JS Apps Exam – Chirper Single Page Application

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 chirps (tweets)**. Users can **register**, **login, logout**, view **feed** page (all chirps from followers sorted by time in descending), **create** a chirp, **delete** a chirp, view **own** chirps, **discover** new people, get **all chirps** by specific user (sorted by time in descending), **follow** or **unfollow** a user.

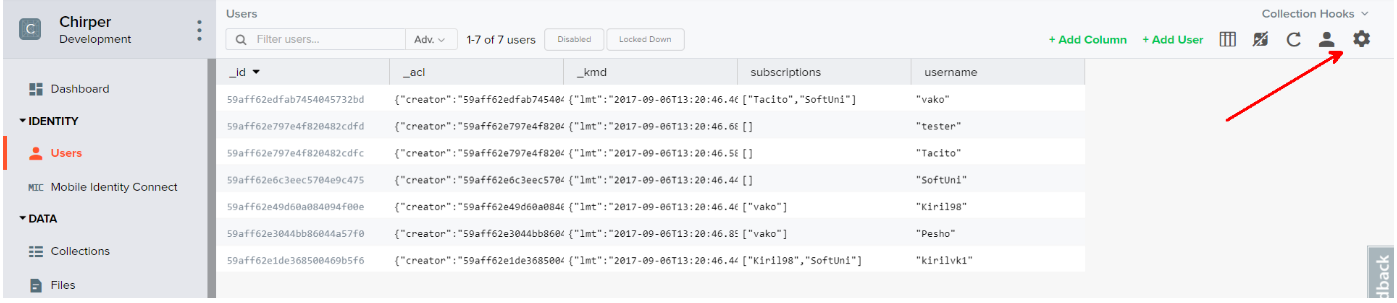
Using libraries like **jQuery**, **Handlebars** and **Sammy** is allowed but is not obligatory.

Problem 1. Create a Kinvey REST Service

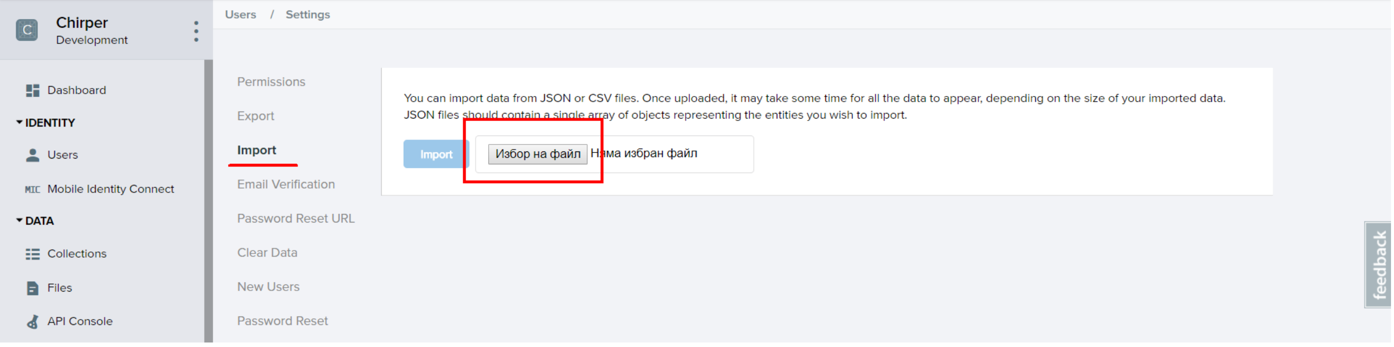
Register at **Kinvey.com** and create an application to keep your data in the cloud.

Each user has a **username**, **password** and **subscriptions (array of usernames)**.

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:

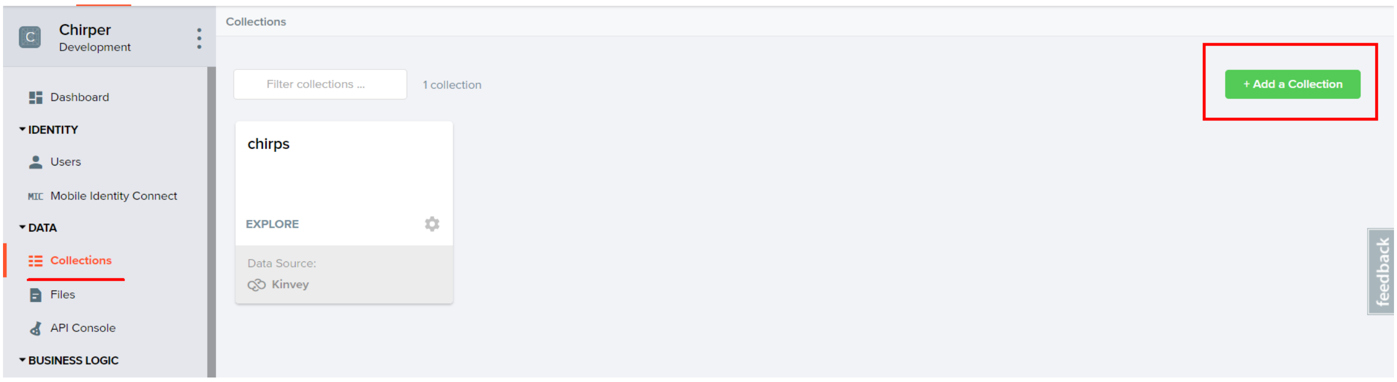


_id	_acl	_kmd	subscriptions	username
59aff62edfab7454045732bd	{"creator":"59aff62edfab745404	{"lmt":"2017-09-06T13:20:46.46	["Tacito","SoftUni"]	"vako"
59aff62e797e4f820482cdfd	{"creator":"59aff62e797e4f8204	{"lmt":"2017-09-06T13:20:46.68	[]	"tester"
59aff62e797e4f820482cdfc	{"creator":"59aff62e797e4f8204	{"lmt":"2017-09-06T13:20:46.58	[]	"Tacito"
59aff62e6c3eec5704e9c475	{"creator":"59aff62e6c3eec5704	{"lmt":"2017-09-06T13:20:46.44	[]	"SoftUni"
59aff62e49d60a084094f00e	{"creator":"59aff62e49d60a0846	{"lmt":"2017-09-06T13:20:46.46	["vako"]	"Kiril98"
59aff62e3044bb86044a57f0	{"creator":"59aff62e3044bb8604	{"lmt":"2017-09-06T13:20:46.85	["vako"]	"Pesho"
59aff62e1de368500469b5f6	{"creator":"59aff62e1de3685004	{"lmt":"2017-09-06T13:20:46.44	["Kiril98","SoftUni"]	"kirilvk1"



You can import data from JSON or CSV files. Once uploaded, it may take some time for all the data to appear, depending on the size of your imported data. JSON files should contain a single array of objects representing the entities you wish to import.

Create a collection **chirps**. Each chirp has a **text** (string with max length 150 and cannot be empty) and an **author** (username of the creator).



© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

You are also given **one** more json file to **import** your **chirps**.

