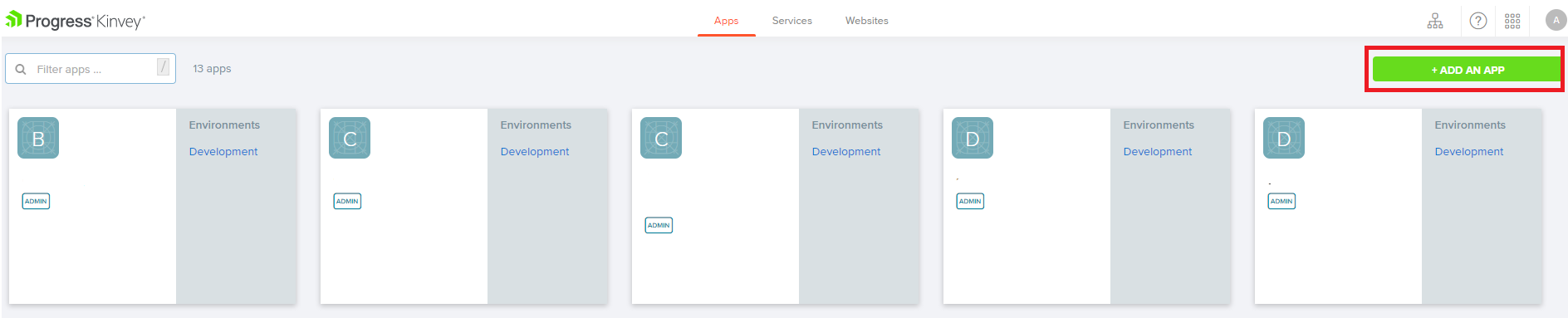
# JS Applications Exam - DonUni SPA

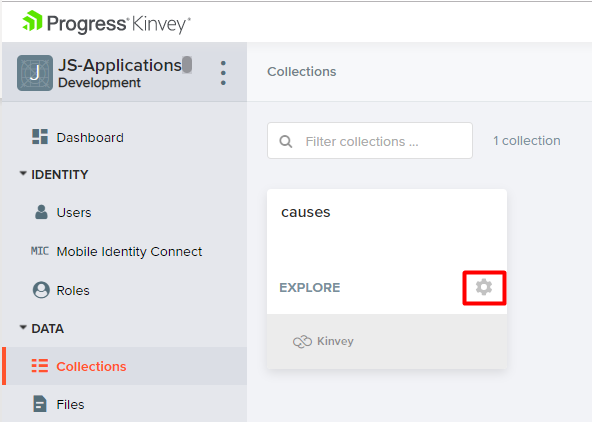
You are assigned to implement a **Web application** (SPA) using HTML5, JavaScript, AJAX, REST and JSON with cloud-based backend (Kinvey). Using libraries like **jQuery**, **Handlebars** and **Sammy** is allowed. The app keeps **users** and **causes**. **Guests** should be able to **register** and **login**. Logged-in users should be able to view **all causes (dashboard)**, **create and view** **cause** and **logout**. Logged-in users should also be able to **edit** or **delete** the causes **they have created**.

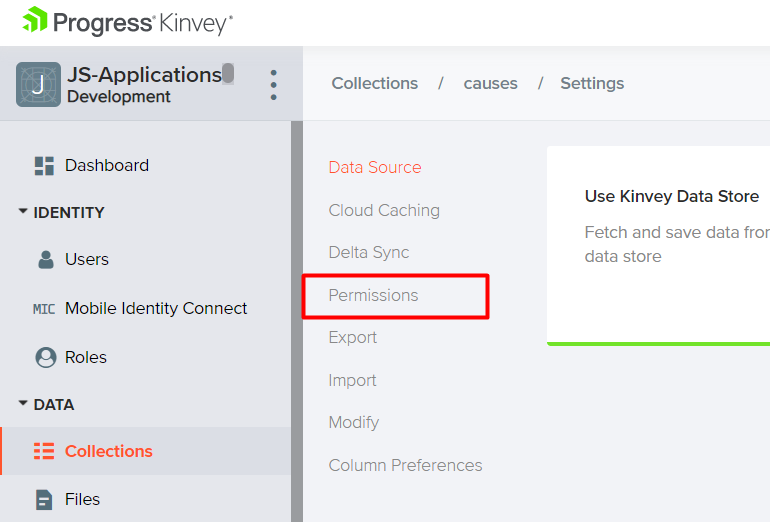
## Create a Kinvey REST Service

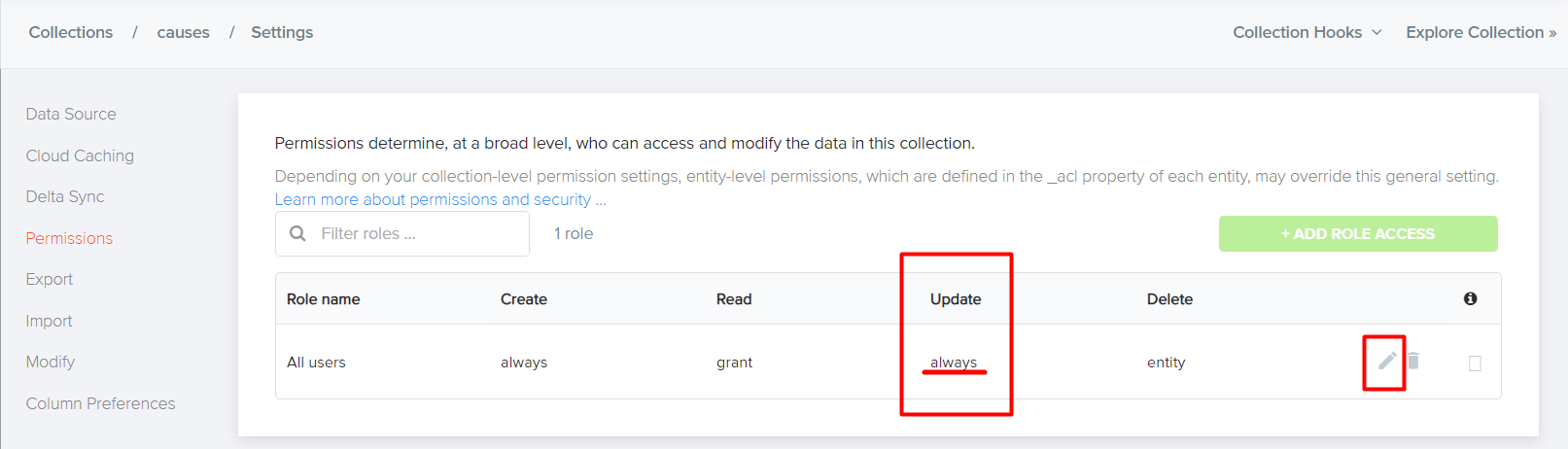
Register at **Kinvey.com** (<https://console.kinvey.com/login>)and create application to keep your data in the cloud.



