# ReactJS – User List Workshop

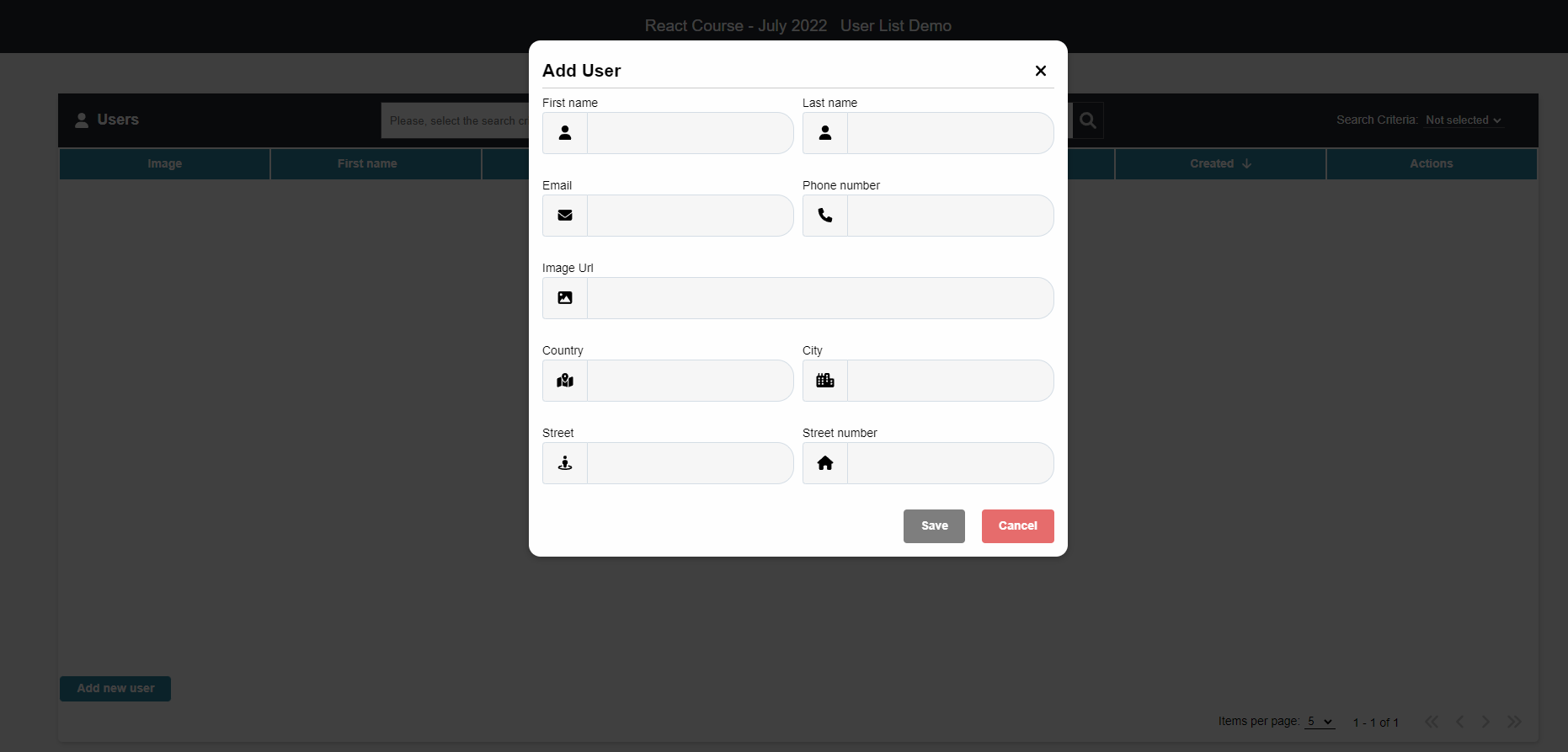
This is application was created as a workshop for the [ReactJS Course](https://softuni.bg/opencourses/react-js) @ [Software University](https://softuni.bg)

