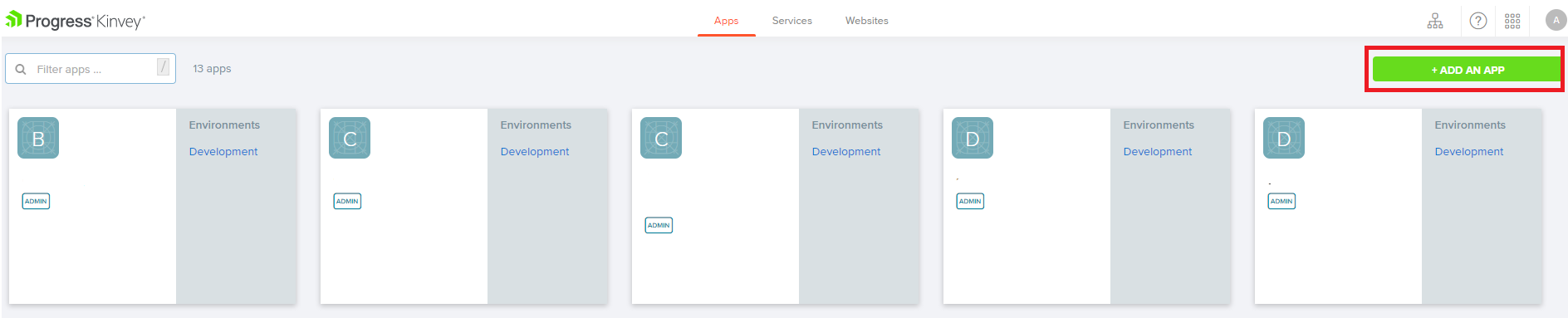
# JS Applications Exam - SoftBay SPA

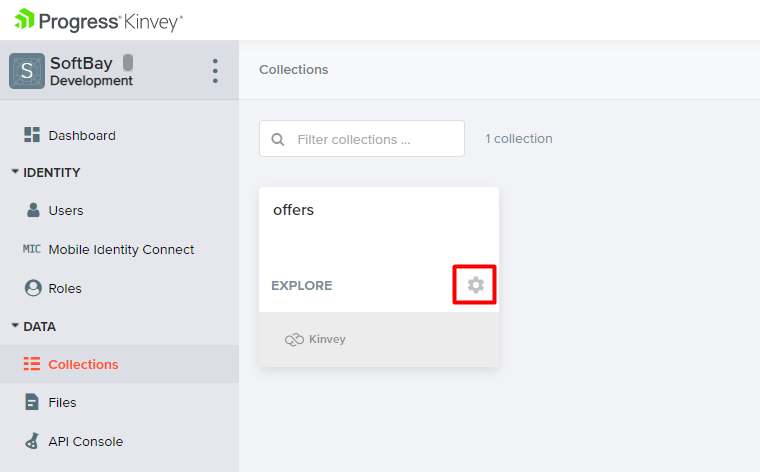
You are assigned to implement a **Web application** (SPA) using HTML5, JavaScript, AJAX, REST and JSON with cloud-based backend (Kinvey). Using libraries like **jQuery**, **Handlebars** and **Sammy** is allowed. The app keeps **users** and **offers**. **Guests** should be able to **register** and **login**.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

## Create a Kinvey REST Service

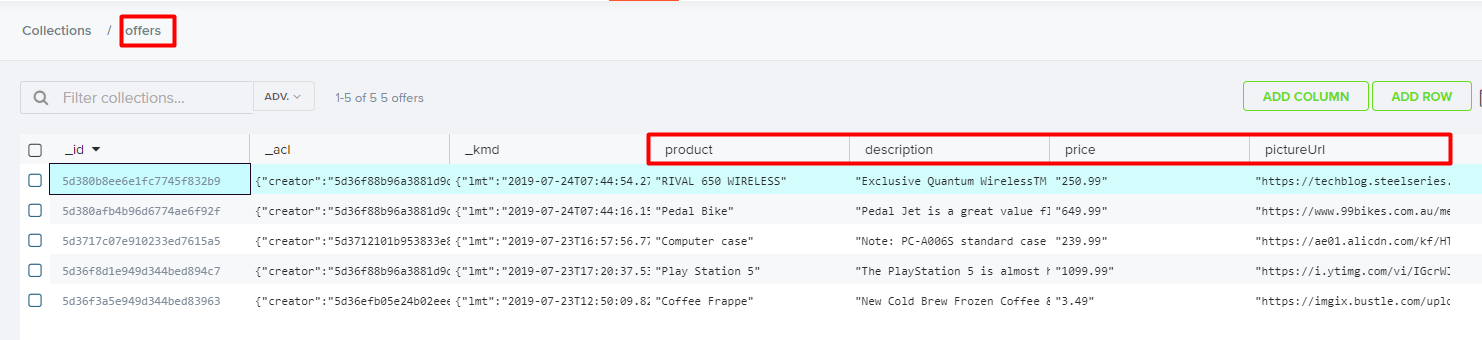
Register at **Kinvey.com** (<https://console.kinvey.com/login>)and create application to keep your data in the cloud.



Create a collection called **offers.** Each **offer** has a **product**, **description**, **price** and **pictureUrl**.