Create a collection called **causes.** Each **cause** has a **donors**, **cause**, **pictureUrl**, **neededFunds**, **description** and **collectedFunds**.



In order to perform a successful donation to other person's cause, you should change the default permissions on the current kinvey collection, like the example below.



Here some example with **already created cause**:



## 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 end points:

### User Registration (Sign-Up)

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

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/user/***app\_key***/](https://baas.kinvey.com/user/app_key/) | |
| **Request headers** | Authorization: Basic base64(app\_key:app\_secret)  Content-Type: application/json |
| **Request body** | {  "username": "testuser",  "password": "weakpass123"  } |
| **Response**  **201 Created** | {  "\_id": "59930c78a743e20c7d3fca77",  "username": "testuser",  "password": "weakpass123"  } |
| **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": "" } |

### User Login (Sign-In)

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

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/user/***app\_key***/login](https://baas.kinvey.com/user/app_key/login) | |
| **Request headers** | Authorization: Basic base64(user\_username:user\_password)  Content-Type: application/json |
| **Request body** | {  "username": "testuser",  "password": "weakpass123"  } |
| **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": "" } |

### User Logout

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

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

### Dashboard (List All Causes)

|  |  |
| --- | --- |
| **GET** [https://baas.kinvey.com/appdata/***app\_key***/causes](https://baas.kinvey.com/appdata/app_key/causes) | |
| **Request headers** | Authorization: Kinvey authtoken |
| **Response**  **200 OK** | [{  "cause": "...",  "donors": [...],  "pictureUrl": "https://....",  "neededFunds": ...,  "description": ...,  "collectedFunds": ...,  "\_acl":  {  "creator":"5bfd4674682ae23931b4f91c"  },  "\_kmd":  {  "lmt":"2018-11-28T15:25:24.521Z",  "ect":"2018-11-28T14:55:00.958Z"  }  }, …] |
| **Error response**  **401 Unauthorized** | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Create Cause

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/appdata/***app\_key***/causes](https://baas.kinvey.com/appdata/app_key/causes) | |
| **Request headers** | Authorization: Kinvey authtoken  Content-Type: application/json |
| **Request body** | {  "cause": "...",  "donors": [...],  "pictureUrl": "https://....",  "neededFunds": ...,  "description": ...,  "collectedFunds": ...,  } |
| **Response**  **201 Created** | {  "cause": "...",  "donors": [...],  "pictureUrl": "https://....",  "neededFunds": ...,  "description": ...,  "collectedFunds": ...,  "\_acl": {  "creator": "5bfd4674682ae23931b4f91c"  },  "\_kmd": {  "lmt": "2018-11-28T15:39:58.801Z",  "ect": "2018-11-28T15:39:58.801Z"  },  "\_id": "5bfeb6ce682ae23931bf7d26"  } |
| **Error response**  **401 Unauthorized** | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Edit Cause

|  |  |
| --- | --- |
| **PUT** [https://baas.kinvey.com/appdata/***app\_key***/causes/cause***\_id***](https://baas.kinvey.com/appdata/app_key/causes/cause_id) | |
| **Request headers** | Authorization: Kinvey authtoken  Content-Type: application/json |
| **Request body** | {  "cause": "...",  "donors": [...],  "pictureUrl": "https://....",  "neededFunds": ...,  "description": ...,  "collectedFunds": ...,  } |
| **Response**  **200 Ok** | {  "\_id": "59931398996ab5127d2a84d1",  "cause": "...",  "donors": [...],  "pictureUrl": "https://....",  "neededFunds": ...,  "description": ...,  "collectedFunds": ...,  ...  } |
| **Error response**  **401 Unauthorized** | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Delete Cause

|  |  |
| --- | --- |
| **DELETE** [https://baas.kinvey.com/appdata/](https://baas.kinvey.com/appdata/app_id/causes/cause_id)***[app\_id](https://baas.kinvey.com/appdata/app_id/causes/cause_id)***[/causes/cause](https://baas.kinvey.com/appdata/app_id/causes/cause_id)***[\_id](https://baas.kinvey.com/appdata/app_id/causes/cause_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": "" } |

## DonUni - HTML and CSS

You have been given the web design of the application as **HTML** + **CSS** files.

Initially all views and forms are shown by the HTML. Initially, all views and forms are shown by the HTML. Your application must display the current views depends on the routing URL. You may render the views/forms/components with **Handlebars**.

Important  
Don't change the elements' **class names** and **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.

## DonUni - Client-Side Web Application

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

### Navigation Bar (Header) (5)

Navigation links should correctly change the current page (view).

Clicking on the links in the **NavBar** should display the view behind the link (views are represented as sections in the HTML code).

Your application must display the views it needs to display.

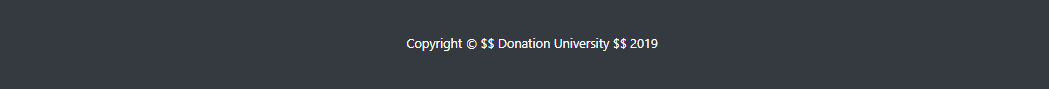
The **Logged-in** **user** navbar should contain the following elements: **[**DonUni], **[**$$$$$$$$$$$$$**]**, and **[**Logout ({username})**]**.