Problem 2. 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)

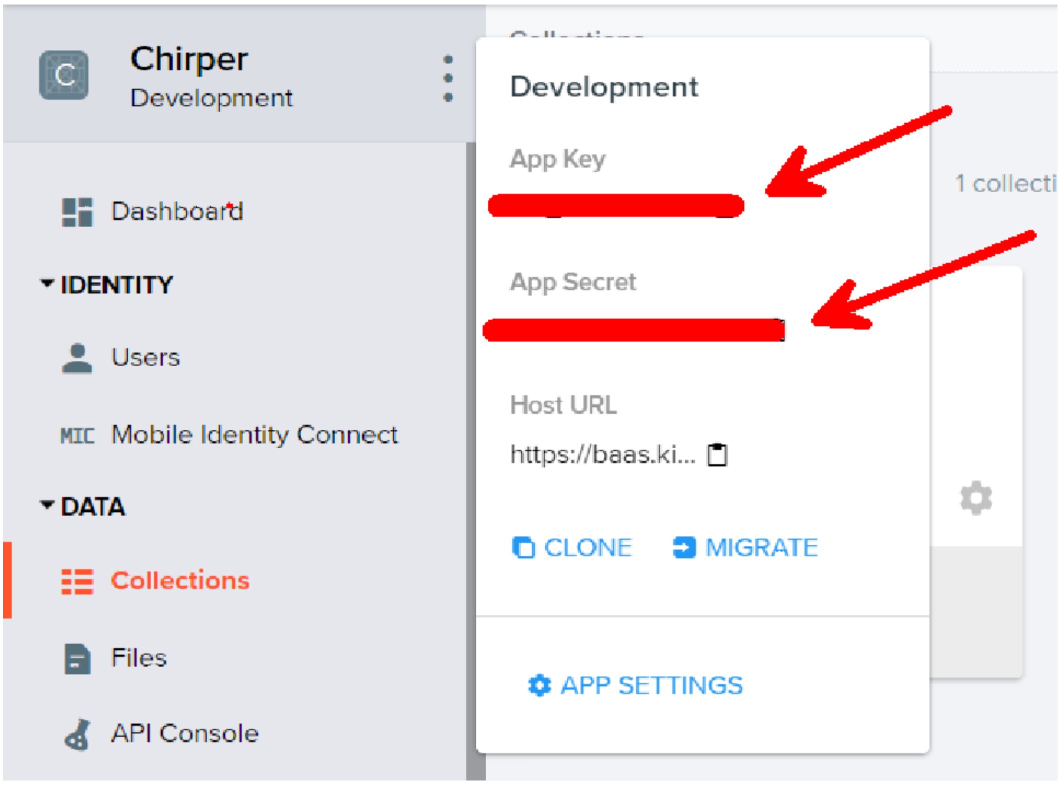
POST https://baas.kinvey.com/user/***app_key***/	
Request headers	Authorization: Basic base64(app_id:app_secret) Content-Type: application/json

Request body	{
	  "username": "testuser",
	  "password": "testuserpass890",
	  "subscriptions": []
	}

Response
201 Created
	{
	  "_id": "**59affdae3044bb86044a79bd**",
	  "username": "**testuser**",
	  "password": "**testuserpass890**",
	  "subscriptions": **[]**
	}

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.



Development

App Key

[redacted]

App Secret

[redacted]

Host URL

https://baas.ki...

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

User Login

POST https://baas.kinvey.com/user/***app_key***/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",
	"subscriptions": []
	"_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. **You may consider saving other useful information for later use.**

User Logout

POST https://baas.kinvey.com/user/***app_key***/_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 Chirps from subscriptions (Feed – sorted by post time, descending)

GET https://baas.kinvey.com/appdata/***app_key***/chirps?query={"author":{"$in": **[*subs*]**}}&sort={"_kmd.ect": 1}	
Request headers	Authorization: Kinvey authtoken

Response
200 OK
	[
	{
	"_id": "59ae7752e884417c04e803a4",
	"text": "yohooo",
	"_acl": {
	"creator": "59ae425c14ee505004862409"
	},
	"author": "vako",
	"_kmd": {
	"lmt": "2017-09-05T11:38:02.944Z",
	"ect": "2017-09-05T10:07:14.314Z"

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

	} },…
Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

Where :**subs** is an array of subscriptions (by **username**).

Create Chirp

POST https://baas.kinvey.com/appdata/***app_key***/chirps	
Request headers	Authorization: Kinvey authtoken Content-Type: application/json

Request body	{
	"text": "**MyTestChirp**",
	"author": "**testuser**"
	}

Response
201 Created
	{
	"_id": "59931398996ab5127d2a84d1",
	"author": "Kiril",
	"title": "Test Post",
	"description": "Test Post Description",
	"url": "https://en.wikipedia.org/wiki/Santorini",…
	}

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

Delete Chirp

DELETE https://baas.kinvey.com/appdata/***app_key***/chirps/***chirp_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": "" }

User Chirps (View all chirps by user, sorted by post time, descending)

GET https://baas.kinvey.com/appdata/***app_key***/chirps?query={"author":"***username***"}&sort={"_kmd.ect": 1}	
Request headers	Authorization: Kinvey authtoken

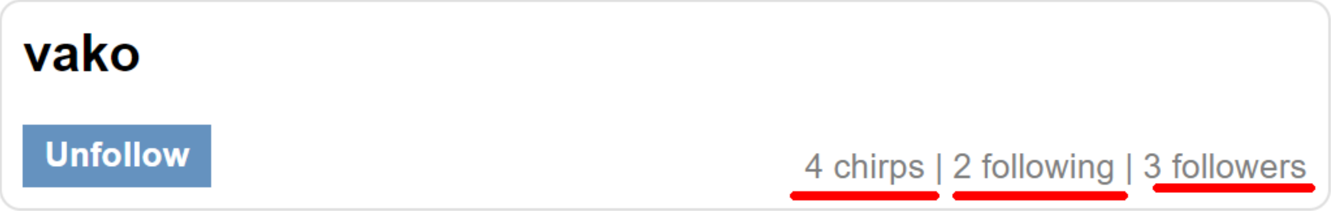
Response
200 OK
	[
	{
	"_id": "59b004cfdfab74540457702a",
	"text": "#testuserchirp",
	"author": "testuser",
	"_acl": {

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

	"creator": "59affdae3044bb86044a79bd"
	},
	"_kmd": {
	"lmt": "2017-09-06T14:23:11.069Z",
	"ect": "2017-09-06T14:23:11.069Z"
	}
	},..

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

The **next** 3 queries are needed to **display** profile **stats.**



Count Chirps

GET https://baas.kinvey.com/appdata/***app_key***/chirps?query={"author":"**username**"}	
Request headers	Authorization: Kinvey authtoken

Response
200 OK
	[
	{
	"_id": "59b004cfdfab74540457702a",
	"text": "#testuserchirp",
	"author": "testuser",
	"_acl": {
	"creator": "59affdae3044bb86044a79bd"
	},
	"_kmd": {
	"lmt": "2017-09-06T14:23:11.069Z",
	"ect": "2017-09-06T14:23:11.069Z"
	}
	},..