Here some example with already created offers:



## 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 Offers)

|  |  |
| --- | --- |
| **GET** [https://baas.kinvey.com/appdata/***app\_key***/offers](https://baas.kinvey.com/appdata/app_key/offers) | |
| **Request headers** | Authorization: Kinvey authtoken |
| **Response**  **200 OK** | [{  "product":"Bike",  "price":"399.99",  "description":"Are you parent of...",  "imageUrl":"https://....",  "\_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 Offer

|  |  |
| --- | --- |
| **POST** [https://baas.kinvey.com/appdata/***app\_key***/offers](https://baas.kinvey.com/appdata/app_key/offers) | |
| **Request headers** | Authorization: Kinvey authtoken  Content-Type: application/json |
| **Request body** | {  "product": "Bike",  "price": "399.99",  "description": "Are you parent of...",  "imageUrl": "https://...",  } |
| **Response**  **201 Created** | {  "product": "Bike",  "price": "399.99",  "description": "Are you parent of...",  "imageUrl": "https://...",  "\_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 Offer

|  |  |
| --- | --- |
| **PUT** [https://baas.kinvey.com/appdata/***app\_key***/offers/offer***\_id***](https://baas.kinvey.com/appdata/app_key/offers/offer_id) | |
| **Request headers** | Authorization: Kinvey authtoken  Content-Type: application/json |
| **Request body** | {  "product": "Bike",  "price": "499.99",  "description": "Are you parent of...",  "imageUrl": "https://...",  } |
| **Response**  **200 Ok** | {  "\_id": "59931398996ab5127d2a84d1",  "product": "Bike",  "price": "499.99",  "description": "Are you parent of...",  "imageUrl": "https://...",  } |
| **Error response**  **401 Unauthorized** | { "error": "InvalidCredentials", "description": "Invalid credentials. Please retry your request with correct credentials", "debug": "" } |

### Delete Offer

|  |  |
| --- | --- |
| **DELETE** [https://baas.kinvey.com/appdata/***app\_id***/offers/offer***\_id***](https://baas.kinvey.com/appdata/app_id/offers/offer_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": "" } |

## SoftBay - 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.

## SoftBay - Client-Side Web Application

**Design** and **implement** a client-side front-end app (SPA) for managing **offers**. 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: **[**SoftBay], **[**Dashboard**]**, **[Create Offer]** and **[**Logout**]**.



The guest users navbar should contain the following elements: **[SoftBay]** and **[Sign In]**