The guest users navbar should contain the following elements: **[DonUni], [Login] and [Register]**



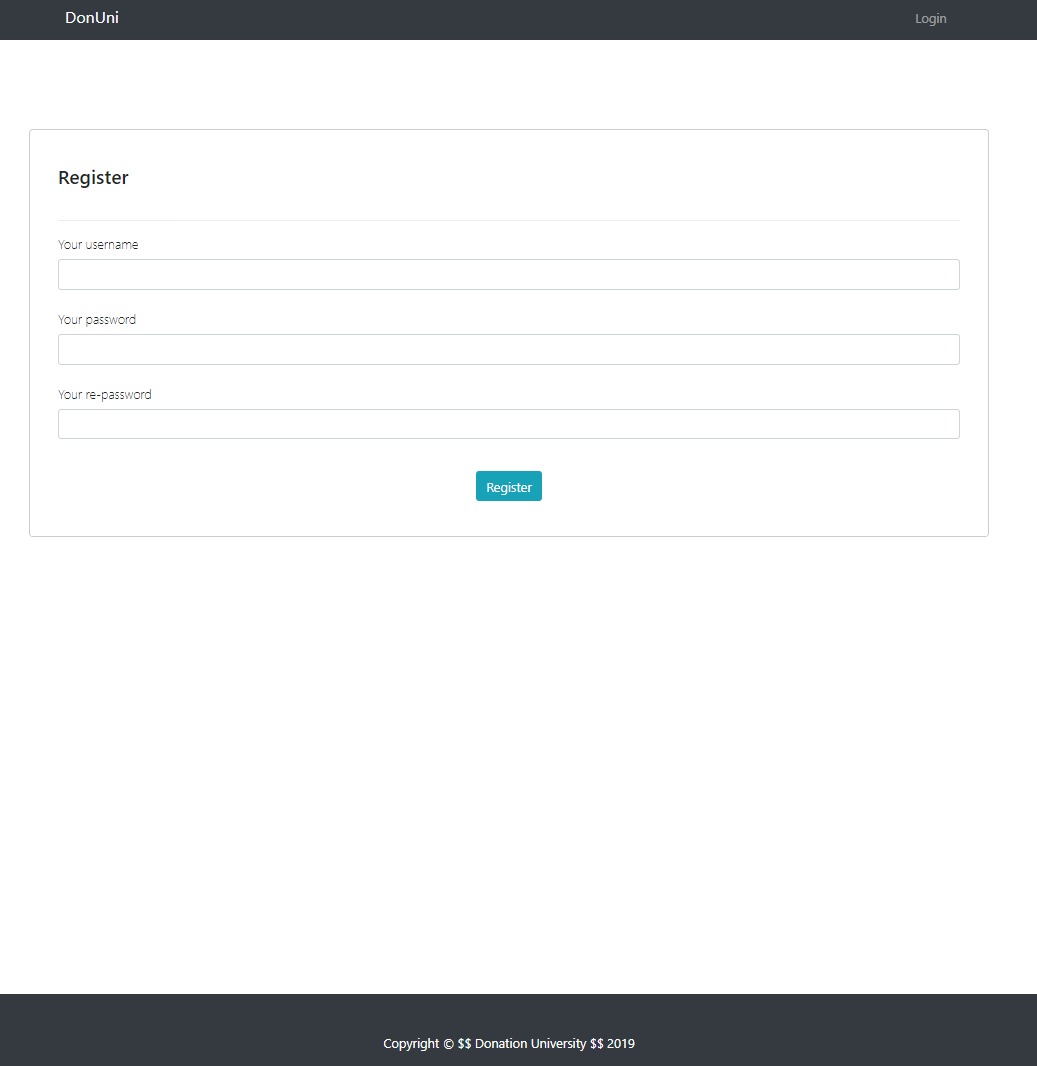
### Footer



### Register Page (5)

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

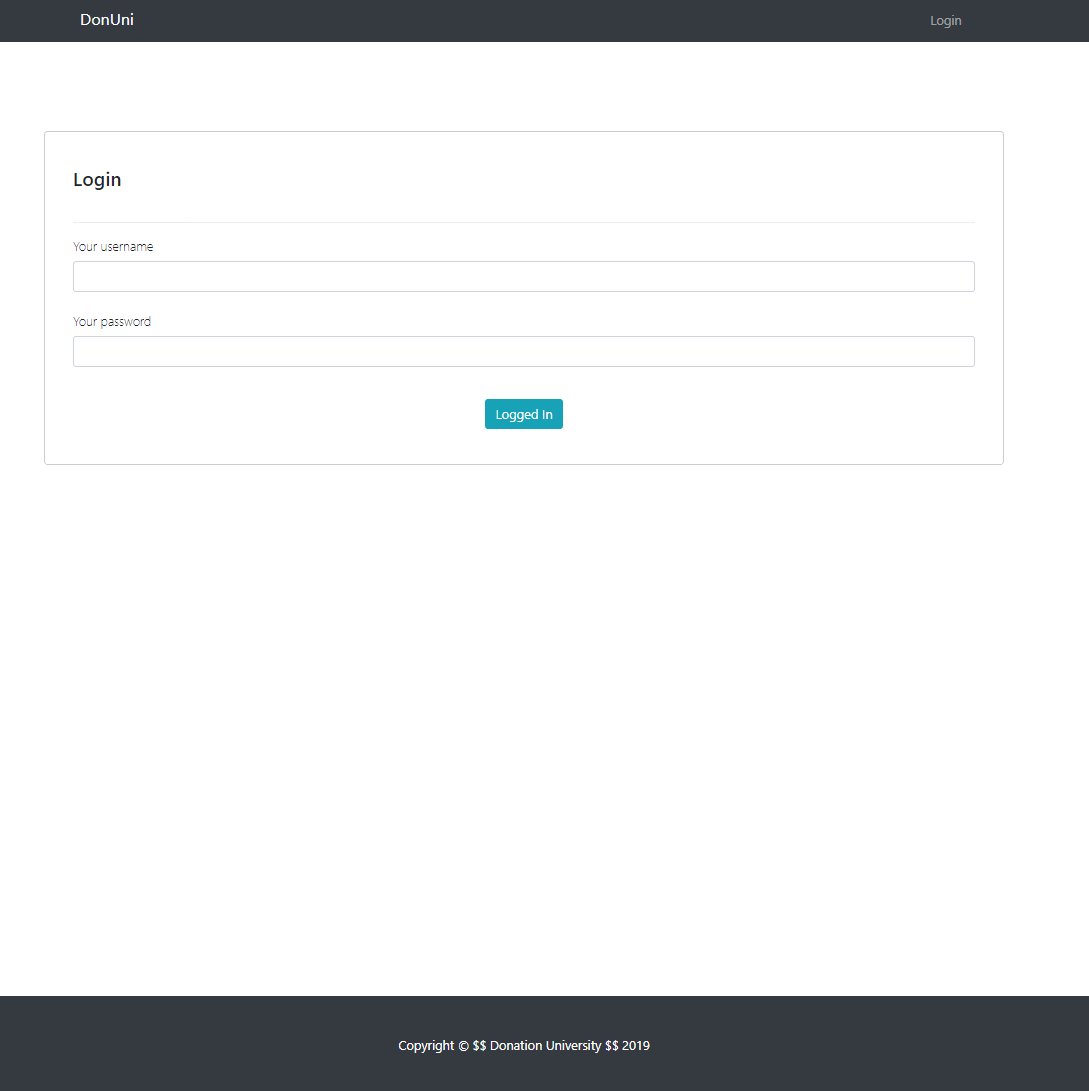
* The following validations should be made:
  + The **username** and **password** must be non-empty string
  + The **re-password** should be **equal to the password**
* After a **successful registration**, **home** **page** should be displayed **with the right navbar**.
* Keep the user session data in the browser's **session or local storage**.