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

This query will return **all** the chirps by **given** user. **Count** them.

Count Following (All users that the currently logged is subscribed to)

GET https://baas.kinvey.com/user/***app_key***/?query={"username":"**username**"}	
Request headers	Authorization: Kinvey authtoken
Response 200 OK	[{ "_id": "59afe6ad68c53f4a045534df",

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

	"subscriptions": [
	"vako"
	],
	"username": "Kiril98",
	"_acl": {
	"creator": "59afe6ad68c53f4a045534df"
	},
	"_kmd": {
	"lmt": "2017-09-06T12:37:47.242Z",
	"ect": "2017-09-06T12:14:37.861Z"
	}
	}
	]

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

This query will return the **needed** user. The **count** of the people that the user is **following** is the **length** of the subscriptions array.

Count Followers (All users that are subscribed to the currently logged)

GET https://baas.kinvey.com/user/***app_key***/?query={"subscriptions":"***username***"}	
Request headers	Authorization: Kinvey authtoken

Response
200 OK
	[
	{
	"_id": "59afe70614ee5050048cecfb",
	"subscriptions": [
	"Kiril98",
	"SoftUni"
	],
	"username": "kirilvk1",
	"_acl": {
	"creator": "59afe70614ee5050048cecfb"
	},
	"_kmd": {
	"lmt": "2017-09-06T12:16:19.636Z",
	"ect": "2017-09-06T12:16:06.222Z"
	}
	},
	{
	"_id": "59affdae3044bb86044a79bd",
	"subscriptions": [
	"Kiril98",
	"Tacito"
	],…
	]

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

The following query will return all **users** that are **subscribed** to given username. You need the length of this array.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

Discover Page

GET https://baas.kinvey.com/user/***app_key***/	
Request headers	Authorization: Kinvey authtoken

Response
200 OK
	[
	{
	"_id": "59ae425c14ee505004862409",
	"subscriptions": [
	"Tacito",
	"SoftUni"
	],
	"username": "vako",
	"_acl": {
	"creator": "59ae425c14ee505004862409"
	},
	"_kmd": {
	"lmt": "2017-09-05T17:14:56.960Z",
	"ect": "2017-09-05T06:21:16.793Z"
	}
	},
	{
	"_id": "59ae864f603d08cc6fedf3ab",
	"username": "Tacito",
	"_acl": {
	"creator": "59ae864f603d08cc6fedf3ab"
	},
	"subscriptions": [],
	"_kmd": {
	"lmt": "2017-09-05T11:42:53.473Z",
	"ect": "2017-09-05T11:11:11.629Z"
	}
	},..

Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

This will return all users. In the discover page you should **display** all users **except** the currently logged.

Follow

PUT https://baas.kinvey.com/user/***app_key***/***user_id***	
Request headers	Authorization: Kinvey authtoken Content-Type: application/json

Request body	{
	"subscriptions": [
	"testuser",
	"vako",
	"**Tacito**"
	]
	}

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

Response 200 OK	{ "subscriptions": ["testuser", "vako", "**Tacito**"], "_id": "59afe6ad68c53f4a045534df", "username": "Kiril98",…
Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

The PUT query does **not** need to include the **whole** user object. Submit **only** the modified subscriptions array.

Unfollow

PUT https://baas.kinvey.com/user/***app_key***/***user_id***	
Request headers	Authorization: Kinvey authtoken Content-Type: application/json

Request body	{
	"subscriptions": [
	"testuser",
	"vako"
	]
	}

Response 200 OK	{ "subscriptions": ["testuser", "vako"], "_id": "59afe6ad68c53f4a045534df", "username": "Kiril98",…
Error response 401 Unauthorized	{ "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" }

The **same** query as "**Follow**". Only this time we **remove** the wanted username and submit the modified subscriptions array.

Problem 3. Chirper–HTML and CSS

You are given the Web design of the **Chirper** 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 **Mustache 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.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

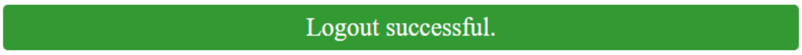
Problem 4. Chirper Client-Side Web Application

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

Notifications

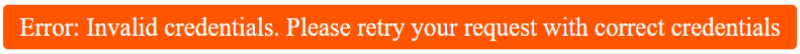
The application should notify the users about the result of their actions.

- 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.



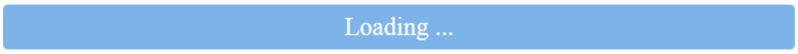
Logout successful.

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



Error: Invalid credentials. Please retry your request with correct credentials

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

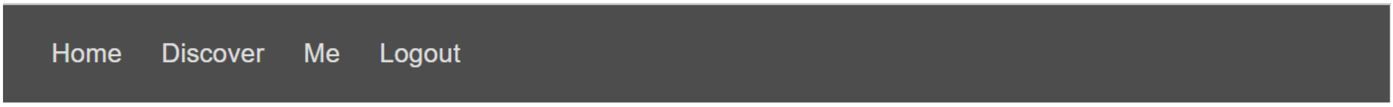


Loading ...

Navigation System (5 pts)

Implement a **navigation system** 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** post 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 **navbar** should be visible **only** for logged in users. Anonymous users can **only** view the **sign in/register** section.

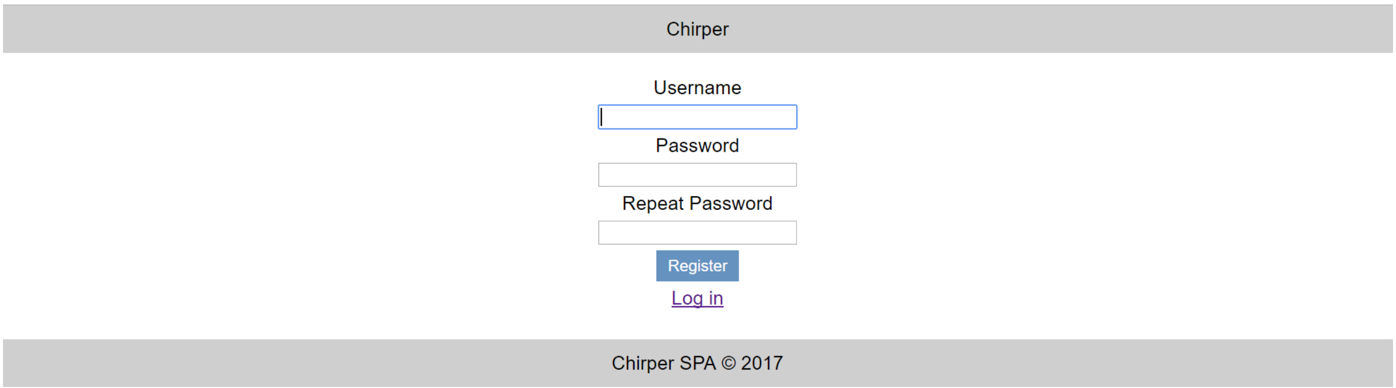


Register User Screen (10 pts)

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 **feed page** with all **chirps by followers** should be shown.
- You **need** to validate the **input**. A username **should** be a string with at **least** 5 characters **long**. Passwords **input** fields shouldn’t be **empty**. Both passwords **should** 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**.
- That **includes** the subscriptions array. Since local/session storage **only** keeps **strings** the array must be **stringified** or **kept** as a variable and **updated** on every init.
- Clear **all** input fields after **successful** register.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.