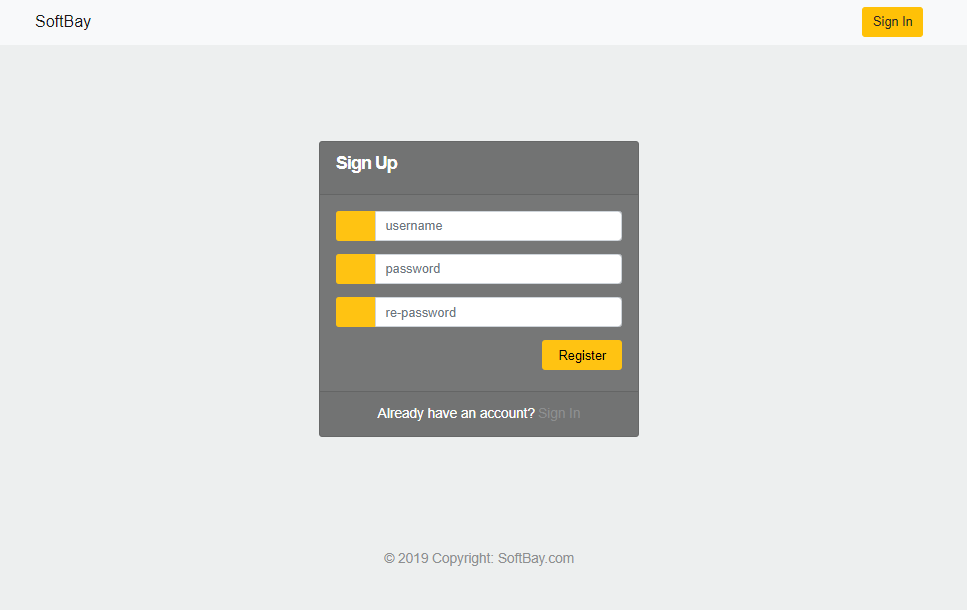
### Footer



### 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**.

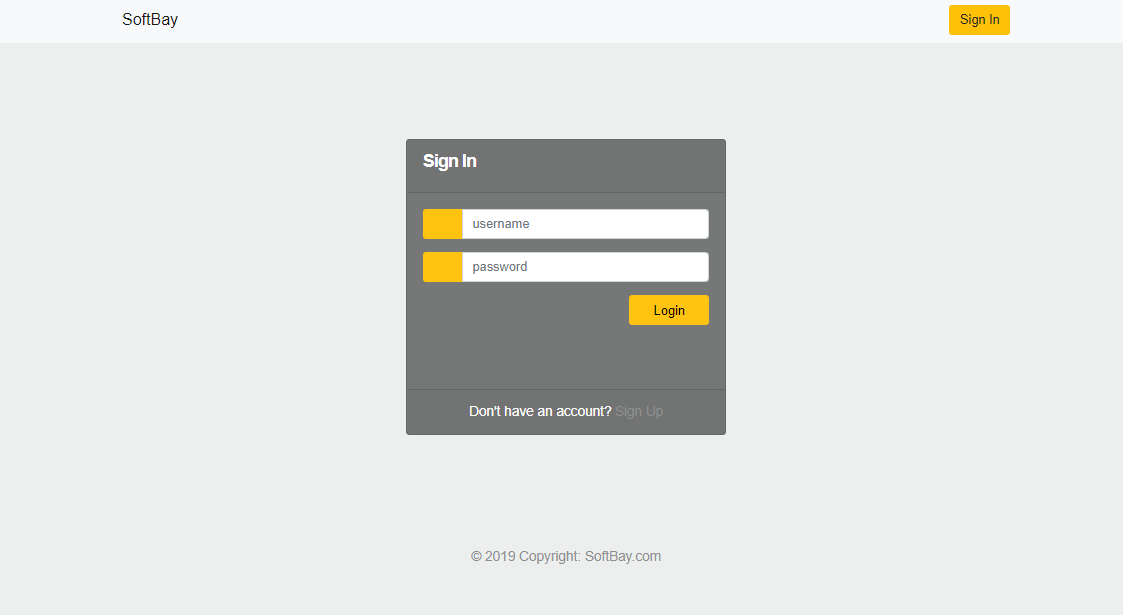


[**Sign In**] **button** should refer to the **sign-in (login) form (view)**

### 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**.



[**Sign Up**] **button** should refer to the **sign-up (register) form (view)**

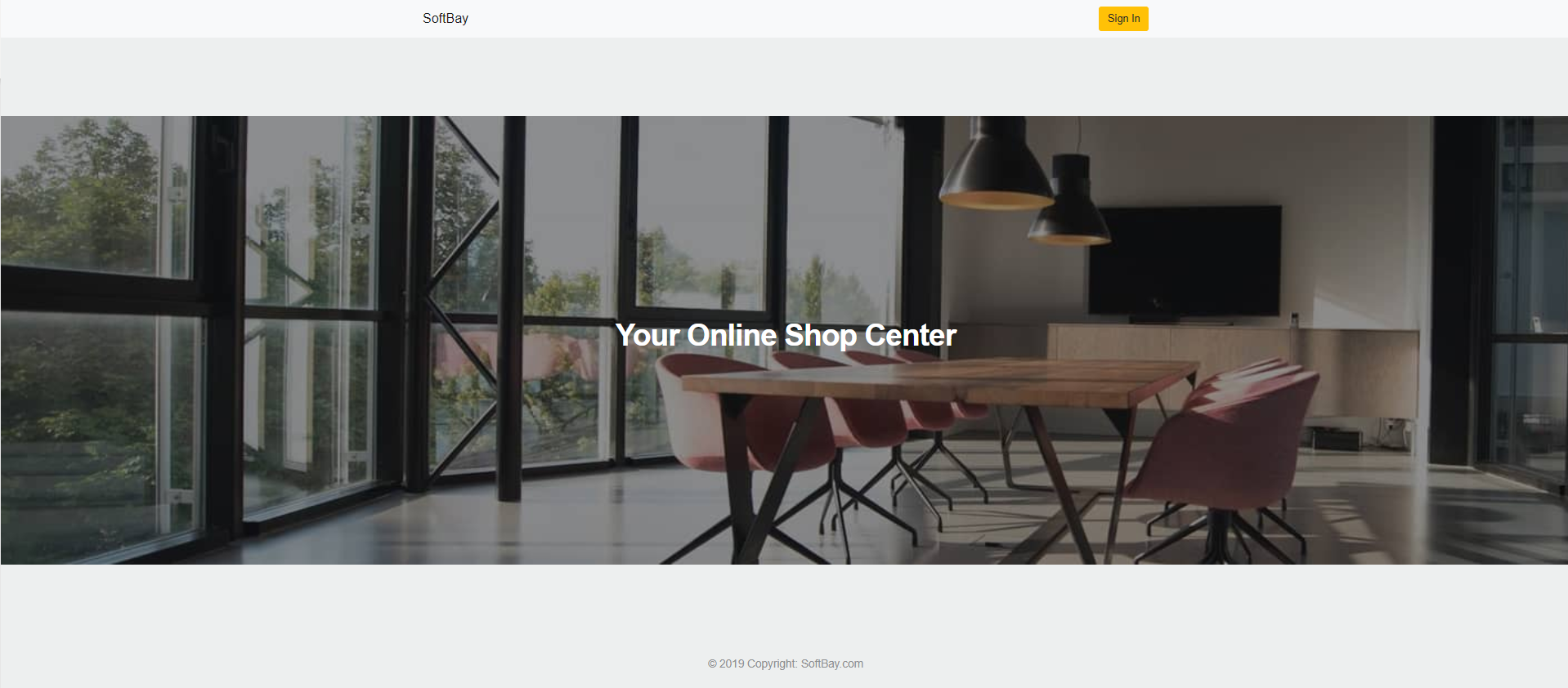
### Logout (5)