### Login Page (5)

By given **username** and **password,** the app should login an existing user.

* After a **successful login**, a **home** **page** should be displayed with the **right** **navbar**.
* Keep the user session data in the browser's **session or local storage**.



### Logout (5)

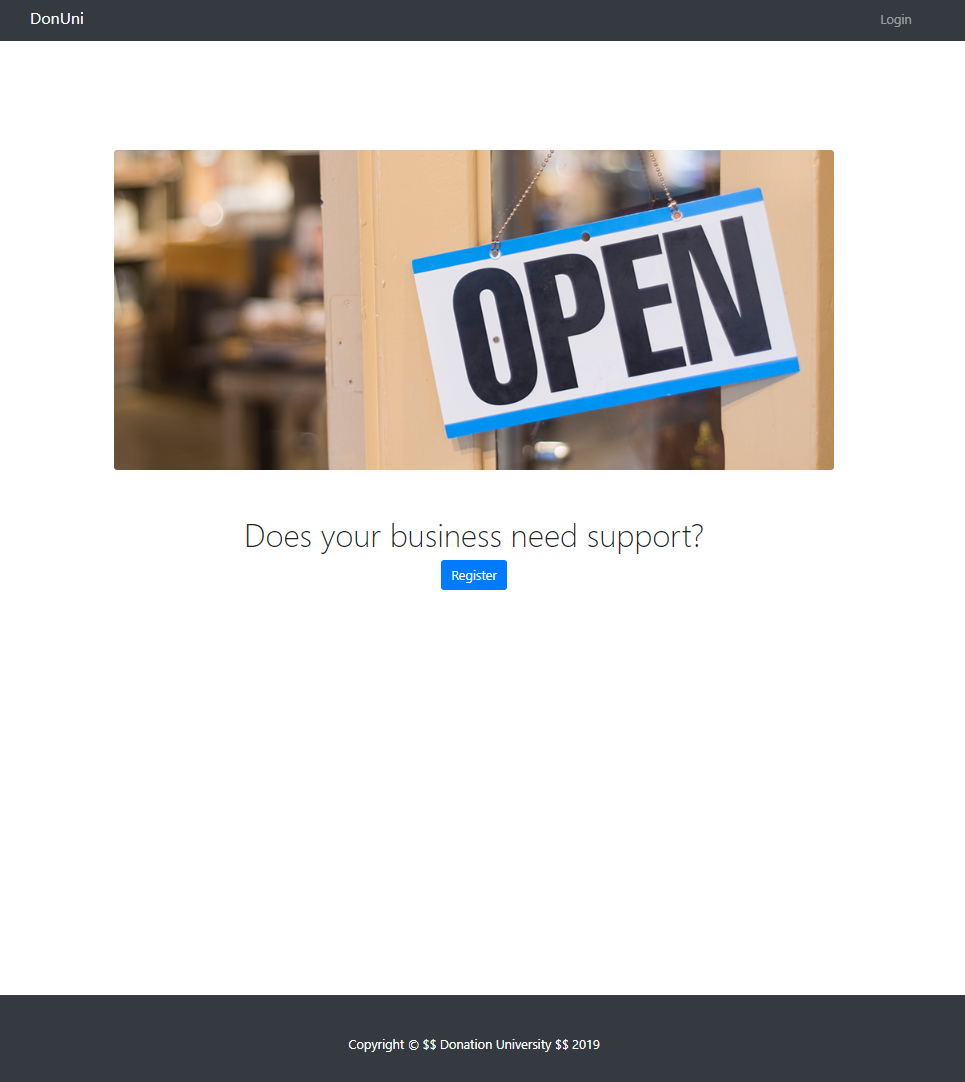
Successfully logged in users should be able to **logout** from the app.

* After a **successful** logout the **home page for guest users** should be shown
* The **"logout" REST service** at the back-end **must** be called at logout
* All local information in the browser (**user session data**) about the current user **should be deleted**