Chirper

Username

Password

Repeat Password

Register

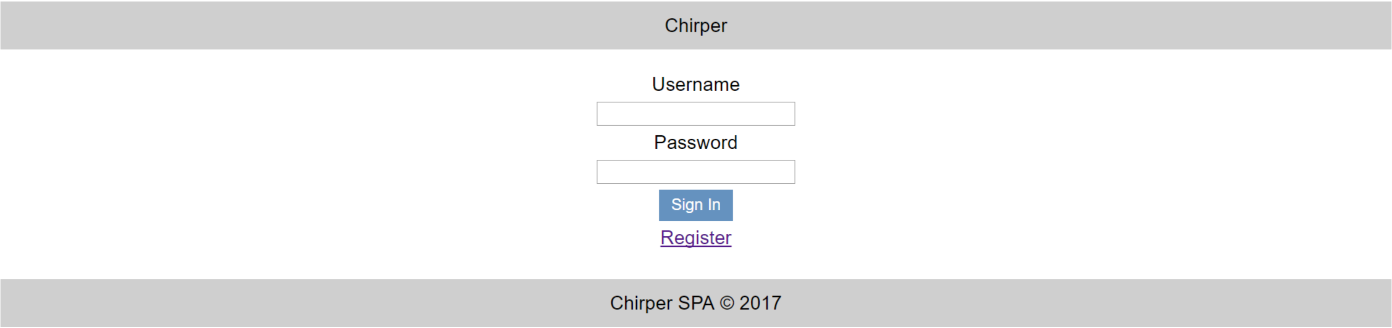
Log in

Chirper SPA © 2017

Login User Screen (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 and the **feed page** with all **chirps by followers** should be shown.
- 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**.
- That **includes** the subscriptions array. Since local/session storage **only** keeps **strings** the array must be **stringified** or **kept** as a variable and **updated** on every init.
- Clear **all** input fields after **successful** login.



Chirper

Username

Password

Sign In

Register

Chirper SPA © 2017

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.

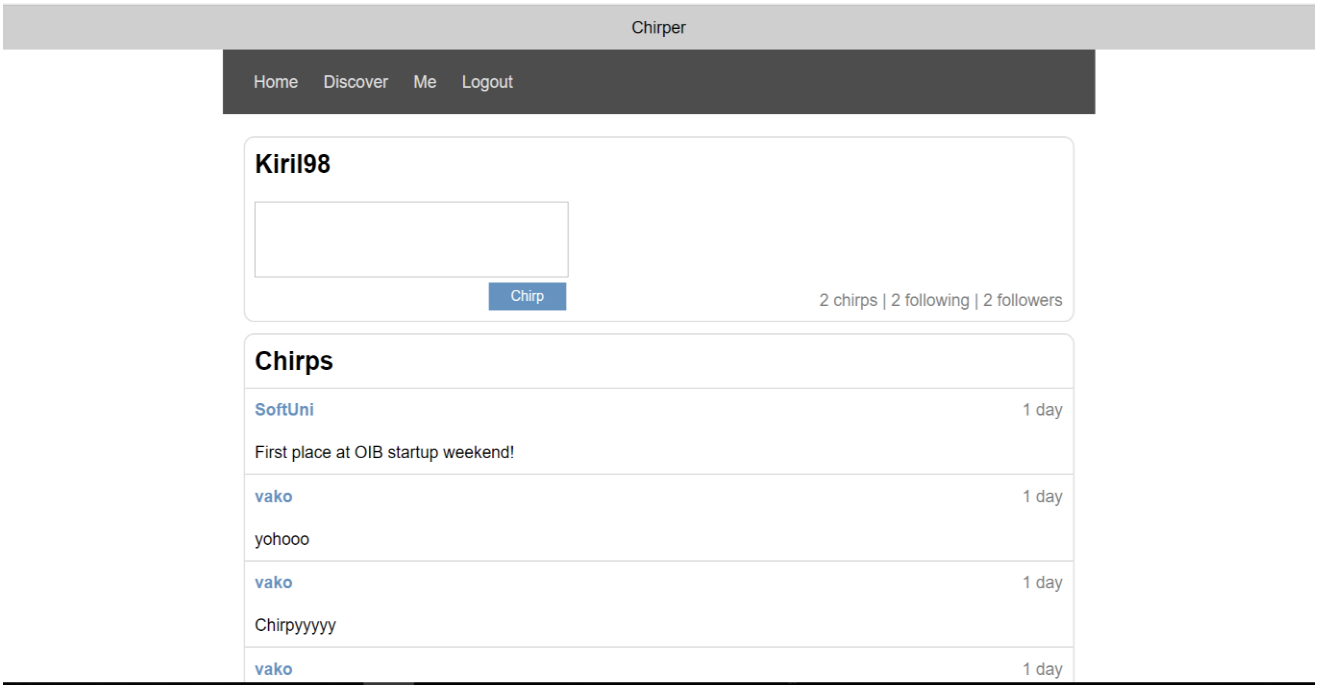
Main Feed/Home Screen (20 pts)

Successfully logged users should be welcomed by the **feed** screen which can be accessed **after** clicking the **[Home]** link in the **navbar**. The feed **screen** contains all chirps from **subscriptions or people that the user is following** (sorted by time posted in descending). It also contains the **create** a chirp **form**.

- The **chirps** should be listed in the **format** as shown in the Web design (see the screenshot below).
- Each **chirp** has **text**, **author** and information about **when** it was **created**. For the **time created** information you will use a helper function given **below**. It requires a **parameter** which is the entity creation date of the post (or **_kmd.ect** property of the post object).
- When the **follower username** link is clicked, your app should **redirect** to the individual feed page of the user.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

- The feed screen should also contain **information** about the **count** all of all chirps that the user has **posted**, number of people that the user is **following**, and number of **followers** that the user has.
- In case of **error** (e.g. Internet connection lost), an error message should be displayed.
- In case of **no chirps**, display ‘No chirps in database’.



