Lesson 11

REACT OBJECTS AND LISTS

WORKING WITH STATE OBJECTS

Routing with state

State values: pageone.js

```
import React, { useState } from 'react';

export const Pageone = (props) => {
  const [firstname, setFirstname] = useState("");
  const [lastname, setLastname] = useState("");

  const handleOnSubmit = () => {
    props.history.push("/pagetwo", {firstname:firstname, lastname : lastname});
  }
}
```

> What if we have many form fields?

> Pass all state values

© 2021 MIU

State values: pageone.js

```javascript
    let page1 = (
    <form>
      <h3>Page 1</h3>
      <div>
        First name
          <input
          type="text"
          placeholder="First name"
          value={firstname}
          onChange={e => setFirstname(e.target.value)} />
      </div>
      <div>
        Last name
          <input
          type="text"
          placeholder="Last name"
          value={lastname}
          onChange={e => setLastname(e.target.value)} />
      </div>
      <button onClick={handleOnSubmit}>Next</button>
    </form>
  );
  return page1;
}
```
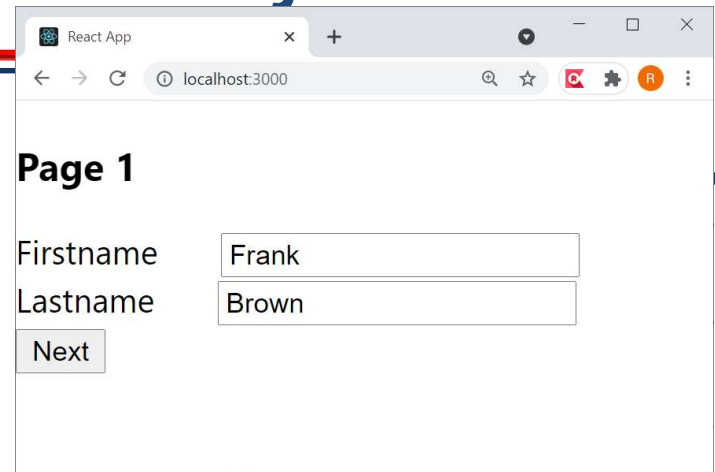
value

Set the value

React App × +

← → C ⓘ localhost:3000

Page 1

Firstname | Frank

Lastname | Brown

Next

State objects: pageone.js

```javascript
import React, { useState } from 'react';

export const Pageone = (props) => {
  const [user, setUser] = useState({
    firstname: "",
    lastname: "",
    address: "",
    creditcard: ""
  })

  const handleOnSubmit = () => {
    props.history.push("/pagetwo", {user :user});
  }
```

user is an object

Pass the whole object to the next page

Page 1

Firstname: Frank
Lastname: Brown
Next

First name: Frank
Last name: Brown

Page 2

Address: Address
Next

© 2021 MIU

State objects: pageone.js

```
let page1 = (
 <form>
   <h3>Page 1</h3>
   <div>
     Firstname
       <input
       type="text"
       placeholder="First name"
       name="firstname"
       onChange={e => setUser({...user,[e.target.name]: e.target.value })} />
   </div>
   <div>
     Lastname
       <input
       type="text"
       placeholder="Last name"
       name="lastname"
       onChange={e => setUser({...user,[e.target.name]: e.target.value,})} />
   </div>
   <button onClick={handleOnSubmit}>Next</button>
 </form>

);
return page1;
}
```