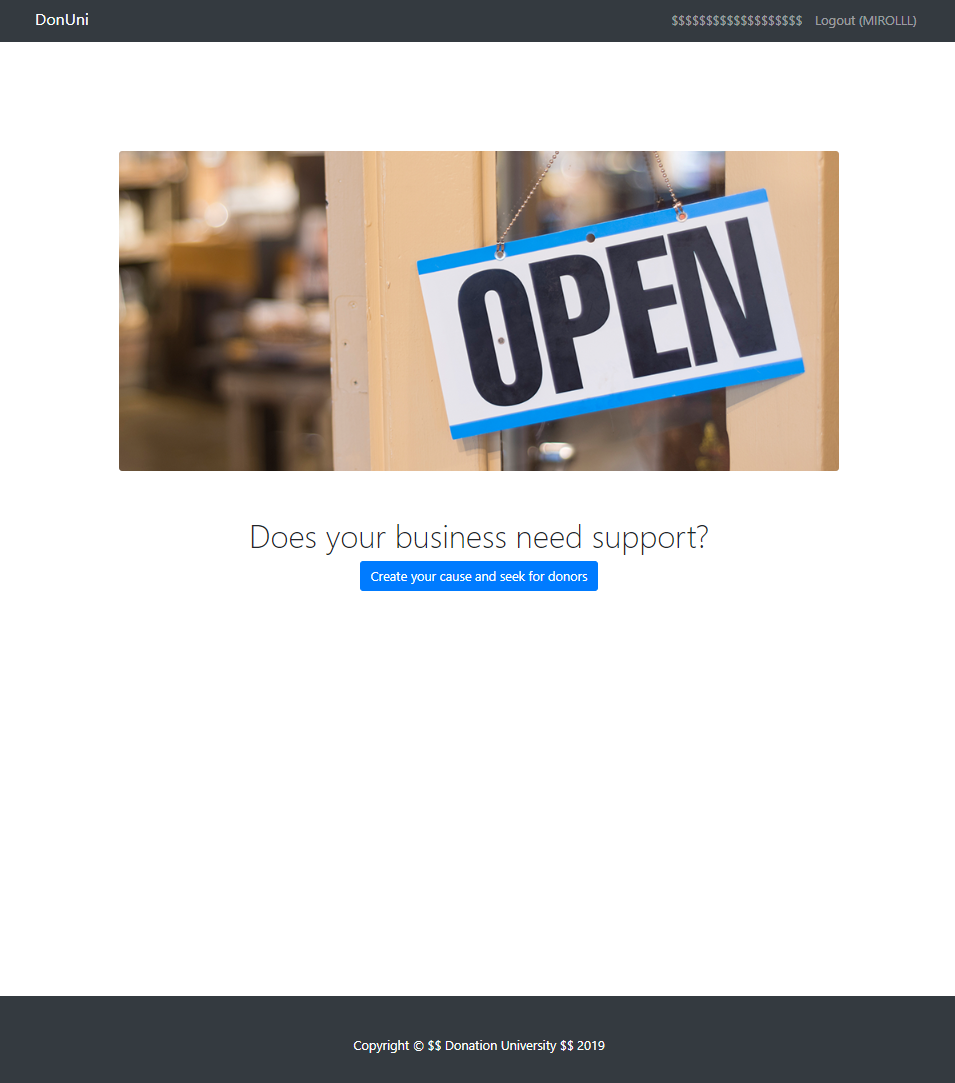
### Home Page (5)

The initial page (view) should display the **navigation bar for guest users** + **Home Page** + Footer. [**DonUni**] **button** should refer to the **home page (view)**

**[Register]** **button** should refer to the **register** view.



If user is logged-in the page (view) should display the **navigation bar logged-in users** + **Home Page** + Footer. And the **[Create your cause and seek for donors] button** should refer to the **create** **cause** **view**.



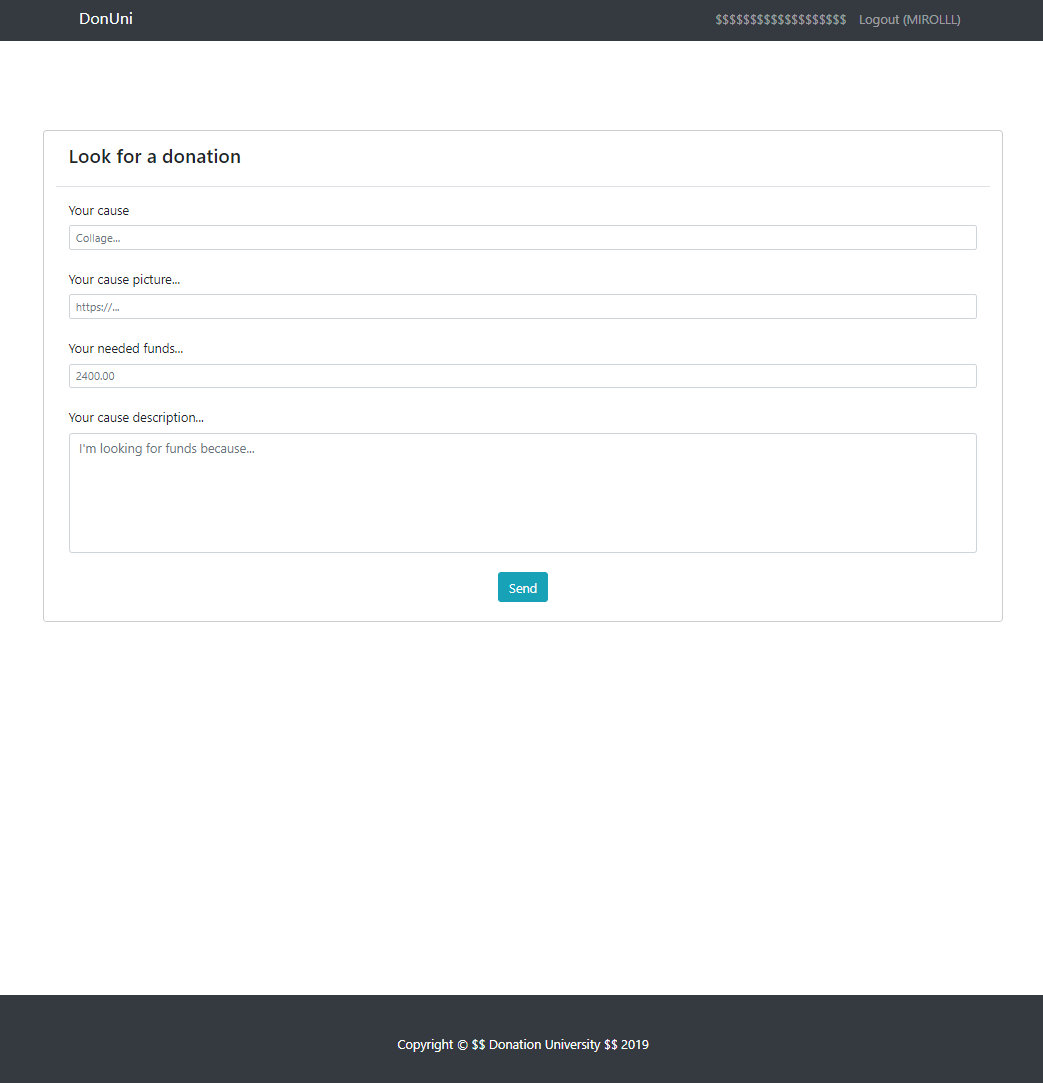
### Create Cause (15)

Logged-in users should be able to **Create causes**.