Chirper

Home Discover Me Logout

Kiril98

Chirp

2 chirps | 2 following | 2 followers

Chirps

SoftUni 1 day

First place at OIB startup weekend!

vako 1 day

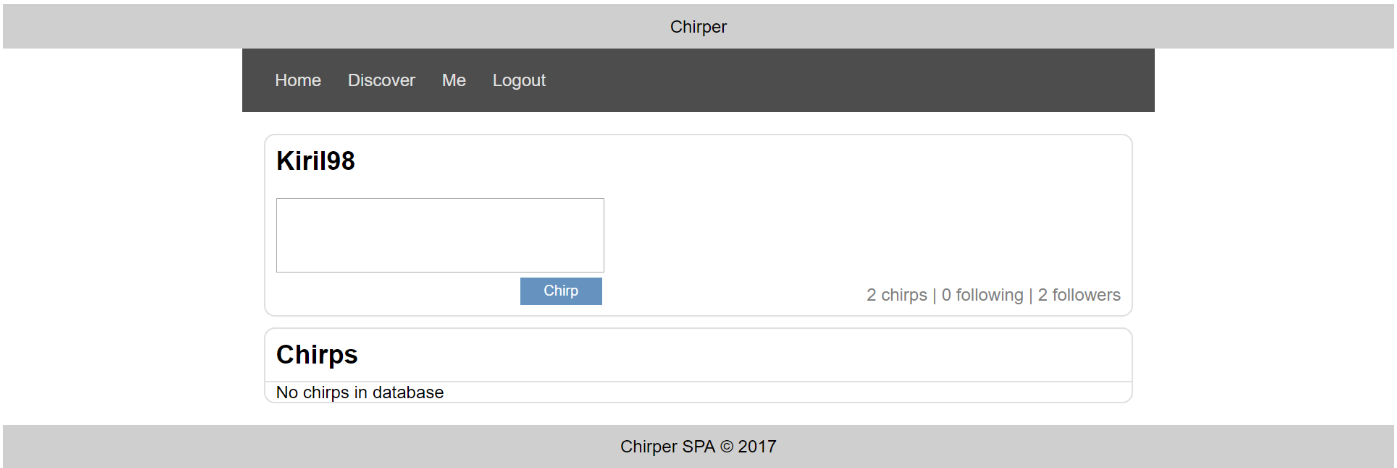
yohooo

vako 1 day

Chirpyyyyy

vako 1 day

If the user isn’t following anyone, display the following:



Chirper

Home Discover Me Logout

Kiril98

Chirp

2 chirps | 0 following | 2 followers

Chirps

No chirps in database

Chirper SPA © 2017

- Format the entity creation **date** of a **chirp** as shown in the screenshot above. You may **use** the following helper function:

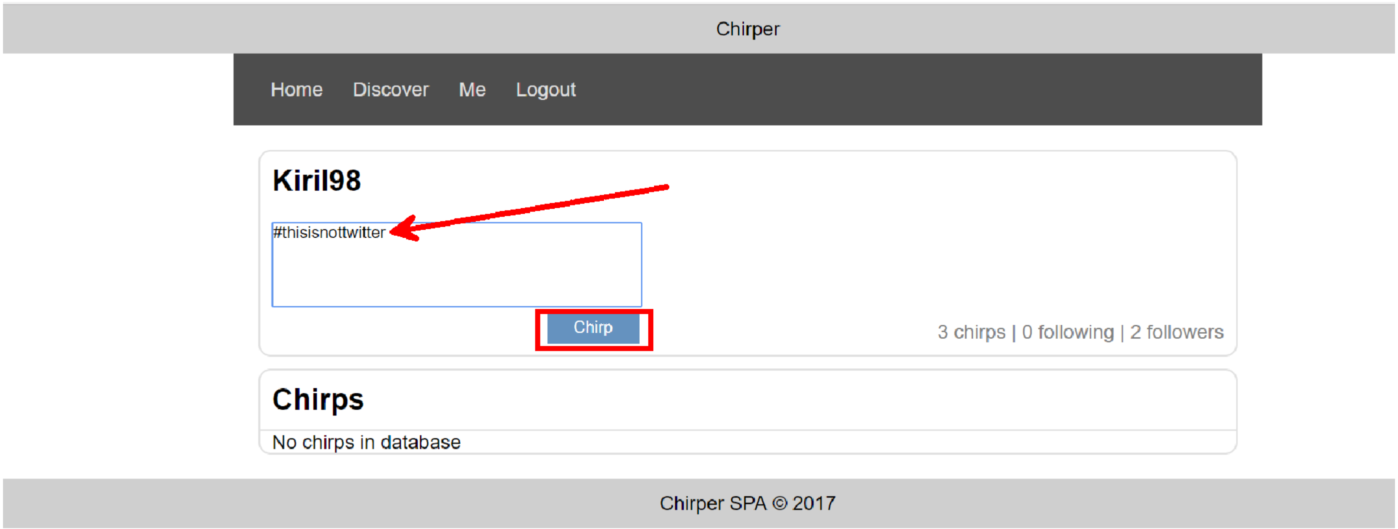
```
function calcTime(dateIsoFormat) {
    let diff = new Date - (new Date(dateIsoFormat));
    diff = Math.floor(diff / 60000);
    if (diff < 1) return 'less than a minute';
    if (diff < 60) return diff + ' minute' + pluralize(diff);
    diff = Math.floor(diff / 60);
    if (diff < 24) return diff + ' hour' + pluralize(diff);
    diff = Math.floor(diff / 24);
    if (diff < 30) return diff + ' day' + pluralize(diff);
    diff = Math.floor(diff / 30);
    if (diff < 12) return diff + ' month' + pluralize(diff);
    diff = Math.floor(diff / 12);
    return diff + ' year' + pluralize(diff);
    function pluralize(value) {
        if (value !== 1) return 's';
        else return '';
    }
}
```

- Ensure you handle properly all HTML **special characters**, e.g. the message text could hold "*Hi, <peter>*".

Post Chirp (10 pts)

Logged in users can **post** chirps. Clicking the **[Chirp]** button should directly **add** a chirp into the **kinvey** database. A chirp has **text** and **author**.

- After a **successful** chirp creation, a notification message "Chirp published." should be displayed and the **user feed** should be shown (the screen with all chirps posted by the **currently logged** user).
- A chirp text shouldn't be **empty** and shouldn't contain more than **150 symbols**.
- Clear the **create** chirp input **field** after **successful** creation.
- Ensure you handle properly all HTML **special characters**, e.g. the message text could hold "*Hi, <peter>*".



