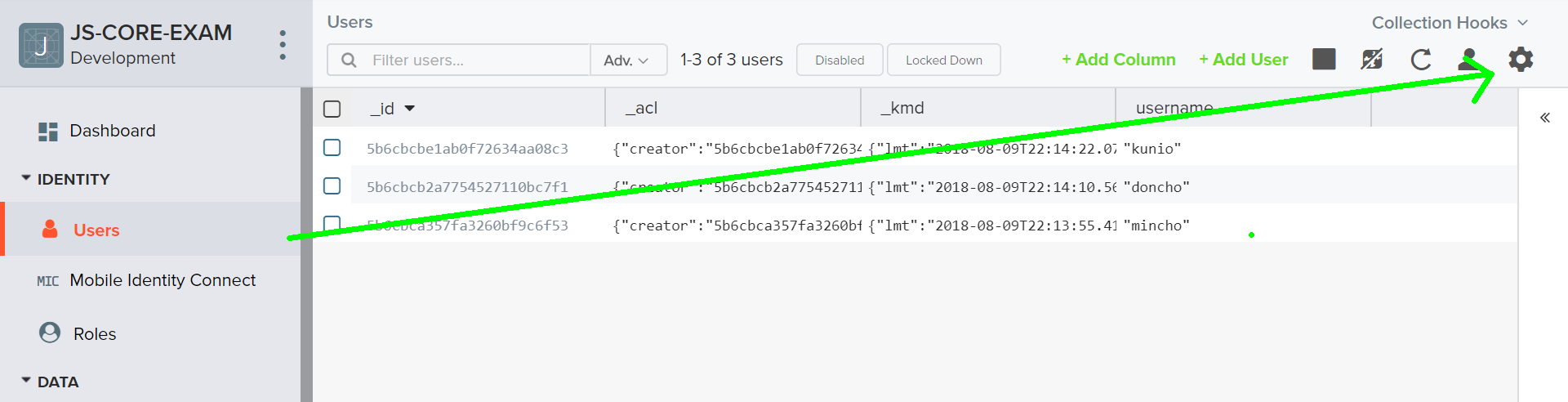
# deJS Apps Exam – Meme Lounge Single Page Application

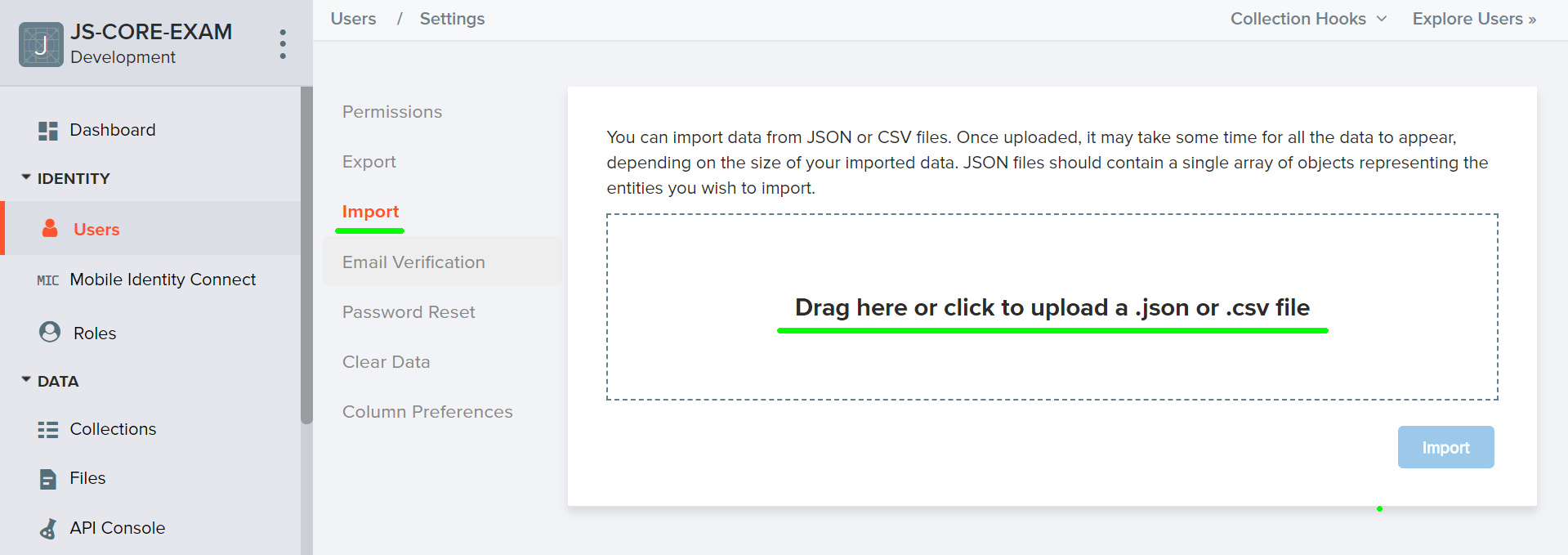
Using libraries like **jQuery**, **Handlebars** and **Sammy** is allowed but is not obligatory. You are assigned to implement a **Web application** (SPA) using HTML5, JavaScript, AJAX, REST and JSON with cloud-based backend (Kinvey). The app keeps **users** and **listings**. Users can **register**, **login**, **logout** , view a **catalog** with all memes sorted by **time** (a helper function will be given), **create** a meme, **edit/delete** their **own** memes,show meme **details** and a **userProfile** where they can view their **own** **memes and delete their profiles**.

