High Level Description

You have to build a very basic application that consumes an REST API and displays information about users. In UI terms, it consists of:

- A List that displays only usernames
- A pop up that allows editing user details

Api Description

1) Get All Users Endpoint

Uri: https://jsonplaceholder.typicode.com/users
Http Method: GET
Sample Response:

```
[
  {
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  }
]
```

2) Update User Endpoint

Uri: https://jsonplaceholder.typicode.com/users**/{userId}**
Http Method: PUT
Sample Request:

Put data at https://jsonplaceholder.typicode.com/users**/1**

```
{
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  }
```

Sample Response:

```
{
    "id": 1,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
```

```
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
}
```

Important: user will not be really updated on the server but it will be faked as if.

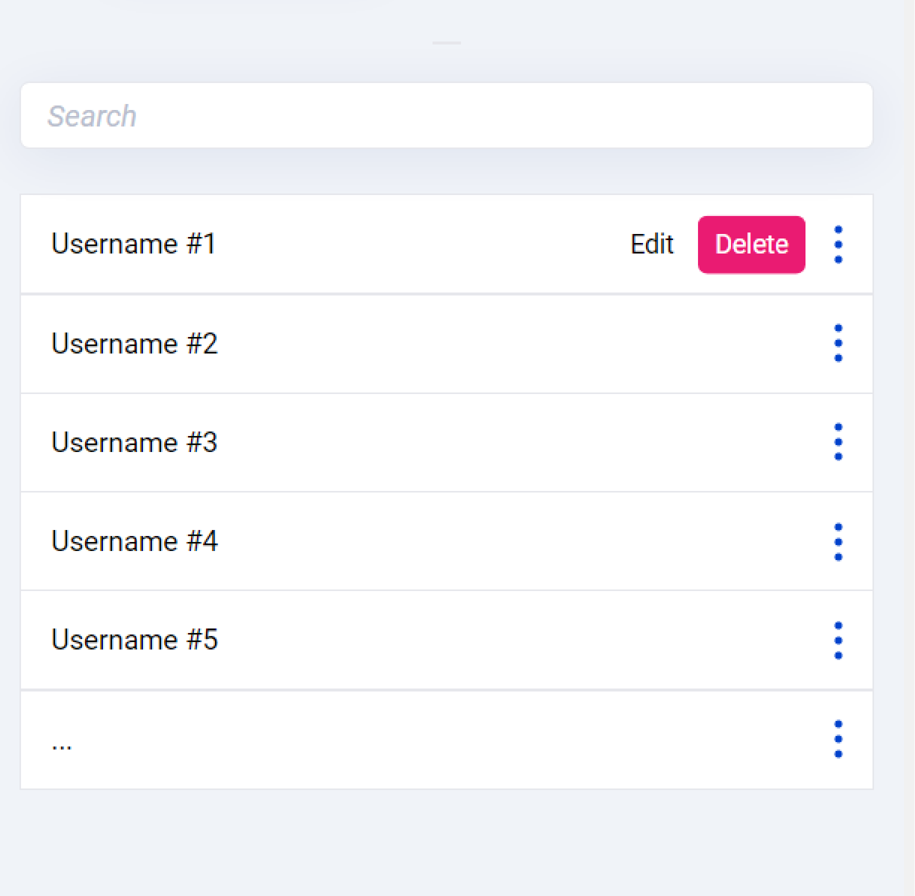
3) Delete User Endpoint

Uri: https://jsonplaceholder.typicode.com/users**/{userId}**
Http Method: DELETE
Sample Response: Empty

Important: user will not be really deleted on the server but it will be faked as if.

UI/UX Description

*Please take special care so that the delivered UI/UX matches as much as possible the one presented in this section (colors, placement, sizes, actions etc)



Username #1

Username #2

Username #3

Username #4

Username #5

...

Search Bar

On the top of the screen, a search bar filters in real time the list as the user types. E.g. When I type “Jo” the list below must display only users with username like “John” and “Jones”

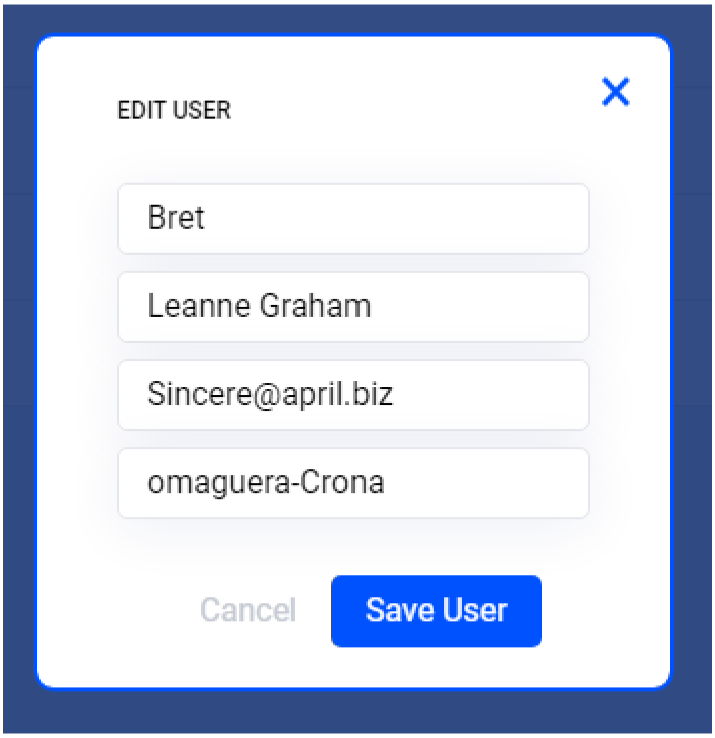
List

All Users fetched using the Get All endpoint must be displayed here. We just need their username to be visible.

Action Buttons

When the “three dots” are clicked, the edit and delete buttons are displayed in the corresponding list row only. By clicking delete, the Delete User endpoint must be called and the row must be removed on success. A confirmation dialog (“Are you sure you want to delete”) must be first displayed.

By clicking edit, the edit user modal is displayed



Bret

Leanne Graham

Sincere@april.biz

omaguera-Crona

Edit User

Display 4 input boxes for the User properties below:

- user.username
- user.name
- user.email
- user.company.name

When the X icon or the Cancel button is clicked, the pop up disappears. When the Save User button is clicked, the Update User endpoint is called.

Validations

The Save user button is not enabled if any of the fields above is empty, or contains whitespace only.

NOTES

API Documentation details: https://jsonplaceholder.typicode.com/guide/

Image Color Picker: https://imagecolorpicker.com/en