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

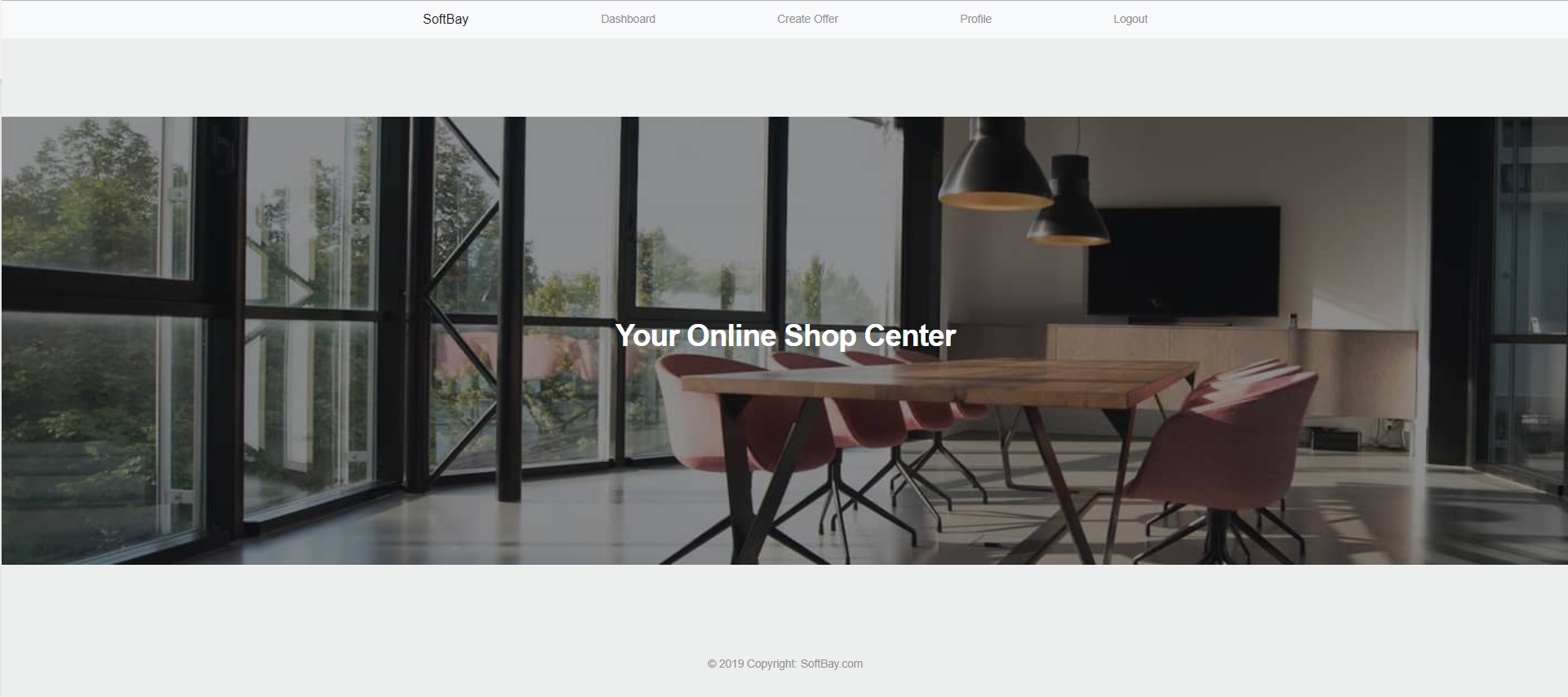
* After a **successful** logout the **home page for guest users** should be shown
* The **"logout" REST service** at the back-end **must** be called at logout
* All local information in the browser (**user session data**) about the current user **should be deleted**

### Home Page (5)

The initial page (view) should display the **navigation bar for guest users** + **Home Page** + Footer.



If user is logged-in the page (view) should display the **navigation bar logged-in users** + **Home Page** + Footer.



### Create Offer (15)

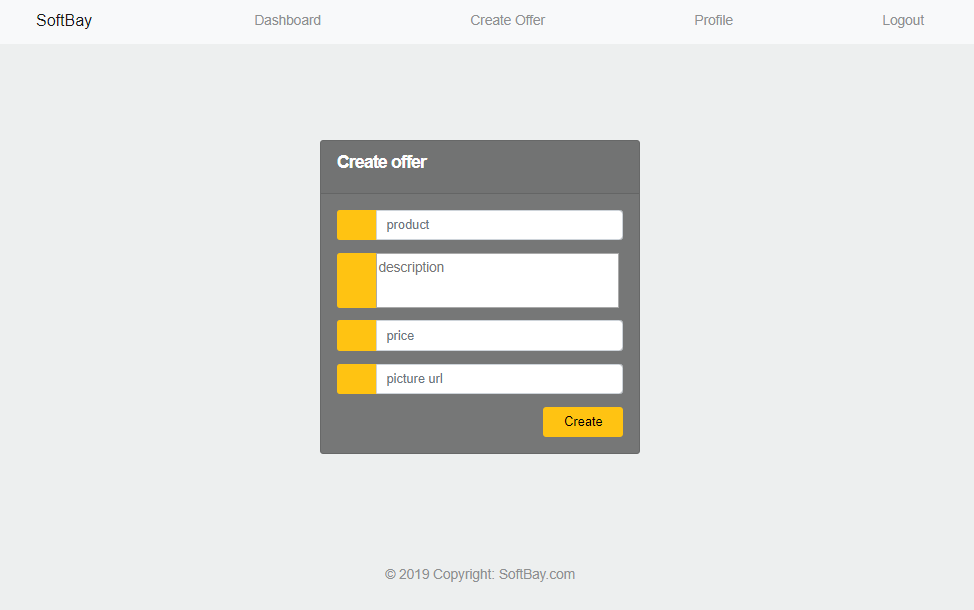
Logged-in users should be able to **Create offers**.