## Create a Kinvey REST Service

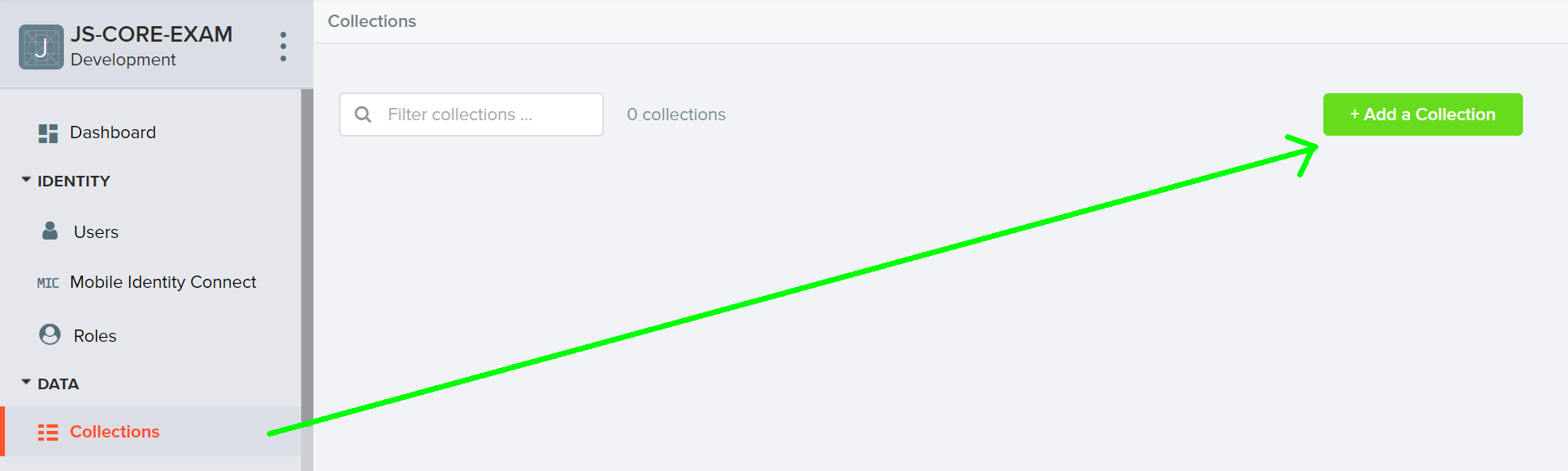
Register at **Kinvey.com** and create an application to keep your data in the cloud.

In the **Users** collection, import the provided JSON file with sample users to get started with template data. In the **Kinvey Console**, select **Users** from the navigation of the left, click **Settings** in the upper right then scroll down to the **Import** section:





Create a collection **memes**. Each **meme** has a **title ,description ,imageUrl** and an **creator** (username of the creator).



You are also given **one** more json files to **import** data for **cars.**

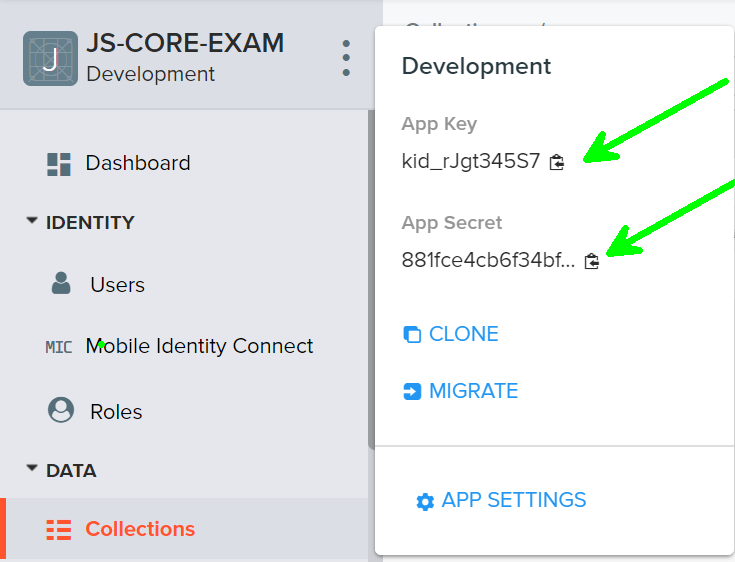
## Test the Kinvey REST Services

Using **Postman** or other HTTP client tool (you can use Kinvey’s built-in **API Console**), test the REST service endpoints:

### User Registration (Sign Up)

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/user/***app\_id***/](https://baas.kinvey.com/user/app_id/) | |
| Request headers | Authorization: Basic base64(app\_id:app\_secret)  Content-Type: application/json |
| Request body | {  "username": "testuser",  "password": "testuserpass890"  } |
| Response  201 Created | {  "\_id": "59930c78a743e20c7d3fca77",  "username": "testuser",  "password": "testuserpass890"  } |
| Error response  409 Conflict | { "error": "UserAlreadyExists", "description": "This username is already taken. Please retry your request with a different username", "debug": "" } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

The request needs “**Basic**” authentication. Use the Kinvey **App Key** and Kinvey **App Secret** as credentials.



### User Login

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/user/***app\_id***/login](https://baas.kinvey.com/user/app_id/login) | |
| Request headers | Authorization: Basic base64(app\_id:app\_secret)  Content-Type: application/json |
| Request body | {  "username": "testuser",  "password": "testuserpass890"  } |
| Response  200 OK | {  "\_id": "59930c78a743e20c7d3fca77",  "username": "testuser"  "\_kmd": {  "**authtoken**": "**8e6471bc-3712-4cfb-b92e-50e62a0c80….Duj5fHdM /7XHIe6KdY=**"  …  },  …  } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

Successful login returns an “**authtoken**” which is later used to authenticate the CRUD operations.

### User Logout

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/user/***app\_id***/\_logout](https://baas.kinvey.com/user/app_id/_logout) | |
| Request headers | Authorization: Kinvey **authtoken** |
| Response  204 No Content |  |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

To logout, you need to provide the “**authtoken**” given by login / register as “**Kinvey**” authorization header.