## Task Requirements

Use the provided **html** **structure** and **styles** to **create** a page with the following functionalities:

### Create user

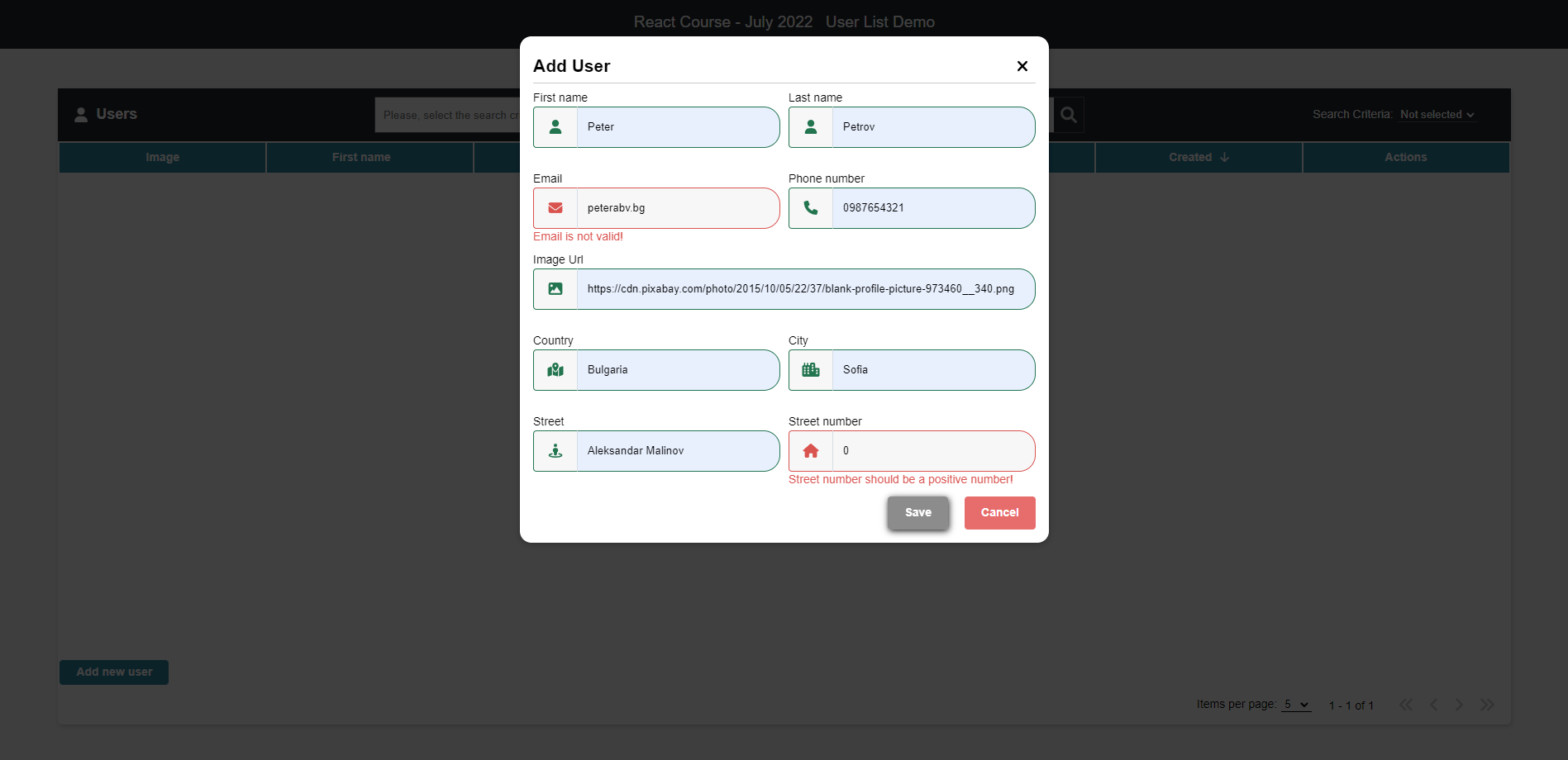
Clicking on the **Add new user** button should display a form for creating users.