Clicking the [Create Offer] **link** in the **NavBar** should **display** the Create Offer page.

* The form should contain the following validations:
  + The input fields for **product**, **description** and **price** should be **non-empty strings**
  + The input field for **imageUrl**, must be **valid** **url** **refering** **to** **picture**

After a **successful** offer creation the Dashboardshould be shown.

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

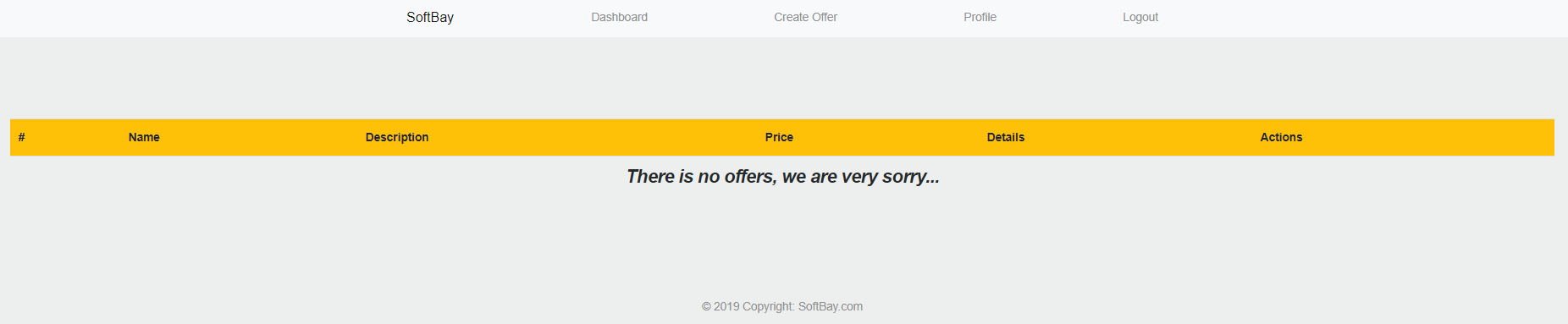


### Dashboard (25)

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

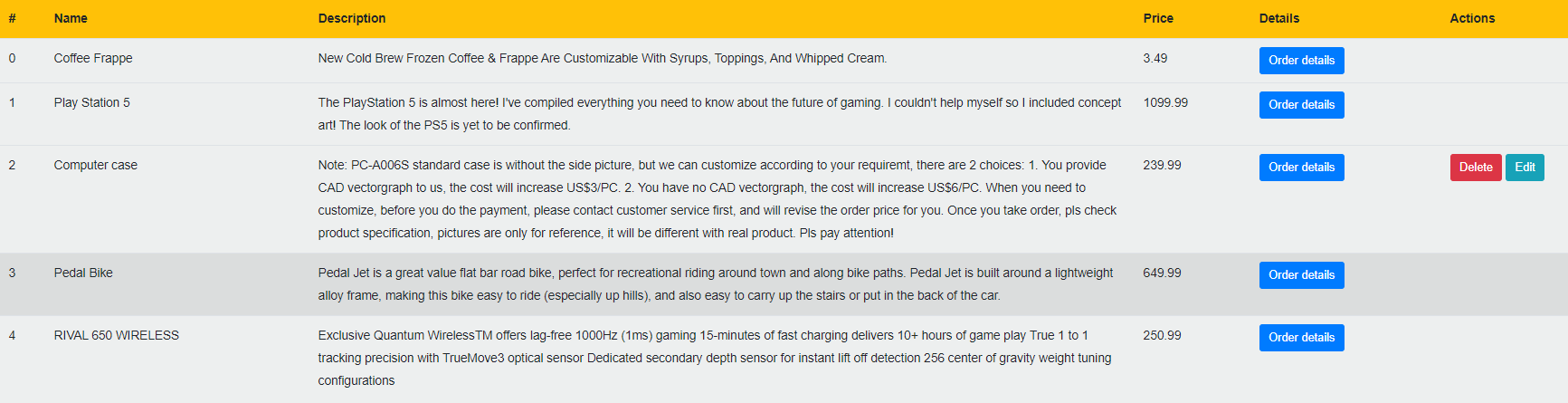
The initial page (view) should display the **navigation bar for logged-in users** + **Dashboard** + Footer.