### List all Memes (MemeFeed – sorted by meme time, descending)

|  |  |
| --- | --- |
| **GET** [https://baas.kinvey.com/appdata/***app\_id***/memes](https://baas.kinvey.com/appdata/app_id/memes)?query={}&sort={"\_kmd.ect": -1} | |
| Request headers | Authorization: Kinvey authtoken |
| Response  200 OK | [  {  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  "\_acl": {  "creator": "kid\_B1zfi\_e\_Z"  },  "\_kmd": {  "lmt": "2017-08-15T14:40:19.182Z",  "ect": "2017-08-15T14:40:19.182Z"  }  },...  ] |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Create Meme

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/appdata/***app\_id***/memes](https://baas.kinvey.com/appdata/app_id/memes) | |
| Request headers | Authorization: Kinvey authtoken  Content-Type: application/json |
| Request body | {  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  } |
| Response  201 Created | {  "\_id": "59931398996ab5127d2a84d1",  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Edit Meme

|  |  |
| --- | --- |
| **PUT** [https://baas.kinvey.com/appdata/***app\_id***/memes/**meme\_id**](https://baas.kinvey.com/appdata/app_id/memes/meme_id) | |
| Request headers | Authorization: Kinvey authtoken  Content-Type: application/json |
| Request body | {  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  } |
| Response  200 Ok | {  "\_id": "59931398996ab5127d2a84d1",  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Delete Meme

|  |  |
| --- | --- |
| **DELETE** [https://baas.kinvey.com/appdata/***app\_id***/memes/***meme\_id***](https://baas.kinvey.com/appdata/app_id/memes/meme_id) | |
| Request headers | Authorization: Kinvey authtoken |
| Response  200 OK | {  "count": 1  } |
| Error response  404 Not Found | { "error": "EntityNotFound", "description": "This entity not found in the collection", "debug": "" } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Get User By Id

|  |  |
| --- | --- |
| **GET** [https://baas.kinvey.com/user/**app\_Id**/**userId**/](https://baas.kinvey.com/user/app_Id/userId/) | |
| Request headers | Authorization: Kinvey authtoken |
| Response  200 OK | [  {  "username": "testuser",  “email": “email”,  “imageUrl”:”url”  }  ] |
|  | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

**User Memes(View all Memes by username, sorted by post time, descending)**

|  |  |
| --- | --- |
| **GET** [https://baas.kinvey.com/appdata/**app\_id**/memes?query={"creator":"**username**"}](https://baas.kinvey.com/appdata/app_id/memes?query=%7b%22creator%22:%22username%22%7d)&sort={"\_kmd.ect": -1} | |
| Request headers | Authorization: Kinvey authtoken |
| Response  200 OK | [  {  "\_id": "599307d36af42b495f87bad2",  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  "\_acl": {  "creator": "kid\_B1zfi\_e\_Z"  },  "\_kmd": {  "lmt": "2017-08-15T14:40:19.182Z",  "ect": "2017-08-15T14:40:19.182Z"  }  },  {  "\_id": "5993178dbce993cf532bbe29",  "title": "Very Funny Meme",  "description":"I cand hold it its hilarious",  "imageUrl":"url-link",  "\_acl": {  "creator": "59930c78a743e20c7d3fca77"  },  "\_kmd": {  "lmt": "2017-08-15T15:47:25.376Z",  "ect": "2017-08-15T15:47:25.376Z"  }  }  ] |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Delete User

|  |  |
| --- | --- |
| **DELETE** [https://baas.kinvey.com/user/**app\_id**/**user\_id**/](https://baas.kinvey.com/user/app_id/user_id/) | |
| Request headers | Authorization: Kinvey authtoken |
| Response  200 OK | {  "count": 1  } |
| Error response  404 Not Found | { "error": "EntityNotFound", "description": "This entity not found in the collection", "debug": "" } |
| Error response  401 Unauthorized | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

## Meme Lounge – HTML and CSS

You аre given the Web design of the application as **HTML** + **CSS** files.

* Initially all views and forms are shown by the HTML. Your application may **hide** by CSS (display: none) or **delete** from the DOM all unneeded elements or just display the views it needs to display.
* You may render the views / forms / components with **jQuery** or **Handlebars**.

**Important**: don’t change the elements’ **class name** and **id**. Don’t rename form fields / link names / ids. You are **allowed** to add **data attributes** to any elements. You may modify **href attributes** of links and add **action/method attributes** to forms, to allow the use of a routing library.

## Meme Lounge Client-Side Web Application

**Design** and **implement** a client-side front-end app (SPA) for managing **memes**. Implement the functionality described below.

### Notifications

The application should notify the users about the result of their actions.

* In case of successful action an **informational (green) notification message** should be shown, which disappears automatically after 3 seconds or manually when the user clicks it.



* In case of **error**, an **error notification message** (red) should be shown which disappears on user click.



* During the AJAX calls a **loading notification message (blue)** should be shown. It should disappear automatically as soon as the AJAX call is completed.