Clicking the **[Create your cause and seek for donors] button** in the **Home page view** should **display** the Create Cause View.

* The input fields must be **non**-**empty** **strings**
* Note that you should add additional two properties (**donors** and **collectedFunds**)
* After a **successful** offer creation the Dashboard or Home Pageshould be shown

The newly created offer should be stored in the Kinvey collection "causes".

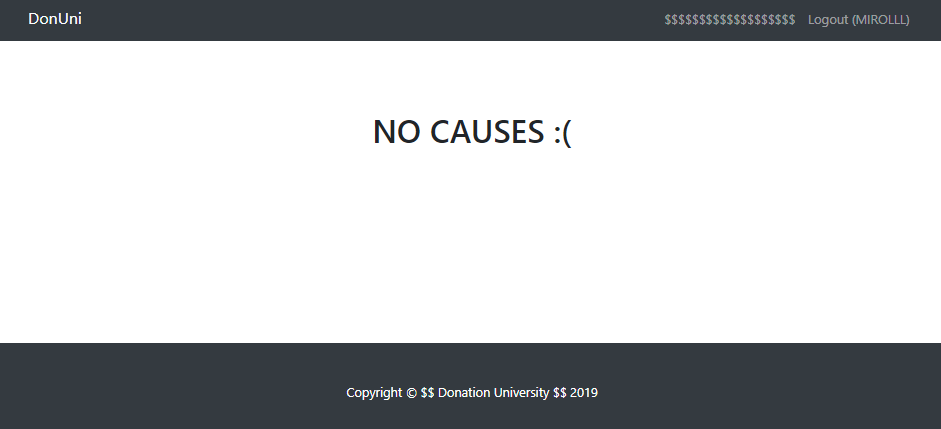


### Dashboard (25)

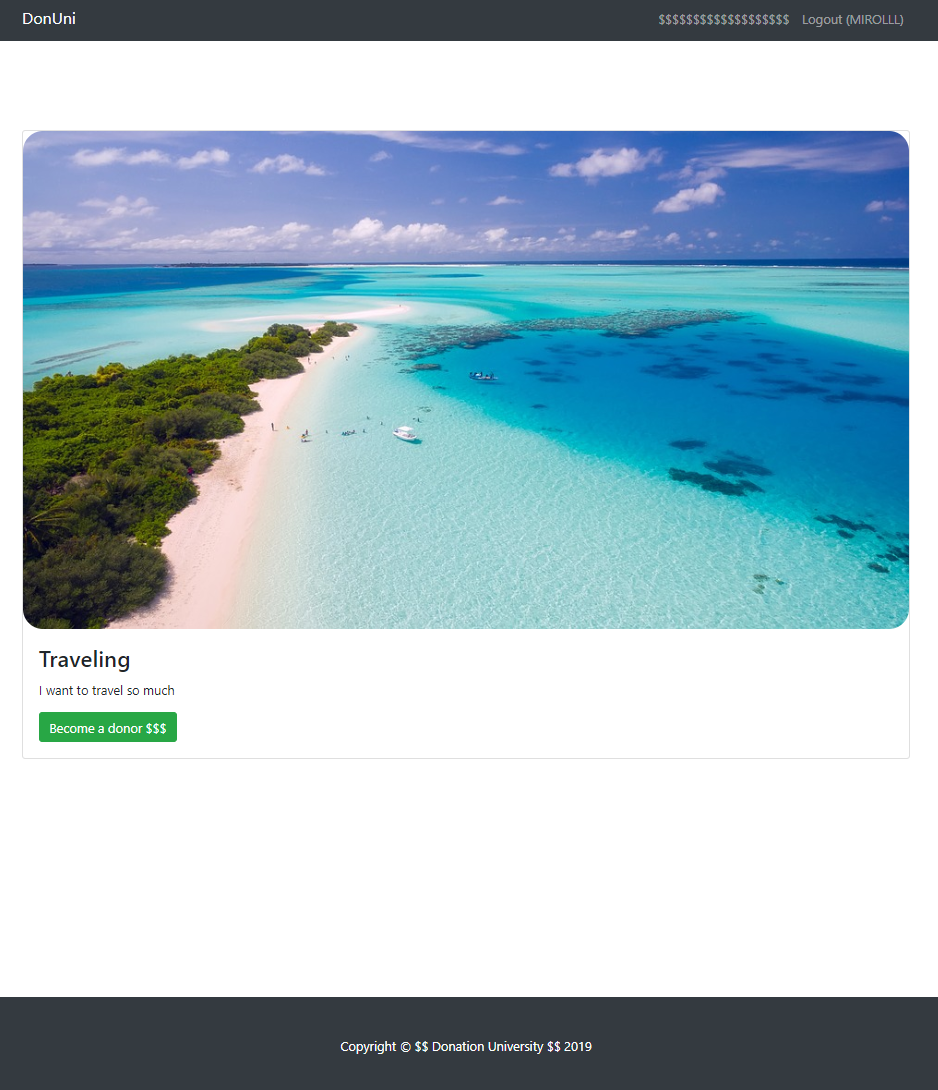
By clicking over [**$$$$$$$$$$$$$$**] button, the currently logged-in user should be able to see all created causes.

The initial page (view) should display the **navigation bar for logged-in users** + **Dashboard** + Footer.

If there are **NO** such causes, the following view should be displayed.