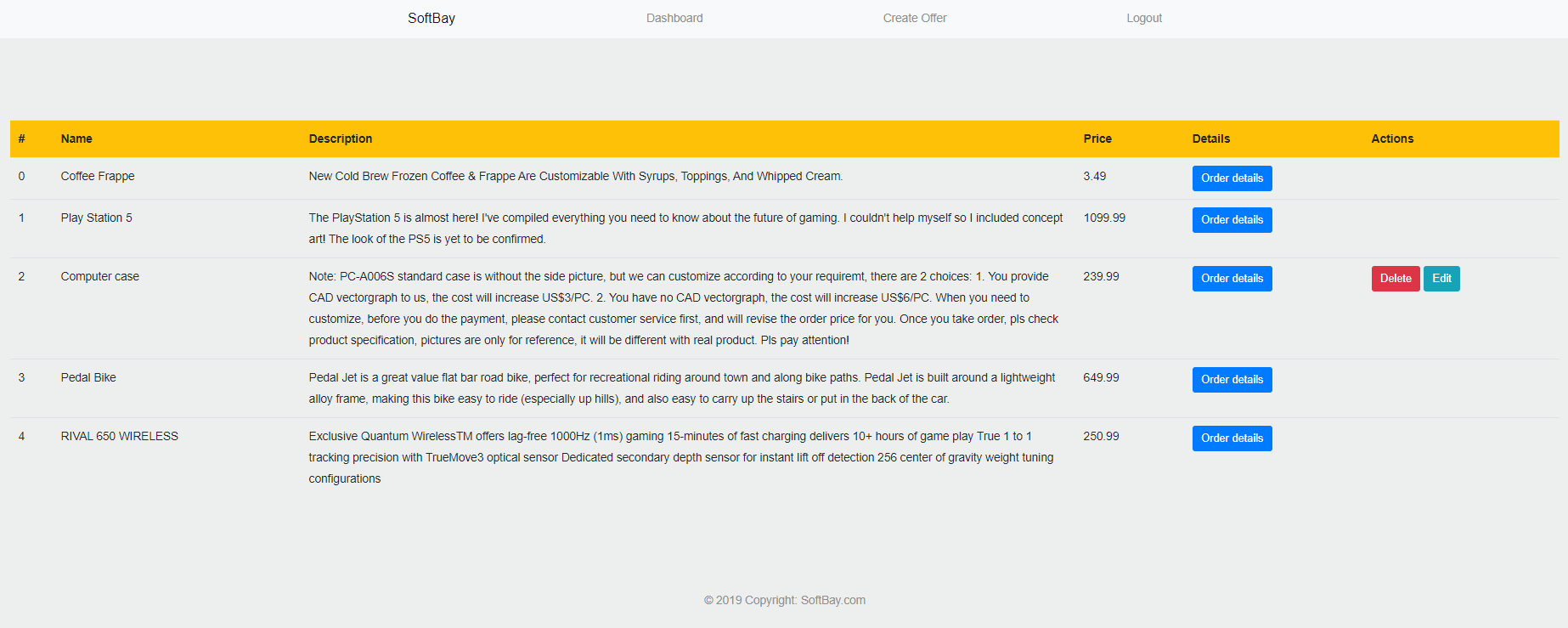
If there are **NO** such offers, the following view should be displayed.



If there is at least **one registered offer,** the offer must be shown like the example below in format:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Name | Description | Price | Details | Actions |
| Current offer **index** | Current product **name** | Current product **description** | Current product **price** | [**Order details**] button which refer to the details for the current offer/product | [**Delete**] [**Edit**]  Buttons, only if the currently logged-in user is the creator of that offer. Where both buttons refer to the delete and edit view (pages) for the current offer/product |

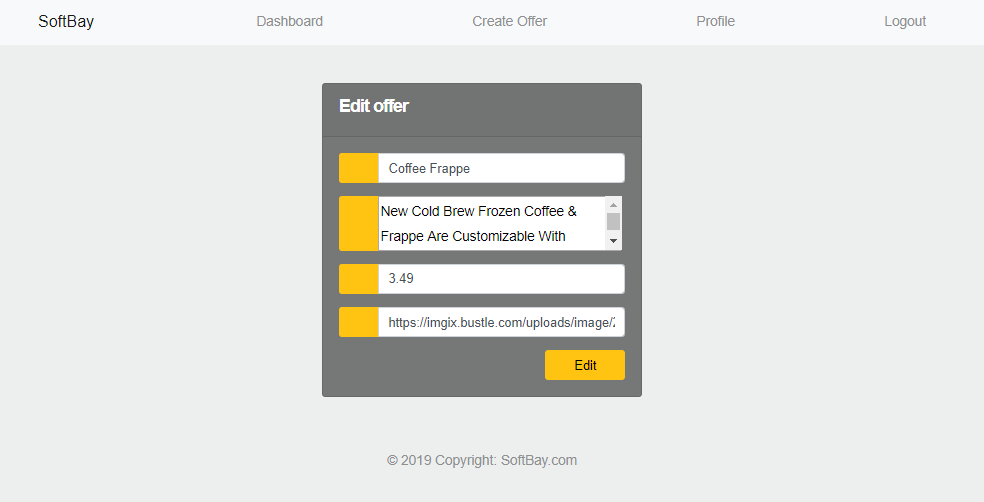




### Edit Offer (15)

Logged-in users should be able to **edit** their **own** offers.

Clicking the [Edit] **link** of a **particular offer**, Edit Offer Page should be displayed with **already filled** input fields with the current offer information. By cliking the [**Edit**] button, a put request should be send to the **offers** **collections** and the current offer **should be changed**.