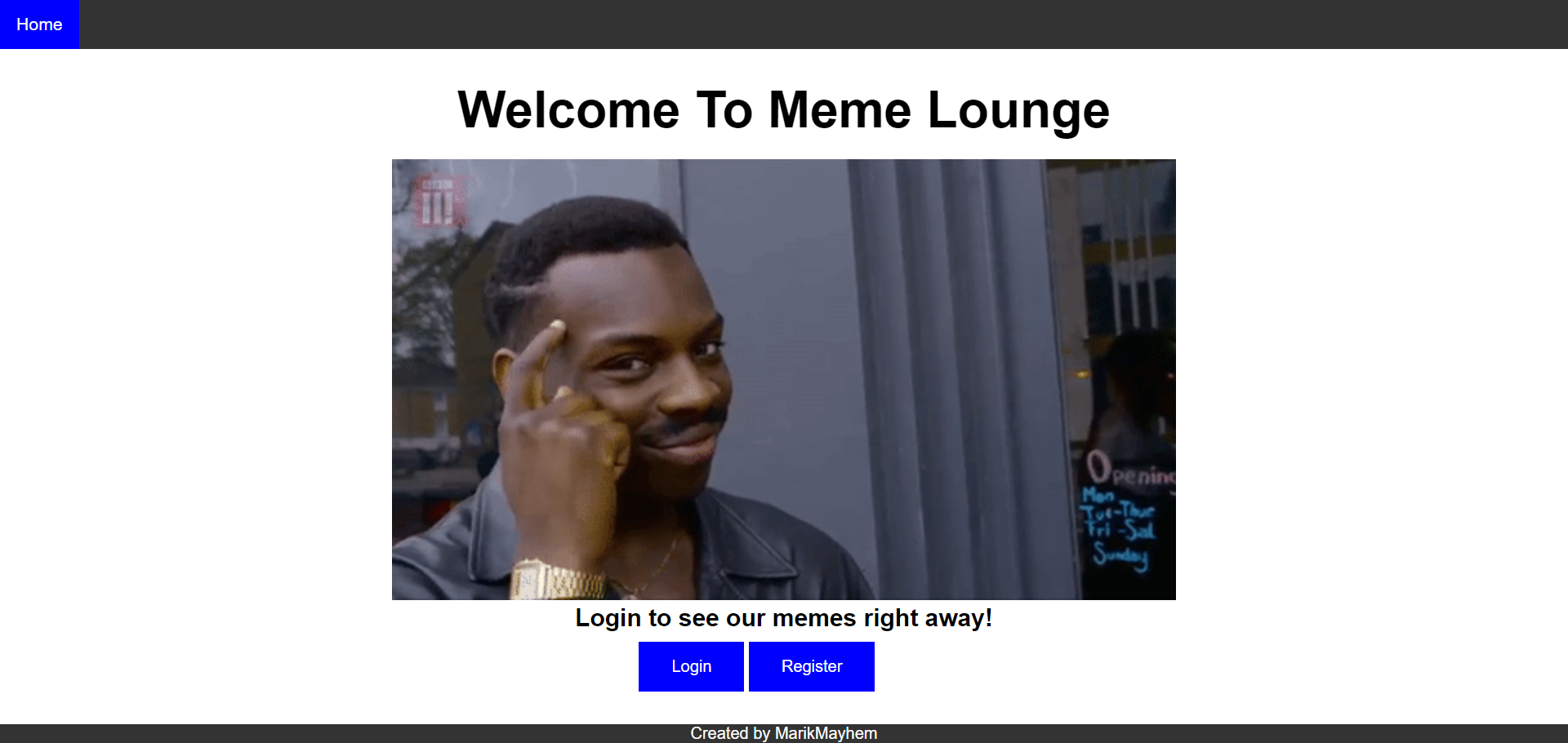
### NavBar (5 pts)

Implement a **NavBar** for the app: navigation links should correctly change the current screen (view).

* Clicking on the links in the **NavBar** or the links on each **individual** meme should display the view behind the link (views are sections in the HTML code).
* Your application may **hide** by CSS (display: none) or **delete** from the DOM all unneeded elements or just display the views it needs to display.
* The given „**Create Meme**, **Welcome, My Profile and logout**“ should be visible **only** for logged in users. Anonymous users can **only** view the **home and the regular navbar** section. 

### Home Screen (10 pts)

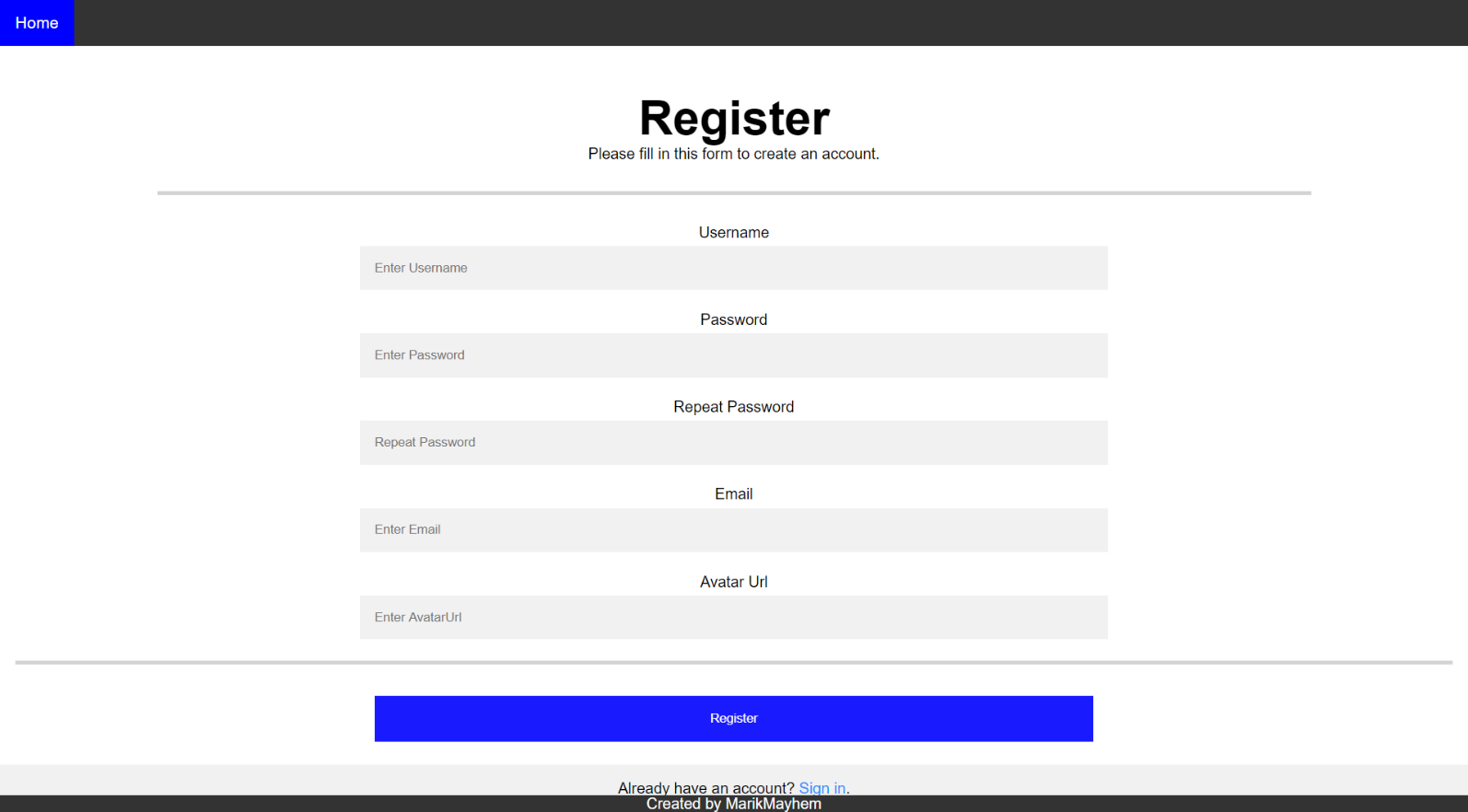
The initial screen should display the navigation, register, login and the initial image + footer.