All fields must be filled **correctly** before sending the request. If any of the fields is not filled or it does not meet its requirements, the **[Save]** **button** should be **disabled** and their **error messages** should be visible.

Keep in mid that there are server validations while **creating** or **editing a** **user**:

* The **First Name** should be **at least three characters**
* The **Last name** should be **at least three characters**
* The **Email** should contain **@** with **atleast 3** character after it
* The **ImageUrl** should should start with **https://**.
* The **Phone nubmer** should start with **0** and be **10** characters
* The **Country** should be **at least two characters**
* The **City** should be **at least three characters**
* The **Street** should be **at least three characters**
* The **Street number** must be a number larger than **0**



Otherwise send the following **request**:

Method: POST

URL: /users

The service expects a body with the following shape:

{

firstName,

lastName,

email,

imageUrl,

phoneNumber,

address: {

country,

city,

street,

streetNumber

}

}

Then the service will return the newly created record.

Картина, която съдържа текст

Описанието е генерирано автоматично

### User info

Clicking the **[ i ]** button in the Actions section should display a window containing the information for the selected user.

Картина, която съдържа текст

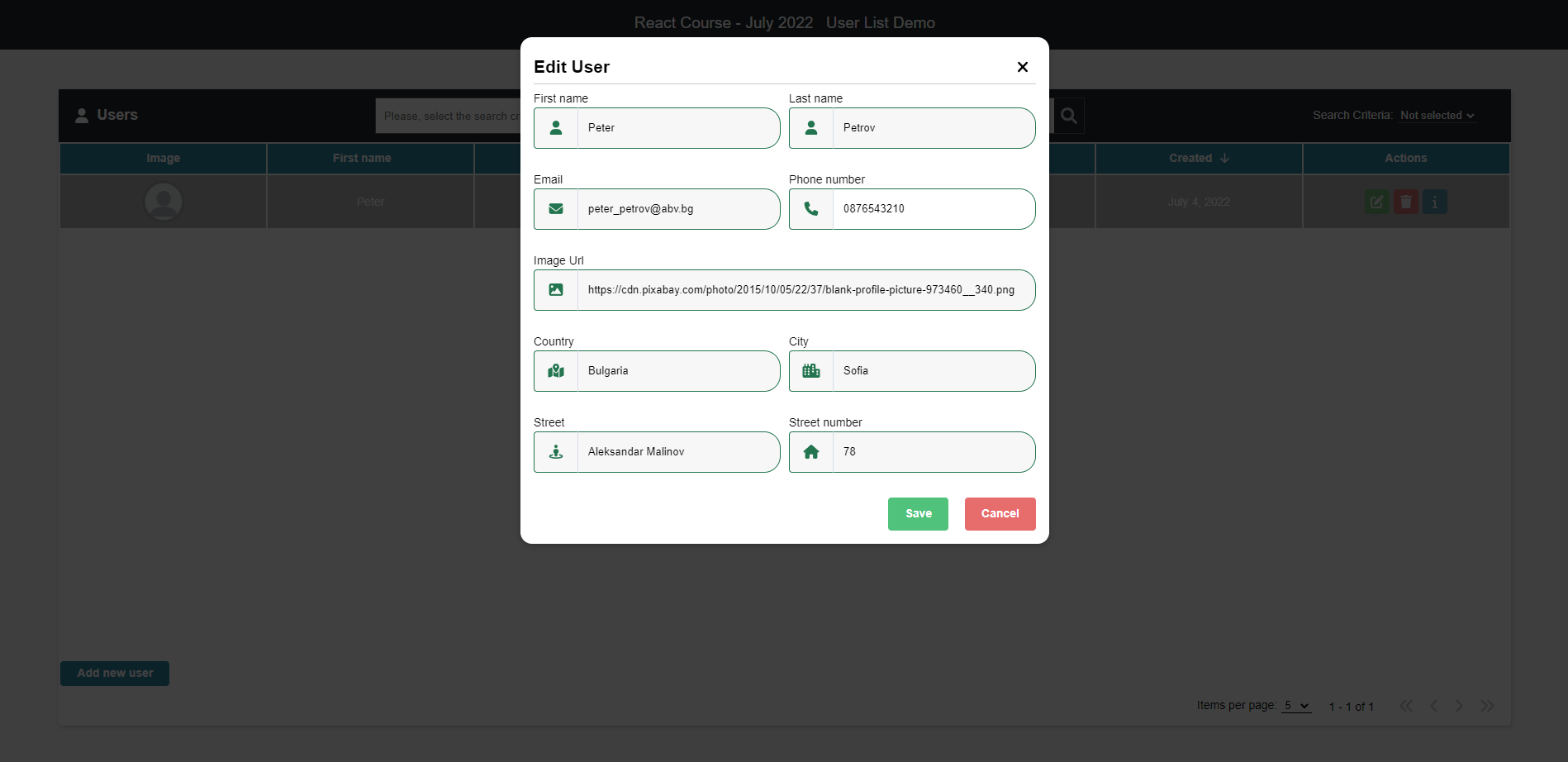
Описанието е генерирано автоматично

To retrieve the information, send the following **request**:

Method: GET

URL: /users/{userId}

### Edit user

Clicking the **[ Edit ]** button in the Actions section should display a form for editing the user information. 

If the form is filled correctly send the following request:

Method: PUT

URL: /users/{userId}

The service expects a body with the following shape:

{

firstName,

lastName,

email,

imageUrl,

phoneNumber,

address: {

country,

city,

street,

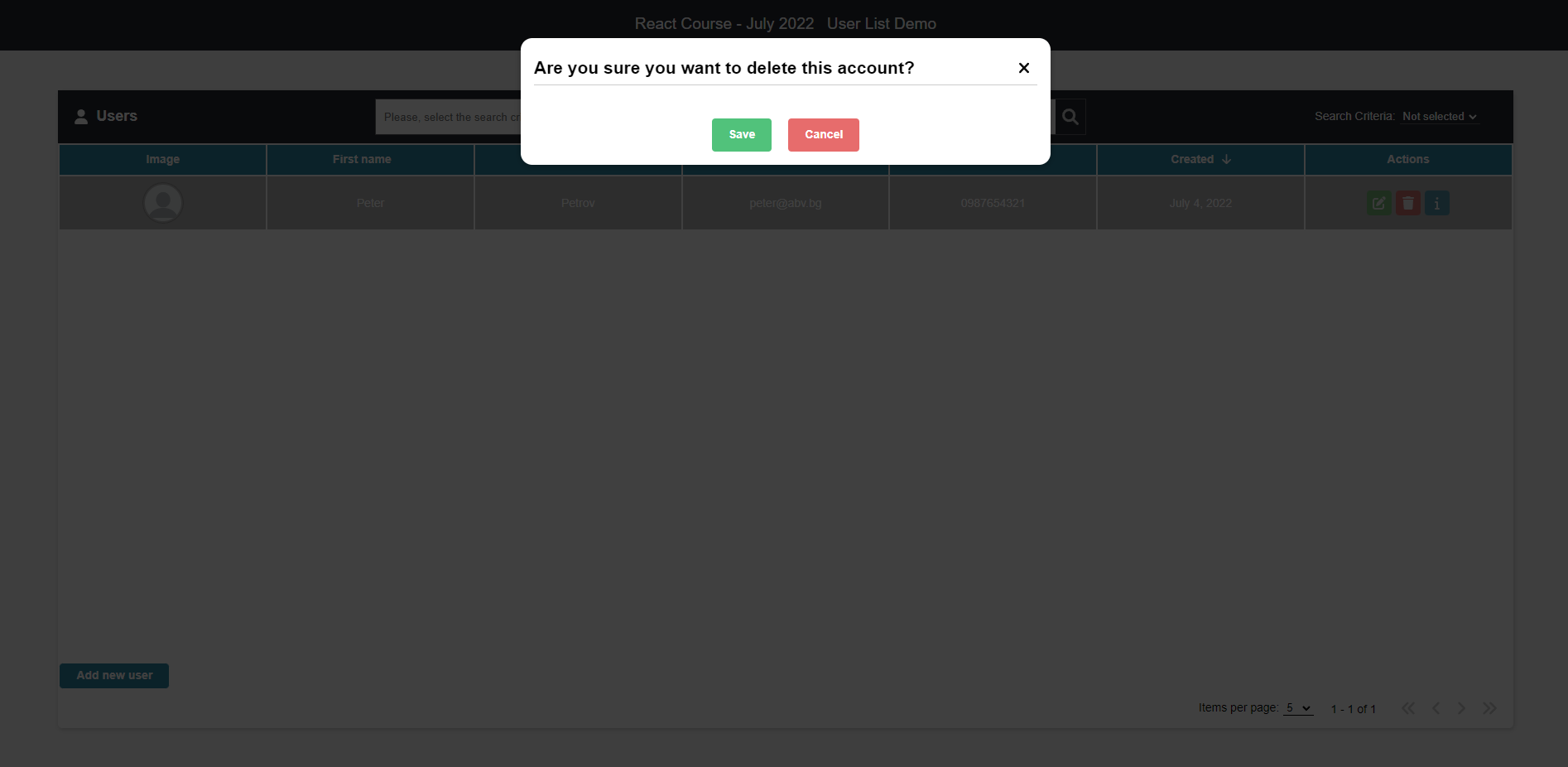
streetNumber

}

}

Then the service will return the newly modified record.

### Delete user

Clicking the **[ Delete ]** button in the Actions section should display a confirmation window. 

Clicking the **[ Save ]** button should send the following request:

Method: DELETE

URL: /users/{userId}

Then the record with the given id will be deleted.

### Search

By selecting a **search criterion** and typing in the search bar the **criteria**, only the matching records should be visible

Картина, която съдържа текст, екранна снимка, монитор, закрито

Описанието е генерирано автоматично

### Pagination

Selecting how many items a page must include, implement a pagination

Картина, която съдържа маса

Описанието е генерирано автоматично

If there are more records than the maximum items per page, a second page must become active.

Картина, която съдържа текст, екранна снимка, монитор

Описанието е генерирано автоматично

### Sorting

Clicking on some of the main fields should **sort** the records in an **ascending** or **descending** order, depending on how many times it is clicked.

Ascending by ‘**Last name’**:

Картина, която съдържа текст, екранна снимка, монитор

Описанието е генерирано автоматично

Descending by ‘**Last name’**:

Картина, която съдържа маса

Описанието е генерирано автоматично

### Spinner

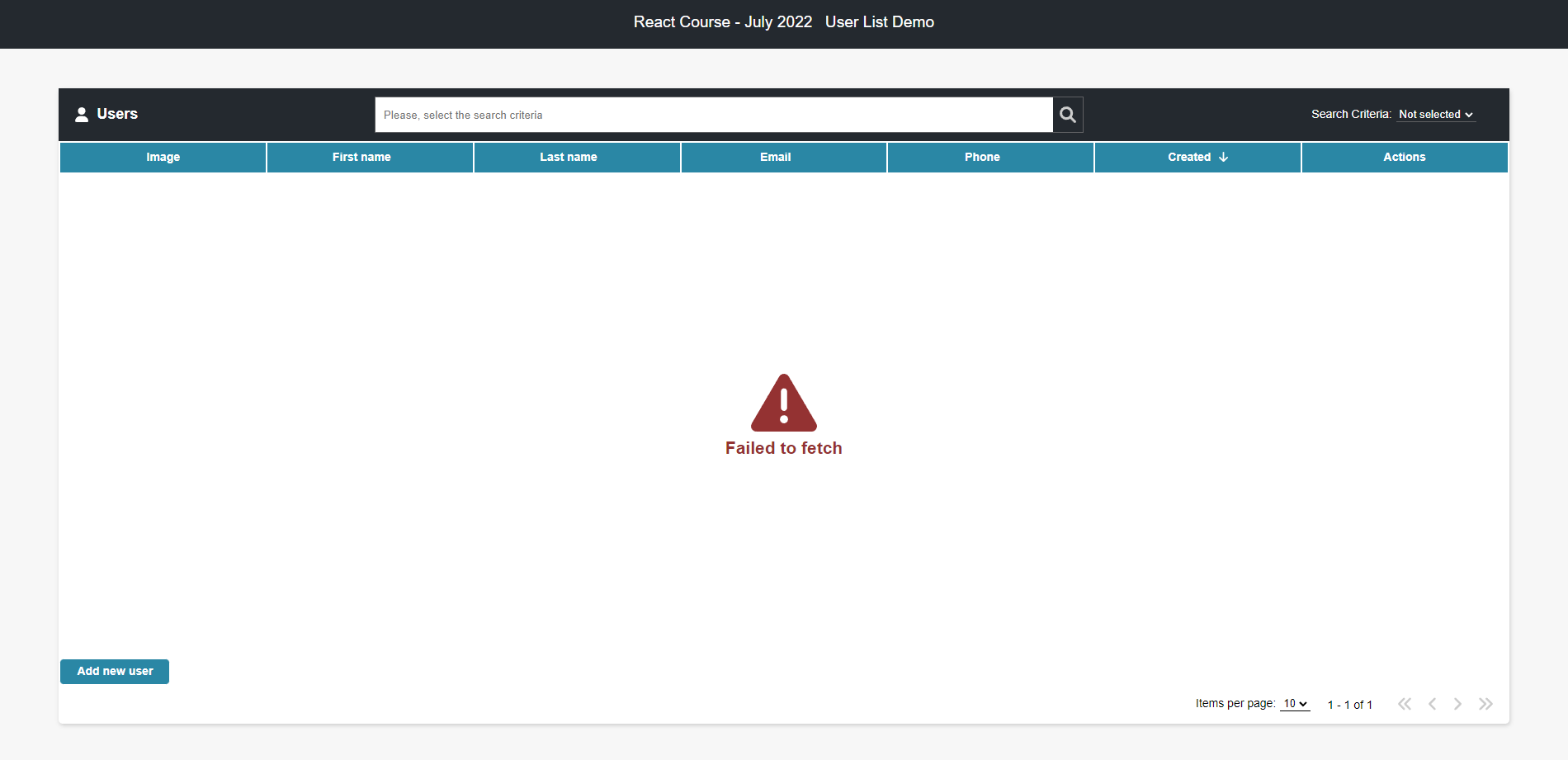
While waiting a response from a request, a spinner should be displayed:

Картина, която съдържа текст, екранна снимка, монитор

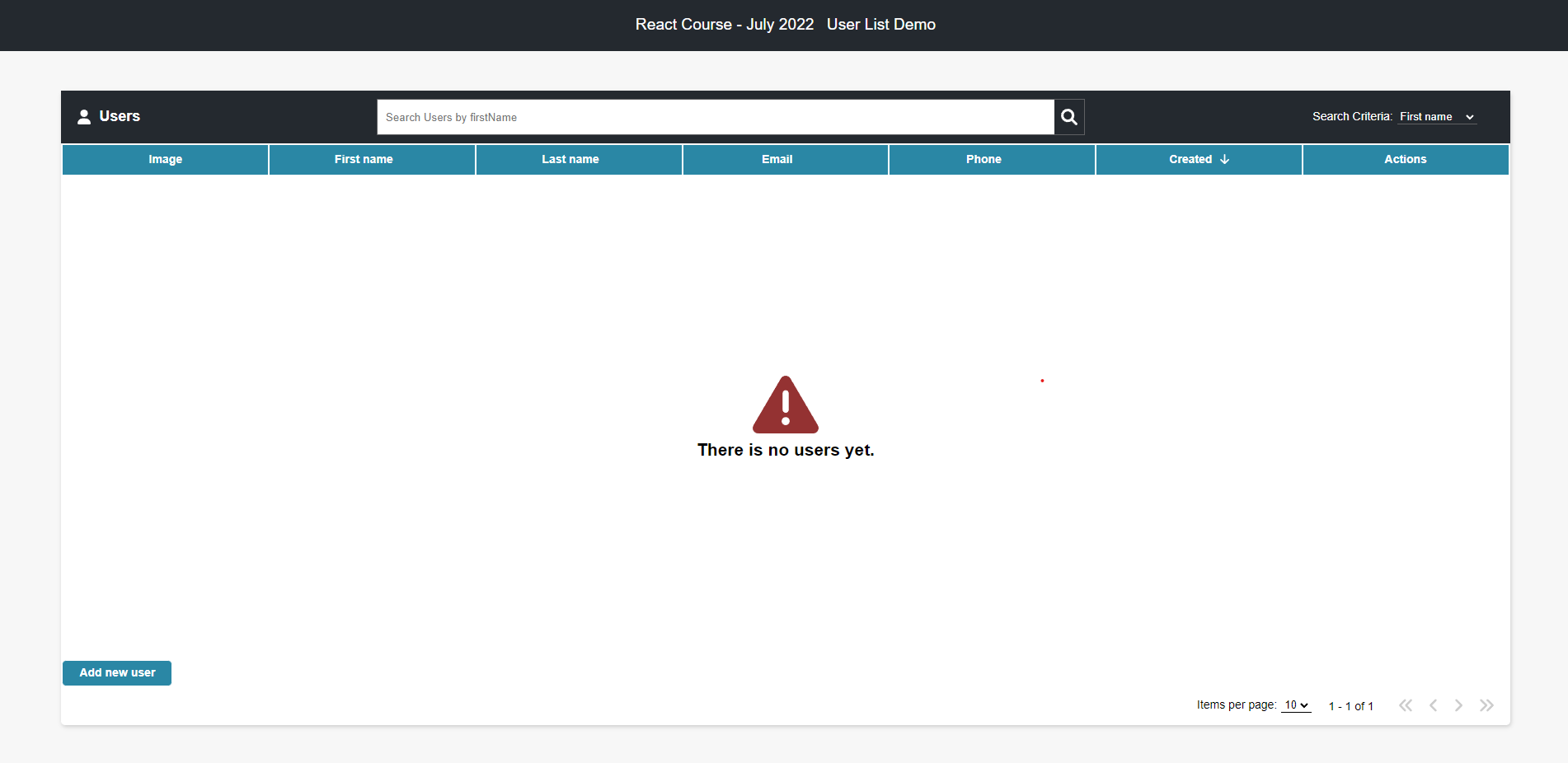
Описанието е генерирано автоматично

### Errors

If a request failed, the following error should be displayed:



If there are no records yet, the following error should be displayed:



If there is no record matching the given search criterion the following error should be displayed:

Картина, която съдържа текст

Описанието е генерирано автоматично

## Initialize the App

Use the **"npx create-react-app user-list-demo"** command to create a new React App.

If you face any trouble, check the following link:  
<https://github.com/facebook/create-react-app#create-react-app-->

### Server

Use the provided server to make the needed requests so you can visualize their results into the **front-end**. If you are experiencing difficulties starting the server, need the right endpoints or see the expected request bodies, make sure to check its **documentation**: <https://github.com/MihailValkov/user-list-demo/blob/main/server/README.md>

## Implementation

Make sure you assign every **required** **attribute** to each Component, so the result is the same as the example. If at some point something is not right, be free to check the structure for each element style for each element from the provided [github link](https://github.com/MihailValkov/user-list-demo) for the demo.