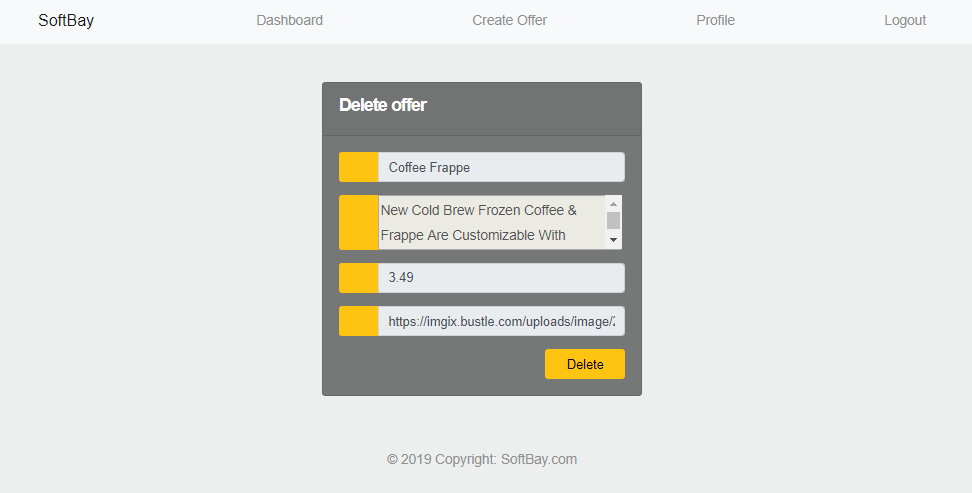
### Delete Offer (5)

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

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 [**Delete**] button in the **Offer Delete Page** is clicked, the current offer must be **deleted** from the DOM and from the Kinvey collection.

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



### Offer Details (15)

Logged-in users should be able to **view details** aboutoffers.

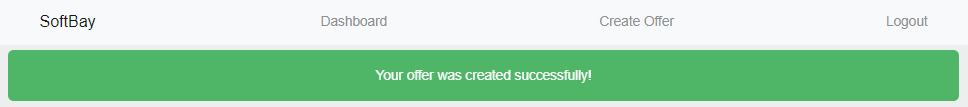
Clicking the [Order details] **link** in of a **particular offer** should **display** the Offer Details page, where the **product** **name**, **picture**, **description** and the **price** are shown in the format below on the picture.



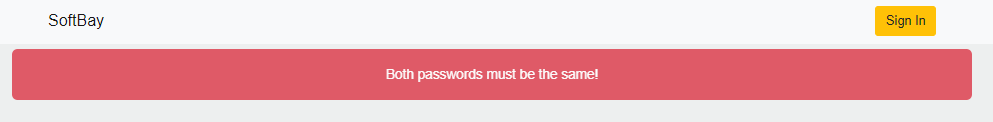
### BONUSES (20)

#### Notifications (5)

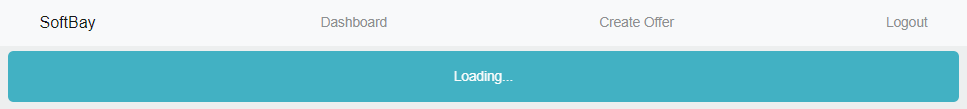
* In case of a **successful** action (create offer, login, registration, edit, delete), a **notification message** **(green)** should be shown, which disappears manually when the user clicks it.



* In case of **error** (wrong passwords, wrong ajax call, authorization, etc...), an **error notification message (red)** should be shown, which disappears on user click.