### Register User (10 pts)

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

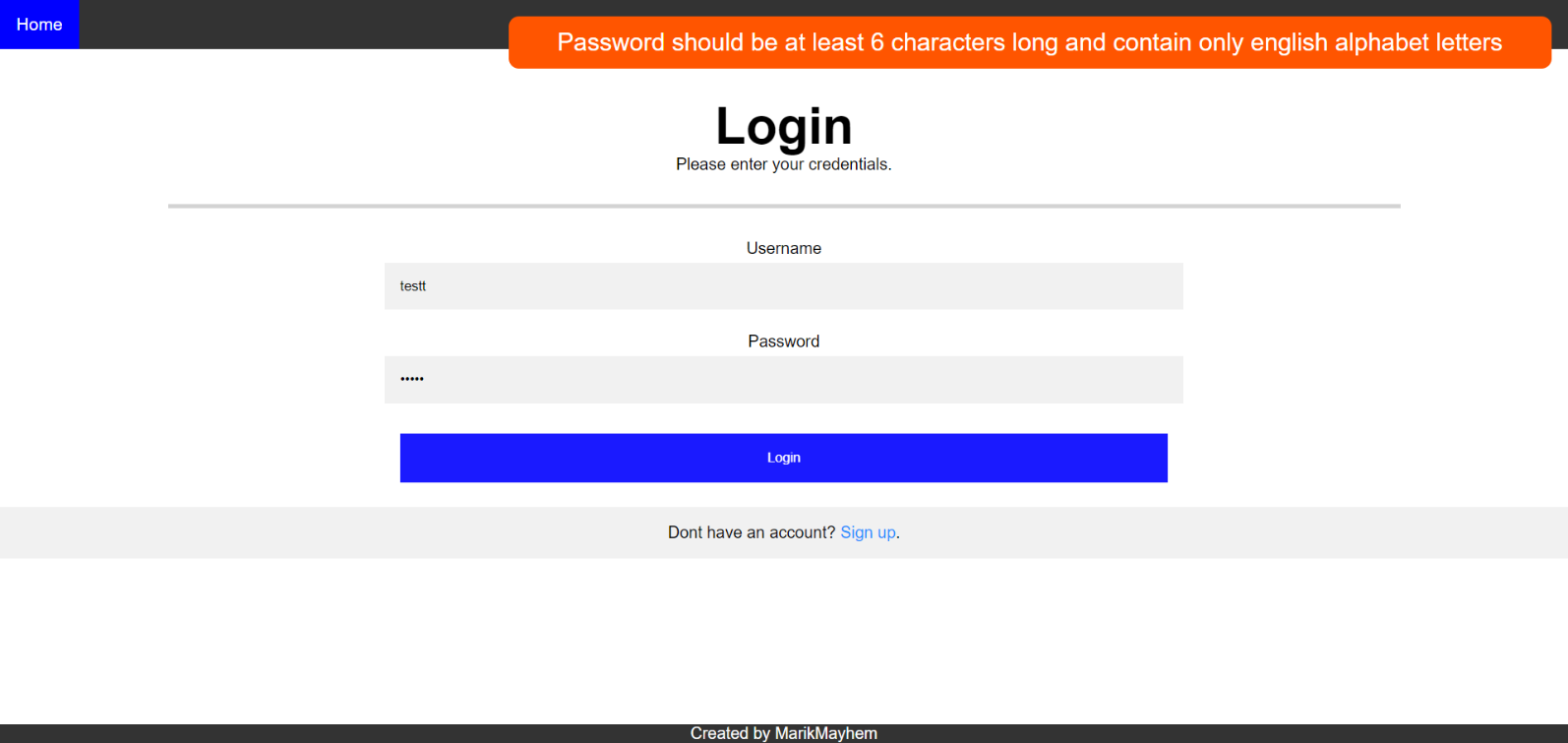
* After a **successful registration**, a notification message “User registration successful.” should be displayed and the **catalog** with all **listings** should be shown.
* You **need** to validate the **input**. A username **should** be at **least** 3 characters **long** and should contain only english alphabet **letters**. A user‘s password should be at **least** 6 characters **long** and should contain only english alphabet **letters** and **digits**. Both passwords **must** match.
* 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 session data in the browser’s **session storage**.



### Login User (5 pts)

By given **username** and **password** the app should be able to login an existing user.

* After a **successful login**, a notification message “Login successful.” should be displayed and the user home screen should be displayed.
* In case of **error**, an appropriate error message should be displayed and the user should be able to fill the login form again.
* **Form validation** should be the **same** as register.
* Keep the user session data in the browser’s **session storage**.
* Clear **all** input fields after **successful** login.