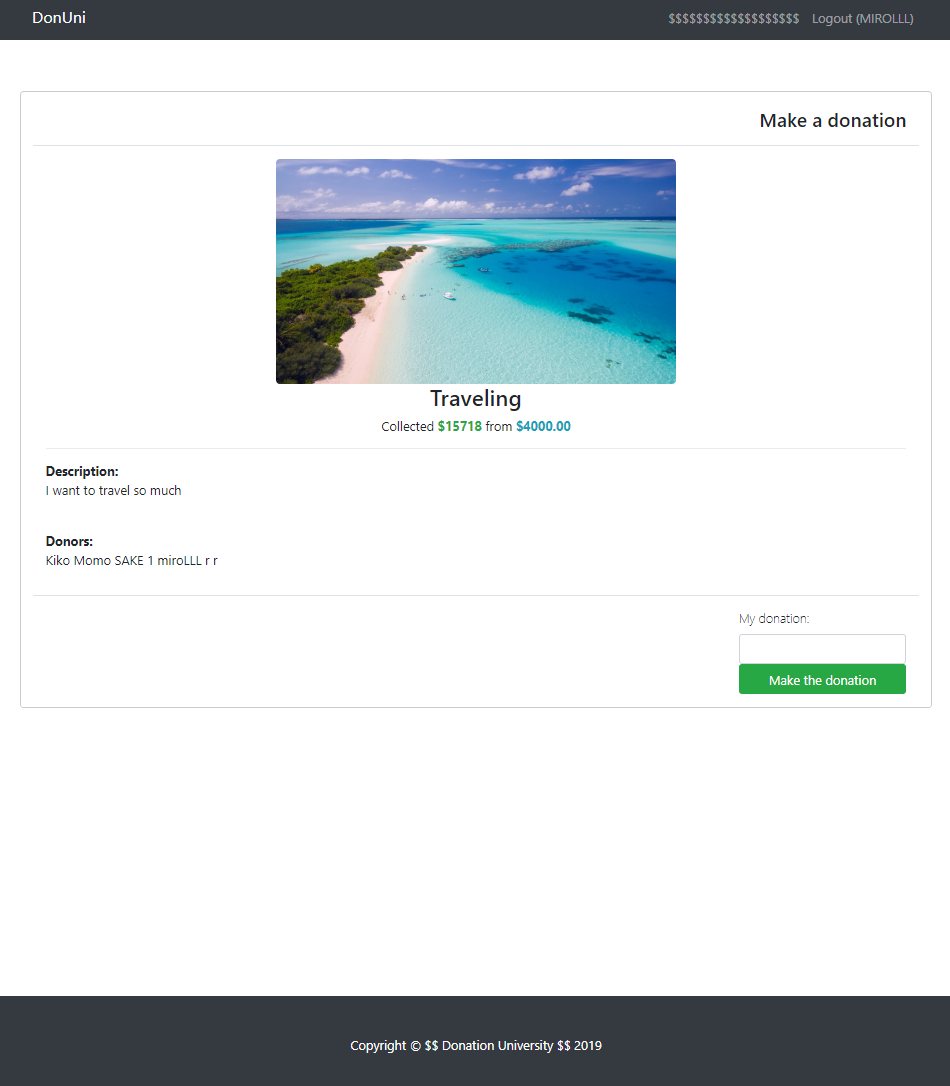
If there is at least **one registered cause,** the causes must be shown like the example below:



### Cause Details (15)

Logged-in users should be able to **view details** aboutcauses and their donations.

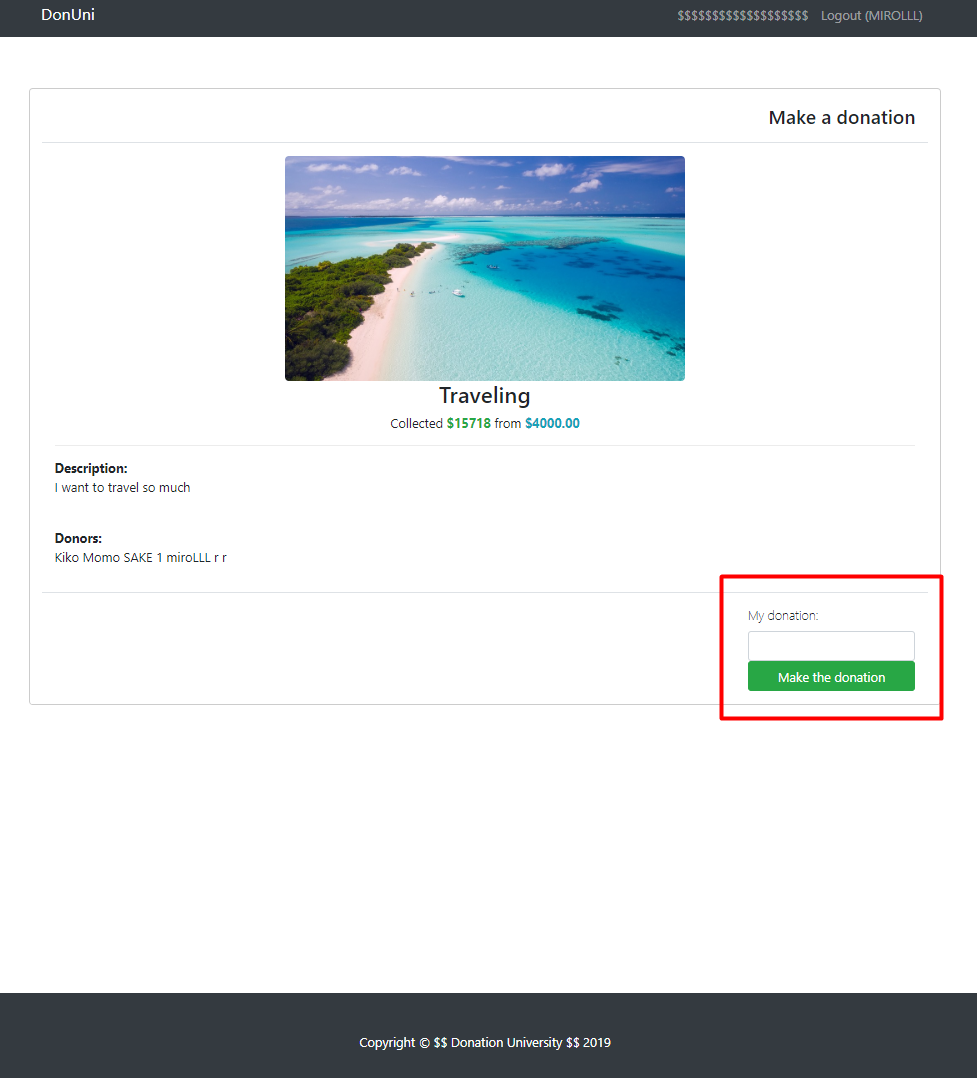
Clicking the [Become a donor $$$] **link** in of a **particular cause** should **display** the Cause Details view, where the cause's **name**, **pictureUrl**, **collected** and **needed** **funds**, **description** and **donors** are shown in the format below on the picture.



### Edit Cause (15)

Logged-in users should be able to **edit (Make a donations) to others causes**.