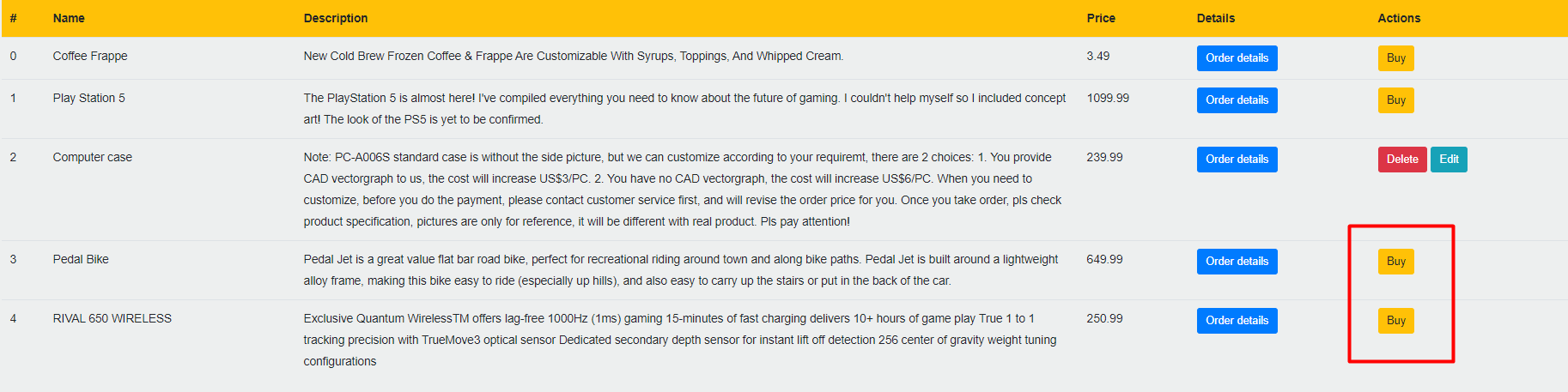


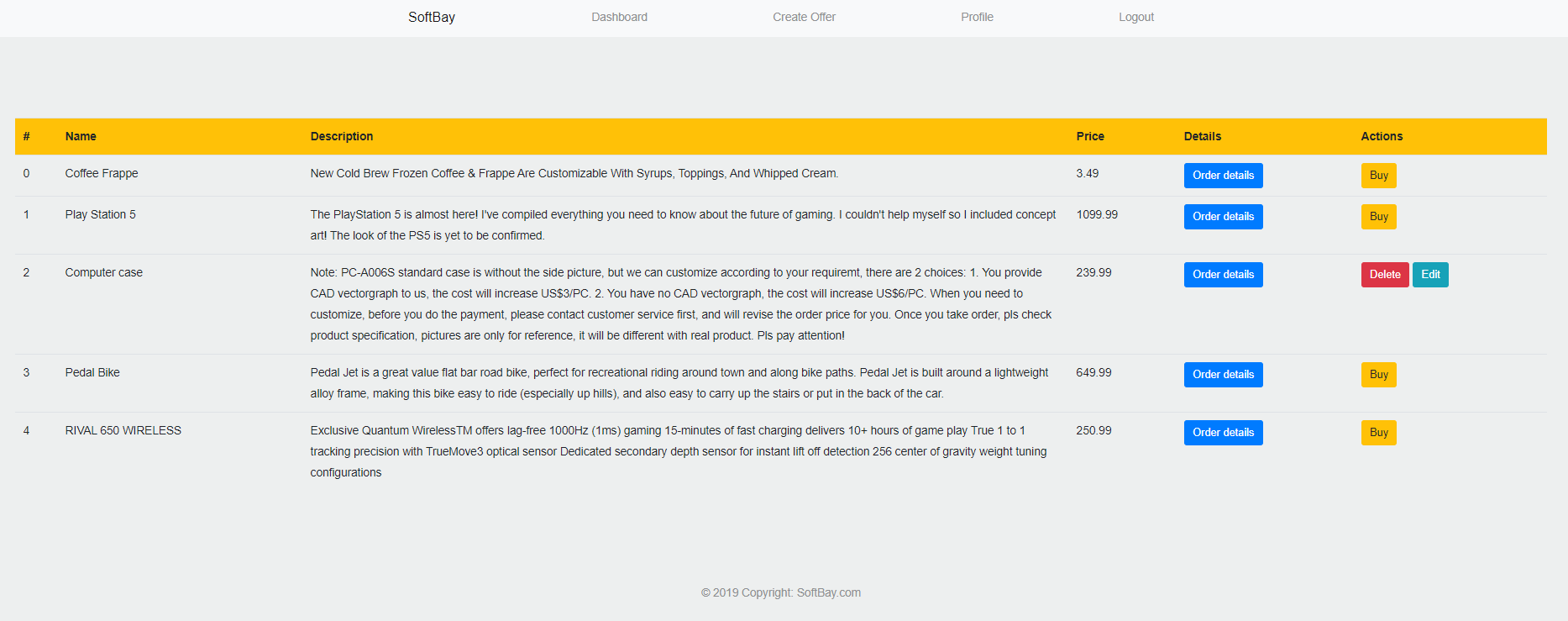
* During the **AJAX calls** a **loading notification message (blue)** should be shown. It should disappear automatically as soon as the AJAX call is completed.



#### Buy an item + Profile page (15)

In these cases when some of the offers is **not made** from the currently logged-in user, in the section "**Actions**" must be added a [**Buy**] button, which by clicking it the current product must be successfully bought by the user (there are no quantities on the products, so you don’t have to remove them from the list or something else, just store the number of purchases in the kinvey for the current user)



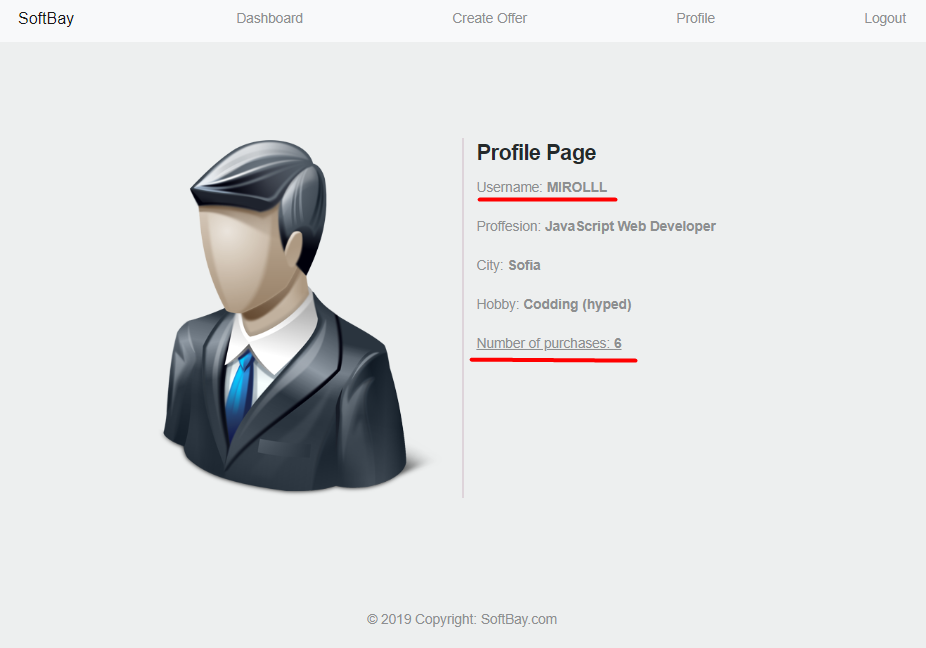


#### Profile Page

In the NavBar (header) must be added a new link [**Profile**] which refer to the currently logged user.



It should be display **2** types of information: **username** for the current user and **number** **of** **purchases** that current user is made. **Any other info is static,** and you don’t have to think about it.



*GOOD LUCK c(:*