name

Page 1

Firstname   Frank

Lastname   Brown

Next

Update user object, leave all fields as is but update the field firstname with the value "Frank"
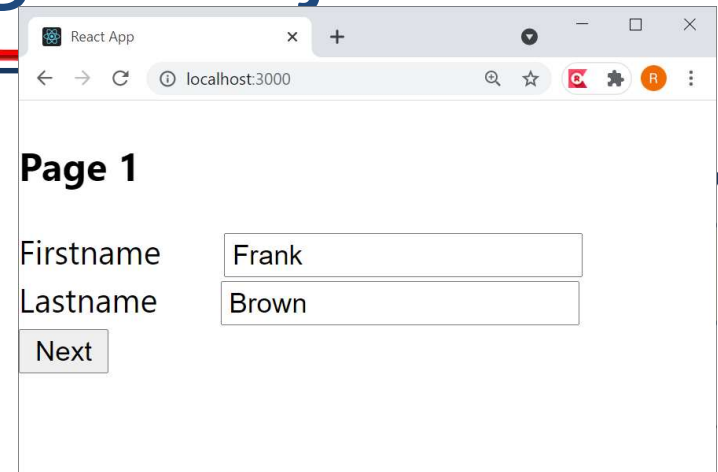
State objects: pageone.js

```
let page1 = (
 <form>
   <h3>Page 1</h3>
   <div>
     Firstname
       <input
       type="text"
       placeholder="First name"
       name="firstname"
       onChange={e => setUser({...user,[e.target.name]: e.target.value })} />
   </div>
   <div>
     Lastname
       <input
       type="text"
       placeholder="Last name"
       name="lastname"
       onChange={e => setUser({...user,[e.target.name]: e.target.value,})} />
   </div>
   <button onClick={handleOnSubmit}>Next</button>
 </form>
);
return page1;
}
```

Same method

React App

localhost:3000

Page 1

Firstname   Frank

Lastname   Brown

Next

State objects: pageone.js

```javascript
let page1 = (
  <form>
    <h3>Page 1</h3>
    <div>
      Firstname
        <input
        type="text"
        placeholder="First name"
        name="firstname"
        onChange={e => setUser({...user,[e.target.name]: e.target.value })} />
    </div>
    <div>
      Lastname
        <input
        type="text"
        placeholder="Last name"
        name="lastname"
        onChange={e => setUser({...user,[e.target.name]: e.target.value,})} />
    </div>
    <button onClick={handleOnSubmit}>Next</button>
  </form>

);
return page1;
}
```
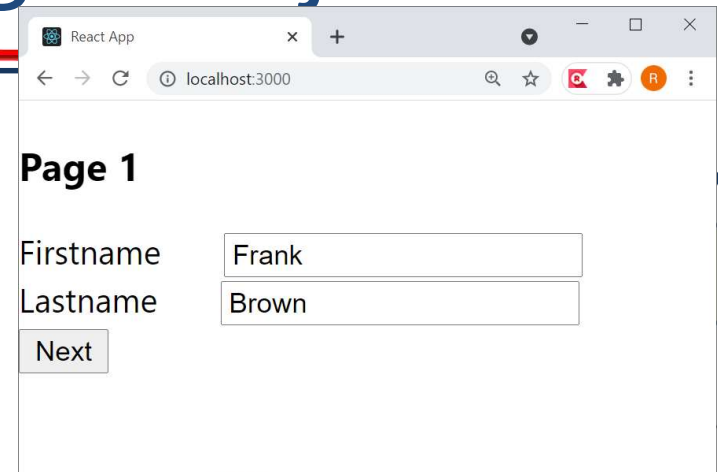
Same method

Page 1

| Firstname | Frank |
| Lastname | Brown |

Next

State objects: pageone.js

```
const [user, setUser] = useState({
  firstname: "",
  lastname: "",
  address: "",
  creditcard: ""
})
…
const handleFieldChange = (e) => {
 setUser({...user,[e.target.name]: e.target.value });
}
let page1 = (
    …
    <div>
      Firstname
        <input
        …
        name="firstname"
        onChange={handleFieldChange} />
    </div>
    <div>
      Lastname
        <input
        …
        name="lastname"
        onChange={handleFieldChange} />
    </div>
```

Define function

Call function

Call function

State objects: pagetwo.js

```javascript
export const Pagetwo = (props) => {
  const [user, setUser] = useState(props.location.state.user);
  const handleOnSubmit = () => {
    props.history.push("/pagethree", {user :user});
  }
  const handleFieldChange = (e) => {
    setUser({...user,[e.target.name]: e.target.value });
  }
  let page2 = (
    <div>
      <div>First name: {user.firstname}</div>
      <div>Last name: {user.lastname}</div>
      <form>
        <h3>Page 2</h3>
        <div>
          Address
          <input
            type="text"
            placeholder="Address"
            name="address"
            onChange={handleFieldChange} />
        </div>
        <button onClick={handleOnSubmit}>Next</button>
      </form>
    </div>
  );
  return page2;
```