Clicking the [Make the donation] **link** of a **particular cause**, a put request should be send to the **causes** **collections** and the current cause'**collected funds** should be changed(**increased** with the **given value**).



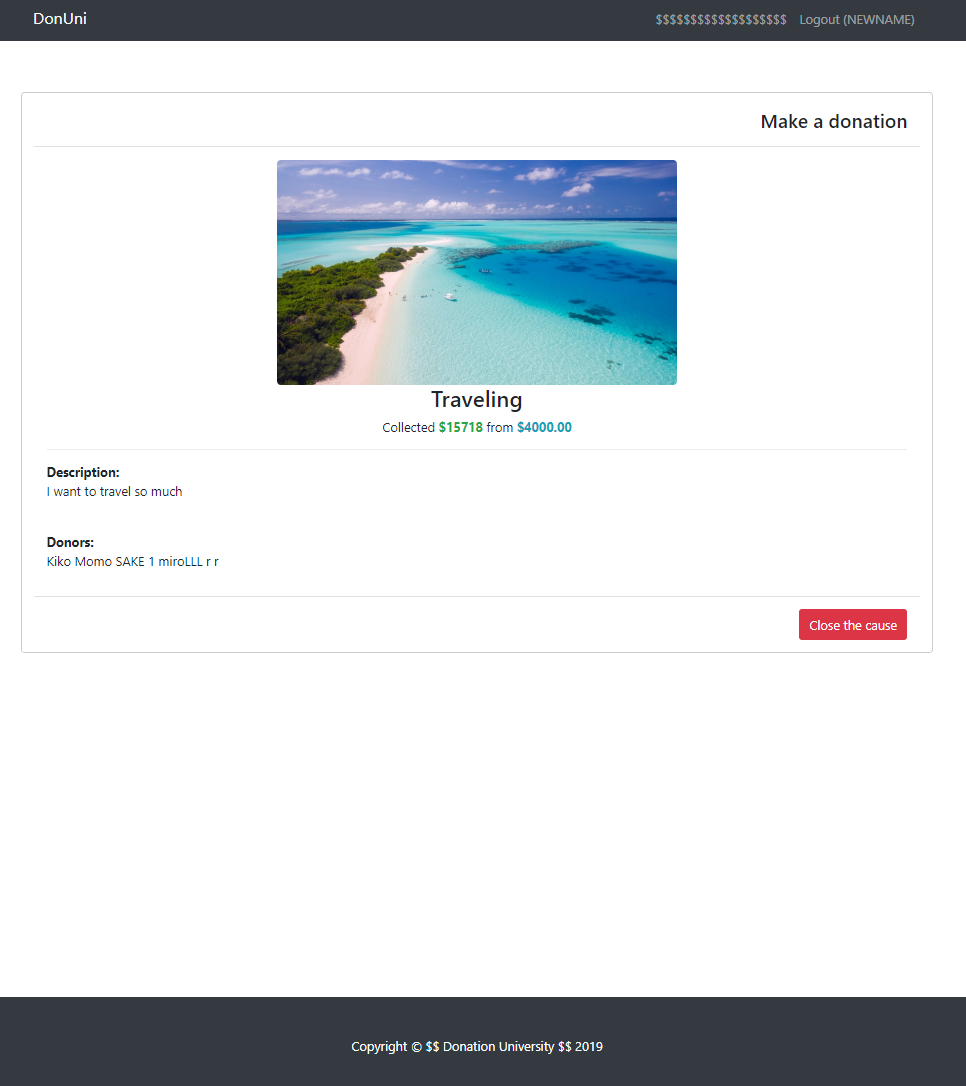
### Delete Offer (5)

Logged-in users should be able to **delete their** causes.

Clicking the [Delete] **link** of an **particular offer** the Offer Delete page should be displayed with already filled information about that offer, but all input fields are disabled.

If the [**Close the cause**] button in the **Cause Details View**, the current offer must be **deleted** from the DOM and from the Kinvey collection.

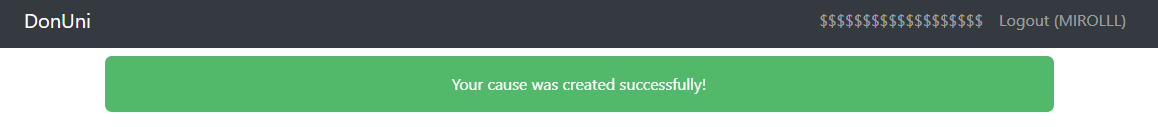
After **successful** **offer delete** a **Dashboard page** must be shown.



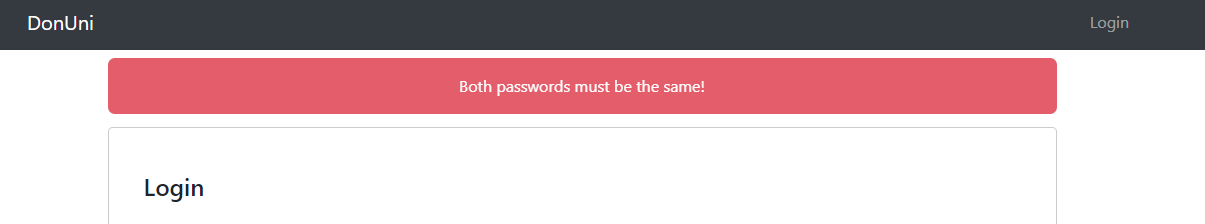
### BONUSES (5)

#### Notifications (5)

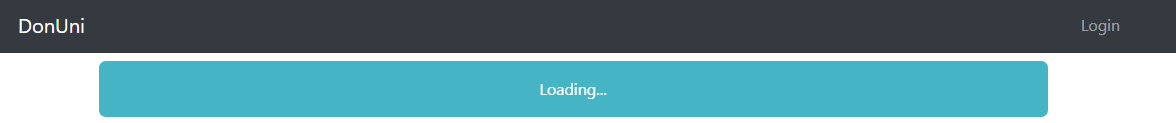
* In case of a **successful** action (create cause, login, registration, edit, delete), a **notification message** **(green)** should be shown, which disappears manually when the user clicks it.



* In case of **error** (wrong passwords, wrong ajax call, authorization, etc...), 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.



### Note

**Each view should have his own unique and properly named action (url), to be able to get the full points from each section.**

*GOOD LUCK!😊*