### Logout (5 pts)

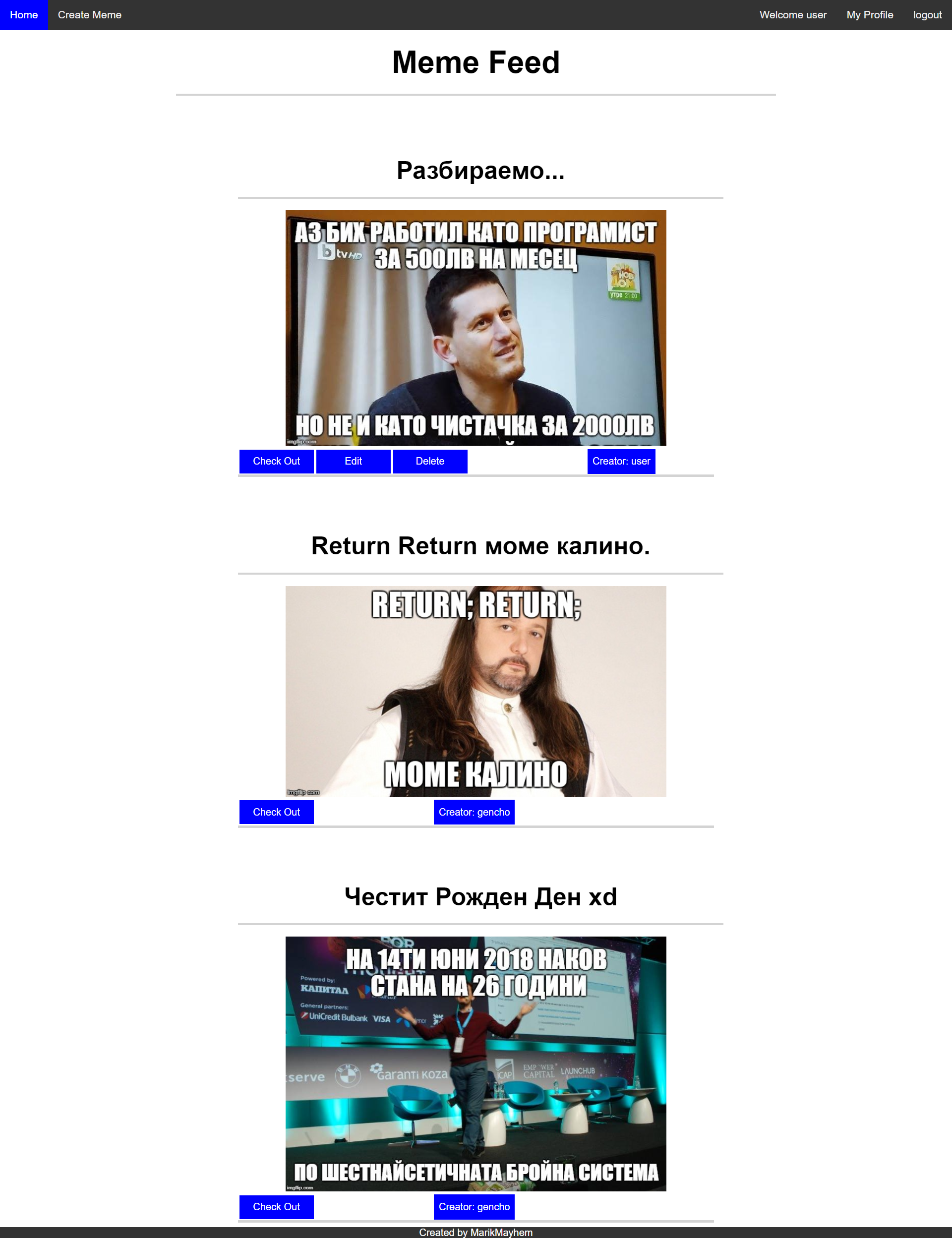
Successfully logged in user should be able to **logout** from the app.

* After a **successful** logout, a **notification** message “Logout successful.” should be displayed.
* After successful logout, the **Sign In screen** should be shown.
* The **“logout” REST service** at the back-end should be obligatory called at logout.
* All local information in the browser (**user session data**) about the current user should be deleted.

### Meme Feed (15 pts)

Successfully logged users should be welcomed by the **meme feed** screen which can be accessed **after** clicking the **Home** link in the **NavBar**. The catalog contains **all** memes on the site.

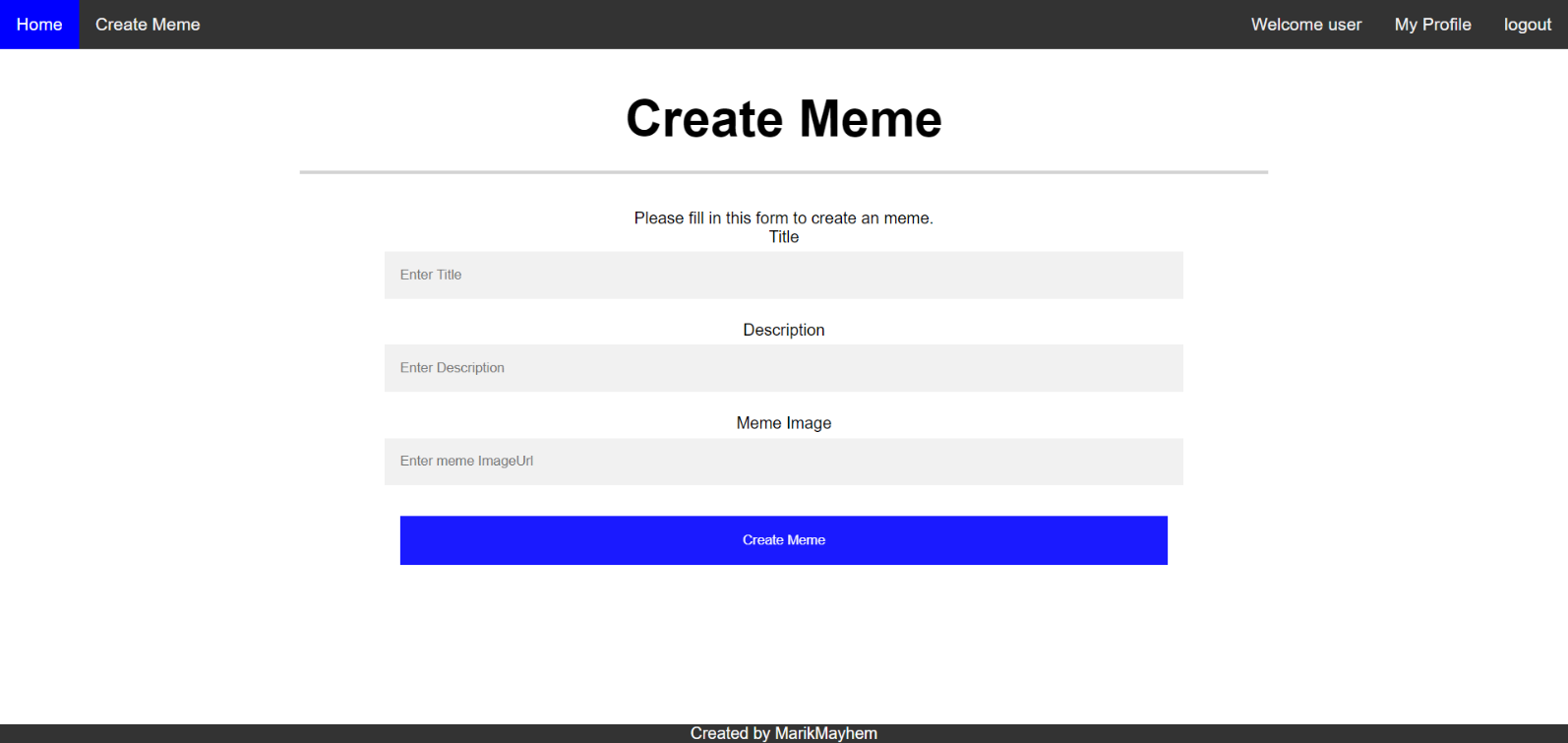
* The **memes** should be listed in the **format** as shown in the Web design (see the screenshot below).
* Each **meme** has **title** and image;
* When the **checkout(meme title and meme image)/edit** link is clicked, your app should **redirect** to the details/edit **section** of the meme.
* When the creator button is clicked you should redirect to the **creator profile**.
* Meme **creators** can edit/delete their **own** memes.
* In case of **error** (e.g. Internet connection lost), an error message should be displayed.
* In case of **no memes**, display ‘No memes in database’.