Kiril98

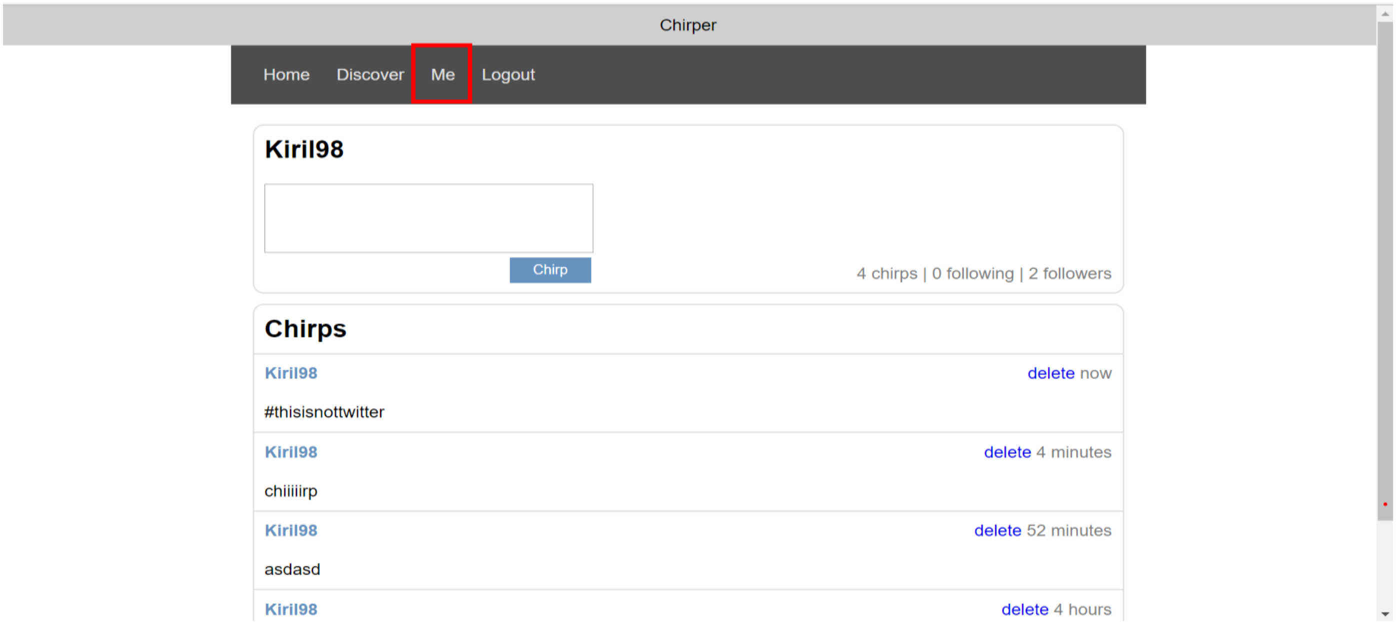
#thisisnottwitter

3 chirps | 0 following | 2 followers

Chirps

No chirps in database

After **successful** creation the **user feed** should be displayed. It can also be accessed by clicking the [**Me**] link in the **navbar**:



Kiril98

4 chirps | 0 following | 2 followers

Chirps

Kiril98 delete now

#thisisnottwitter

Kiril98 delete 4 minutes

chiiiirp

Kiril98 delete 52 minutes

asdasd

Kiril98 delete 4 hours

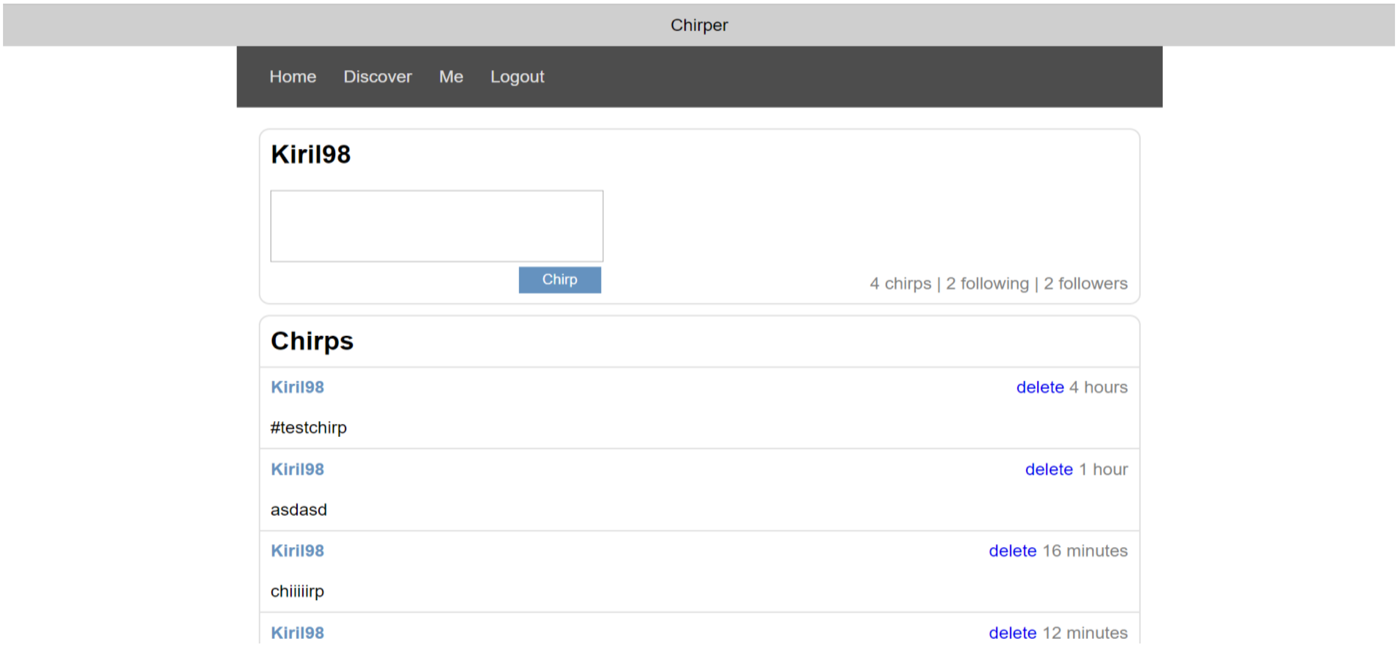
© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

User Feed (15 pts)

Logged in users can view **individual** feeds (profile page). The **currently logged** user has an individual feed page where he can view his own chirps, **create** a chirp and delete his own **chirps**. Each logged in user can **access** their own feed by clicking the [**Me**] link.

- After a **successful** chirp creation, a notification message "Chirp published." should be displayed and the **user feed** should be shown again.
- The **user feed** screen should also contain **information** about the **count** all of all chirps that the user has **posted**, number of people that the user is **following**, and number of **followers** that the user has.
- All chirps are listed in **descending** by time **creation**.
- In case of **no chirps**, display 'No chirps in database'.
- In case of **error** (e.g. Internet connection lost), an error message should be displayed.