### Create Meme Screen (10 pts)

Logged in users can **create** meme. Clicking the **[Create Meme]** link should open a form.

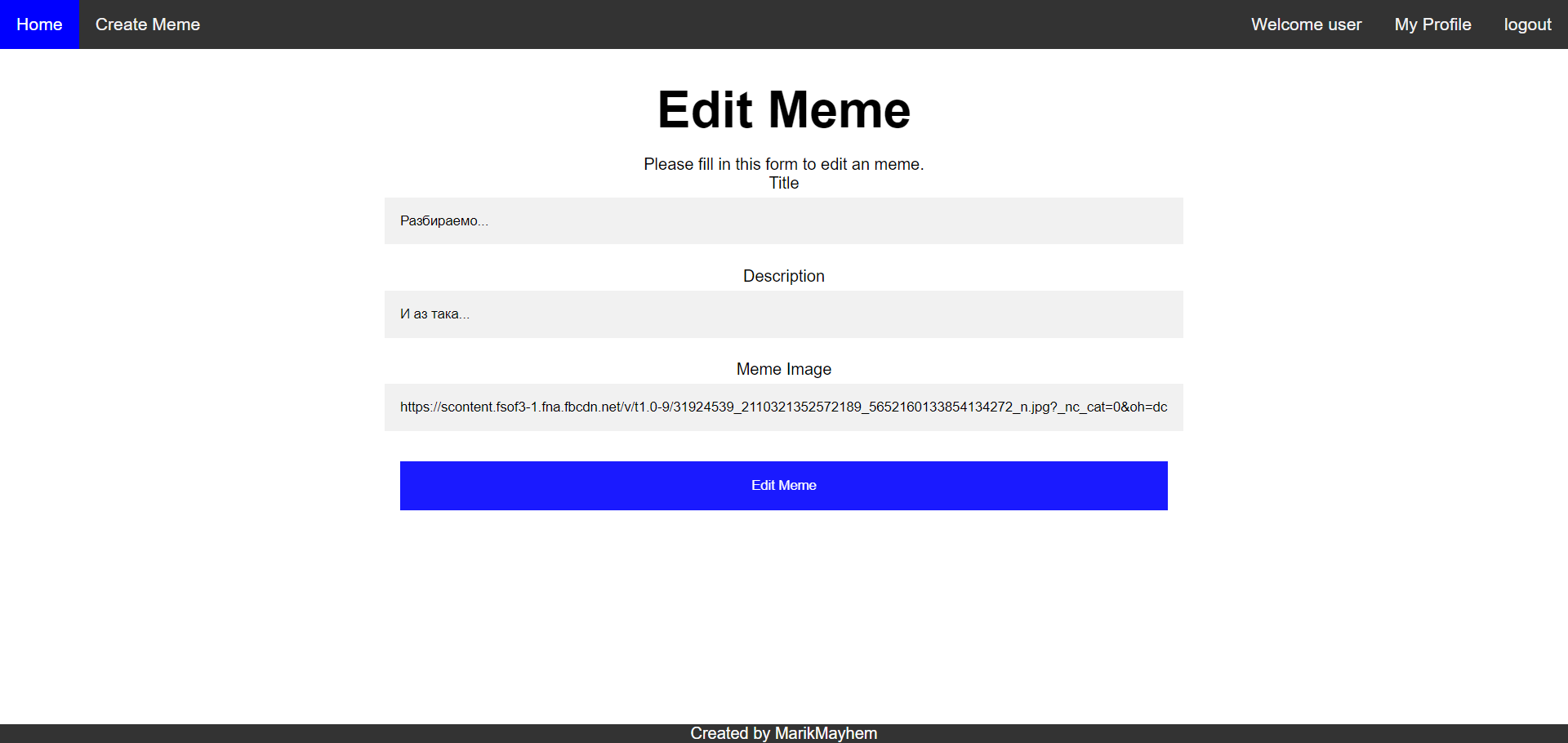
* After a **successful** meme creation, a notification message “meme created.” should be displayed and the **meme feed** should be shown.
* The **title** length must not exceed 33 characters!
* The **description** length must not exceed 450 characters and should be at least 30!
* Link url should always start with “**http**”.
* In case of **empty/incorrect** input fields an appropriate **error** message should be **displayed** and the user should be able to fill the **form** again.
* Ensure you handle properly all HTML **special characters**, e.g. the description text could hold "*Hi, <peter>*".