My own feed page:



Chirper

Kiril98

4 chirps | 2 following | 2 followers

Chirps

Kiril98 delete 4 hours

#testchirp

Kiril98 delete 1 hour

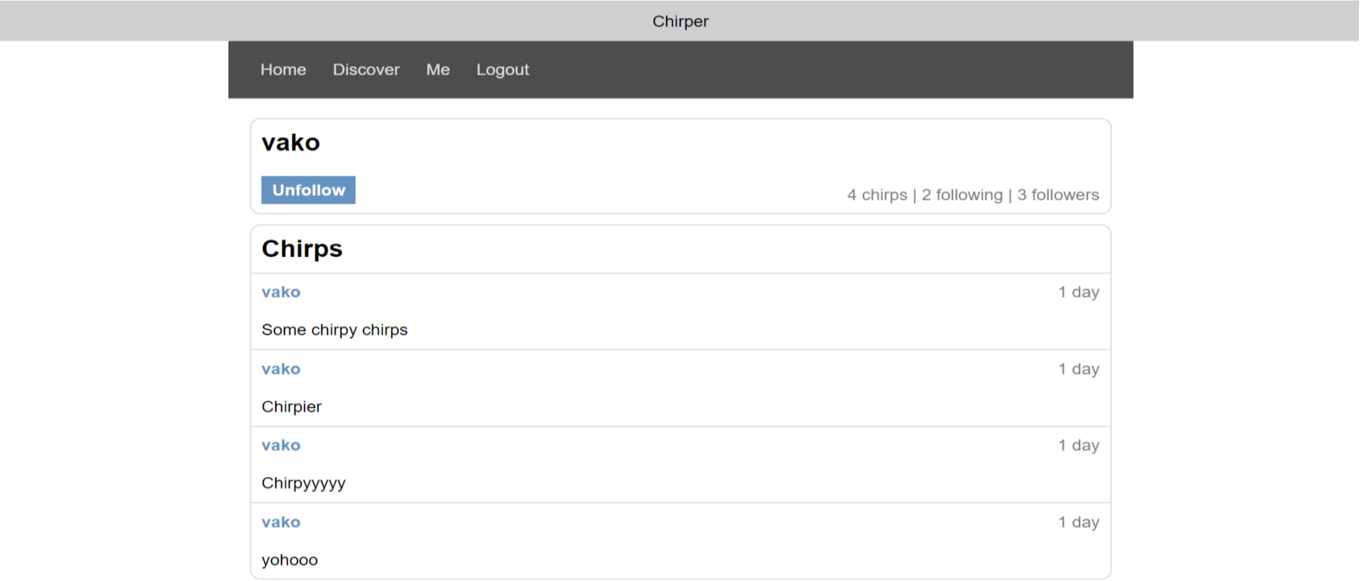
asdasd

Kiril98 delete 16 minutes

chiiiiirp

Kiril98 delete 12 minutes

Another user's feed page:



Chirper

vako

4 chirps | 2 following | 3 followers

Chirps

vako 1 day

Some chirpy chirps

vako 1 day

Chirpier

vako 1 day

Chirpyyyyy

vako 1 day

yohooo

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.

The **button** on the top left of the screen can be either **Follow** or **Unfollow** depending on if the user's username is in the **subscriptions** array.

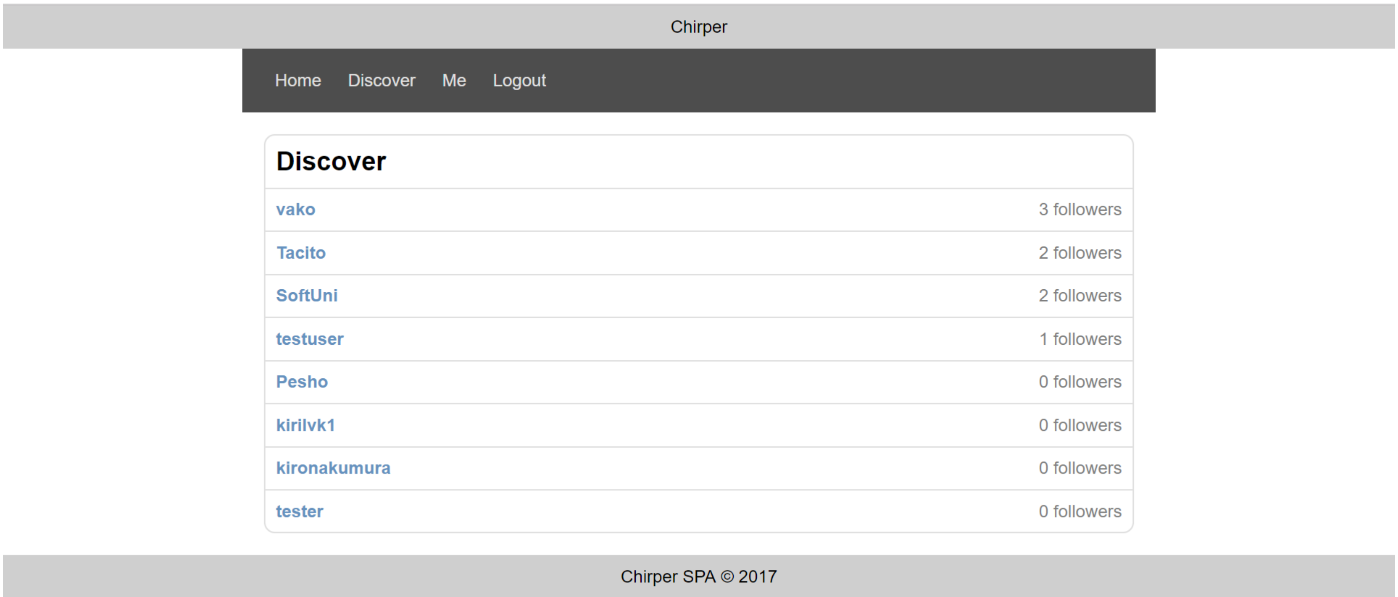
Delete Chirp (5 pts)

Authors **should** be able to **delete** their **own** chirps by clicking the [**delete**] button on each individual chirp.

- After **successful** post delete a notification message "**Chirp deleted.**" should be displayed and the **user feed** should be shown again.
- 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**.

Discover People Screen (15 pts)

In order to **follow**/**unfollow** people we must first **find** them. Implement a page where all users are **listed** and can be followed. By clicking the [**Discover**] link from the navigation the **following** screen should be displayed:



Discover

vako	3 followers
Tacito	2 followers
SoftUni	2 followers
testuser	1 followers
Pesho	0 followers
kirilvk1	0 followers
kironakumura	0 followers
tester	0 followers

Chirper SPA © 2017

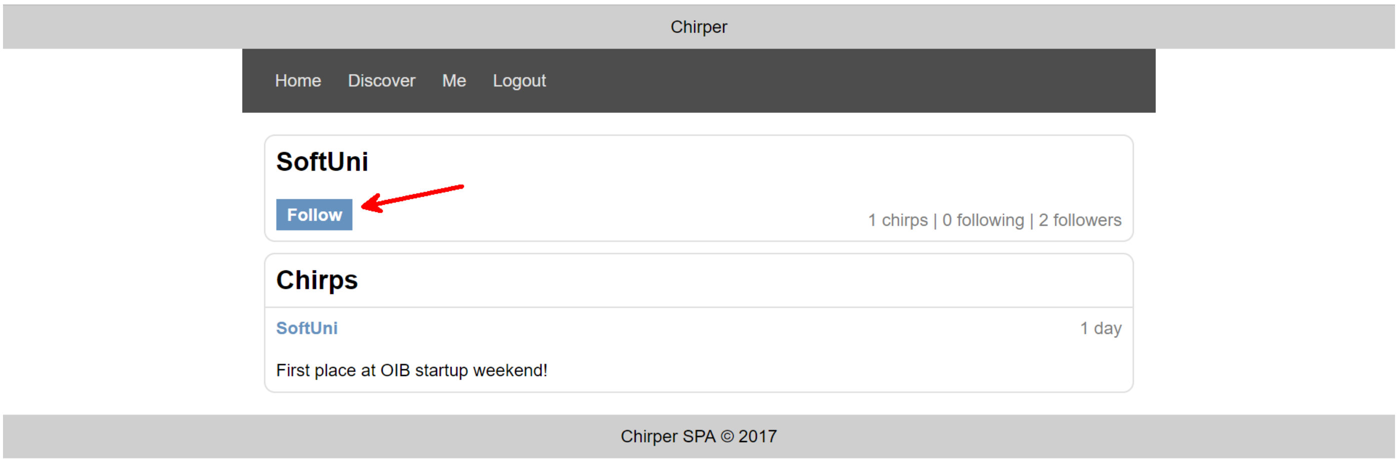
- These are all of the **users** in the database **except** the currently logged.
- Clicking on the username should **redirect** to the user's feed page
- Next to **each** username is the number of **followers** that the user has.
- In case of **error** (e.g. Internet connection lost / unauthorized request / missing message), an error message should be displayed.

Follow User (5 pts)

Logged in users can **follow** other users by clicking on the [**Follow**] link on each user **feed page**.

- After a **successful** following, a notification message "**Subscribed to {username}**" should be **displayed** and the **user feed** should be shown again with **updated** stats.
- The **username** of the **newly** followed user should be added in the **subscriptions** array of the currently logged in user.
- In case of **error** (e.g. Internet connection lost / unauthorized request / missing message), an error message should be displayed.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.



Chirper

SoftUni

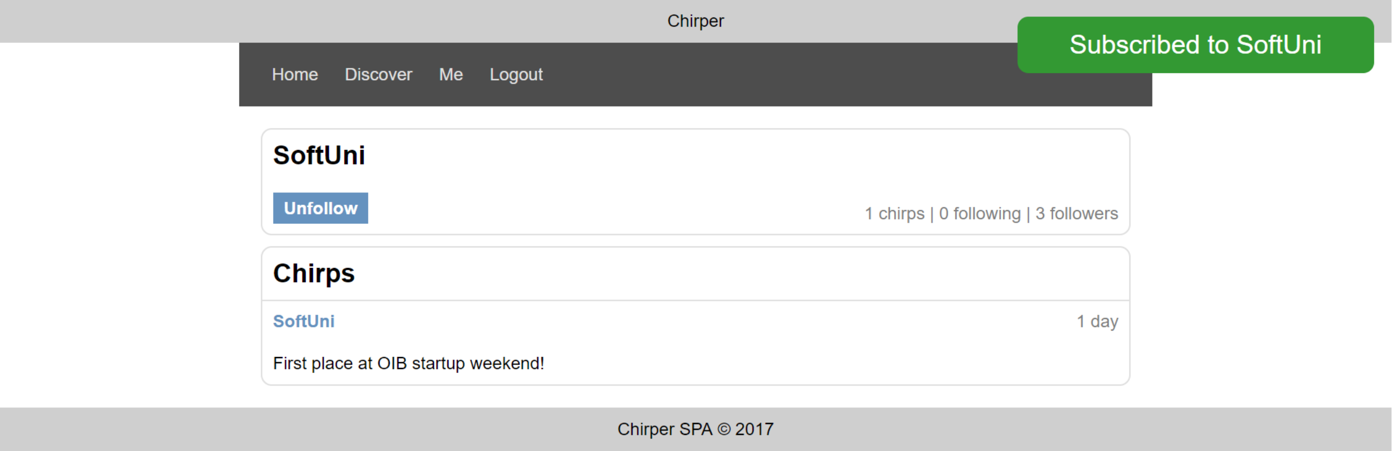
1 chirps | 0 following | 2 followers

Chirps

SoftUni 1 day

First place at OIB startup weekend!

Chirper SPA © 2017



Chirper

Subscribed to SoftUni

SoftUni

1 chirps | 0 following | 3 followers

Chirps

SoftUni 1 day

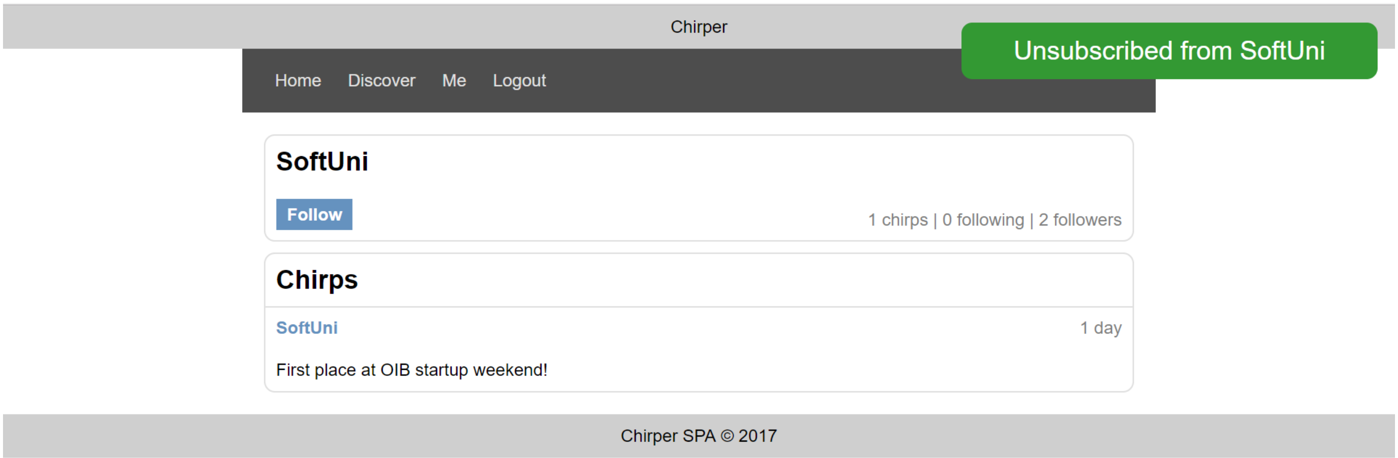
First place at OIB startup weekend!

Chirper SPA © 2017

Unfollow User (5 pts)

Logged in users can **unfollow** other users by clicking on the [**Unfollow**] link on each user **feed page**.

- After a **successful** unfollow, a notification message "**Unsubscribed to {username}**" should be **displayed** and the **user feed** should be shown again with **updated** stats.
- The **username** of the **newly** followed user should be **removed** from the **subscriptions** array of the currently logged in user.
- In case of **error** (e.g. Internet connection lost / unauthorized request / missing message), an error message should be displayed.



Chirper

Unsubscribed from SoftUni

SoftUni

1 chirps | 0 following | 2 followers

Chirps

SoftUni 1 day

First place at OIB startup weekend!

Chirper SPA © 2017

Problem 5. Subtmitting Your Solution

Place in a ZIP file your project folder. Exclude the **`node_modules`** folder. Upload the archive to the Judge.

© Software University Foundation (softuni.org). This work is licensed under the CC-BY-NC-SA license.