### Edit Meme Screen (15 pts)

Sellers **should** be able to **edit** their own **listings**. Clicking the [**Edit**] link on each meme should open an edit **form**. Inside the edit form all input fields should be **filled** with the **current** data of each post.

* After a **successful** meme update, a notification message “Meme {memeTitle} updated.” should be displayed and the **meme Feed** should be shown.
* **The validation is same as create meme!**
* Ensure you handle properly all HTML **special characters**, e.g. the description text could hold "*Hi, <peter>*".



### Delete Meme (5 pts)

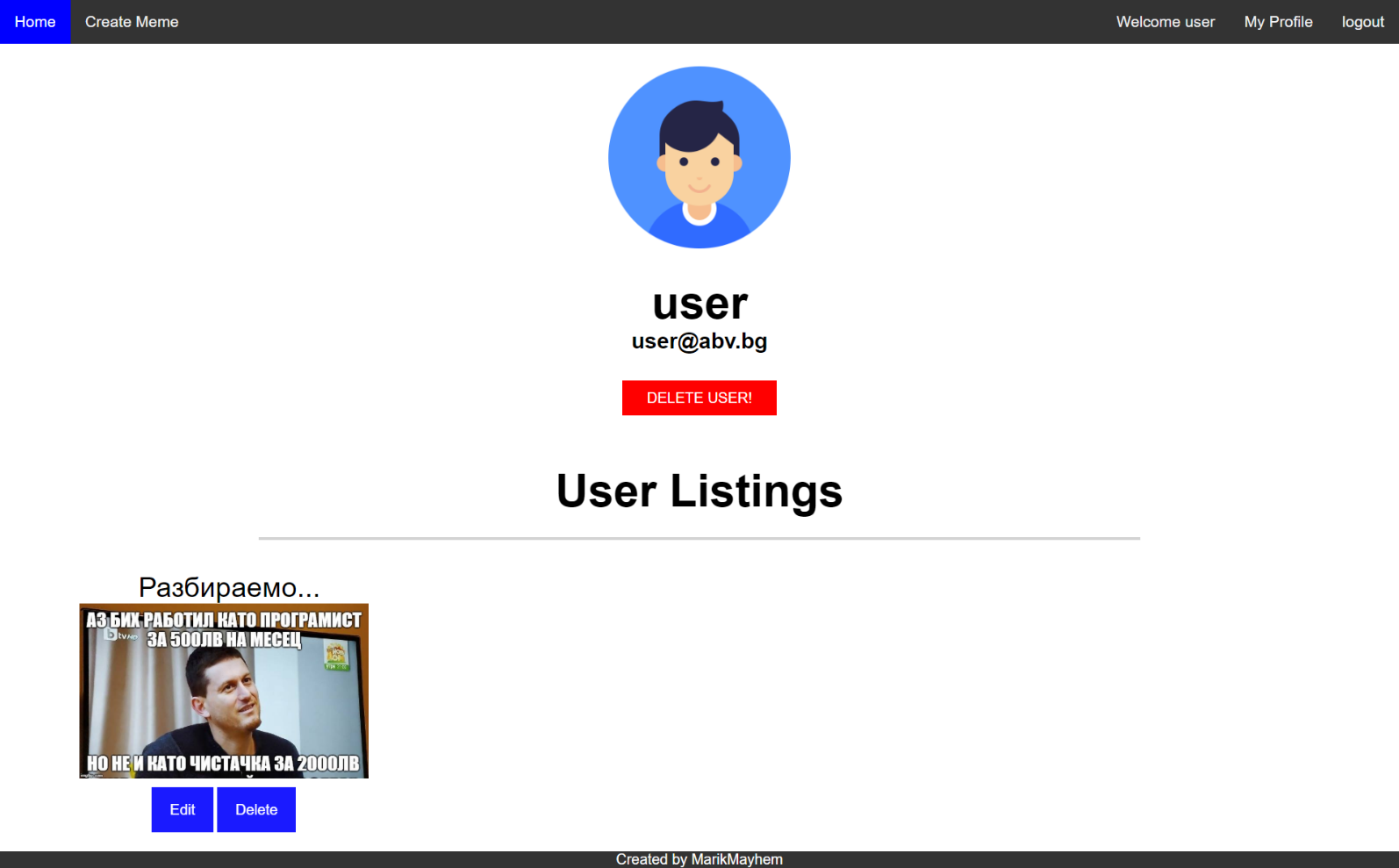
Sellers **should** be able to **delete** their **own** memes by clicking the [**Delete**] button.

* After **successful** meme delete a notification message “Meme deleted.” should be displayed and the **meme feed** should be **shown**.
* In case of **error** (e.g. Internet connection lost / unauthorized request / missing message), an error message should be displayed.
* Deleting works immediately, with **no confirmation**.

### User Profile (15 pts)

Each **user** should be able to view his own profile by clicking [**My Profile**]. The user can also see other people profiles by clicking the links on their memes.

* Each user profile should display user info - **avatarImage**,**username** and **email**.
* The **memes** should be listed like in the **meme Feed** section below the user Info.
* If the profile is of the current logged in user you should also have Delete user button.
* The user delete button should destroy the current user profile.
* After **successful** user delete a notification message “User deleted.” should be displayed and the **meme feed** should be **shown**.
* In case of **error** (e.g. Internet connection lost), an error message should be displayed.
* In case of **no memes**, display ‘No memes in database’.



### Meme Details (10 pts)

Each **meme** has a details view. By clicking the [**details**] link **partial** information about each meme should be listed!

* In case of **error** (e.g. Internet connection lost), an error message should be displayed.
* Ensure you handle properly all HTML **special characters**, e.g. the description text could hold "*Hi, <peter>*".



## Subtmitting Your Solution

Place in a ZIP file your project folder. Exclude the node\_modules folder. Upload the archive to the Judge.