Create a local user object based on the state of the passed user object

First name: John
Last name: Doe

Page 2

Address   Mainstreet 1

Next

WORKING WITH LISTS

List of strings

```jsx
import React, { useState } from 'react';

export const ToDoList = () => {

  const todoList = [
    'Learn JavaScript',
    'Learn React',
    'Learn Spring Boot',
  ];

  return (
    <div>
      <h1>Todo list</h1>
      <table>
        <tbody>
          {todoList.map(item => (
            <tr key={item}>
              <td>{item}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
};
```
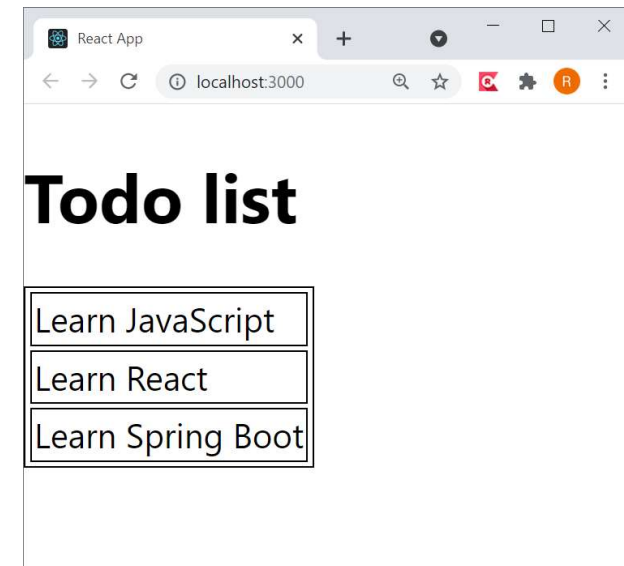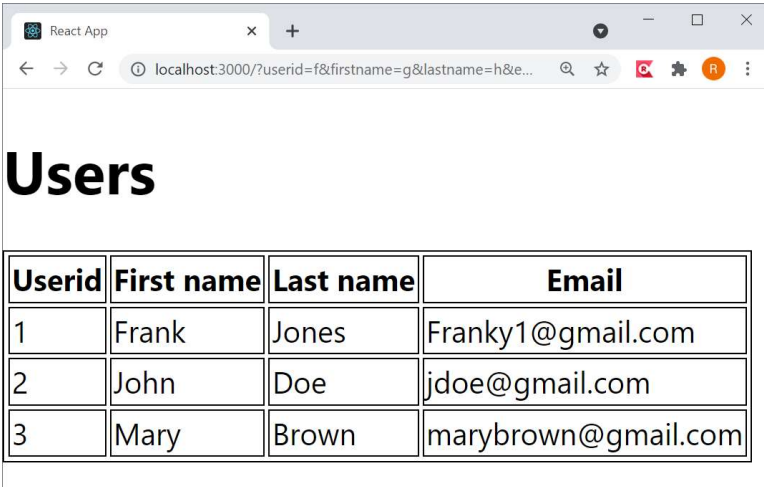
Map expects a unique key

item is the key

React App — localhost:3000

Todo list

| Learn JavaScript |
| Learn React |
| Learn Spring Boot |

List of users

```
export const UsersList = () => {
  const initialList = [
  { userid: "1", firstname: "Frank", lastname: "Jones", email: "Franky1@gmail.com" },
  { userid: "2", firstname: "John", lastname: "Doe", email: "jdoe@gmail.com" },
  { userid: "3", firstname: "Mary", lastname: "Brown", email: "marybrown@gmail.com" }
  ];
  const [userlist, setUserlist] = useState(initialList);
  return (
    <div>
      <h1>Users</h1>
      <table>
        <thead>
          <tr><th>Userid</th><th>First name</th><th>Last name</th><th>Email</th></tr>
        </thead>
        <tbody>
          {userlist.map(user => (
            <tr key={user.userid}>
              <td>{user.userid}</td>
              <td>{user.firstname}</td>
              <td>{user.lastname}</td>
              <td>{user.email}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
```
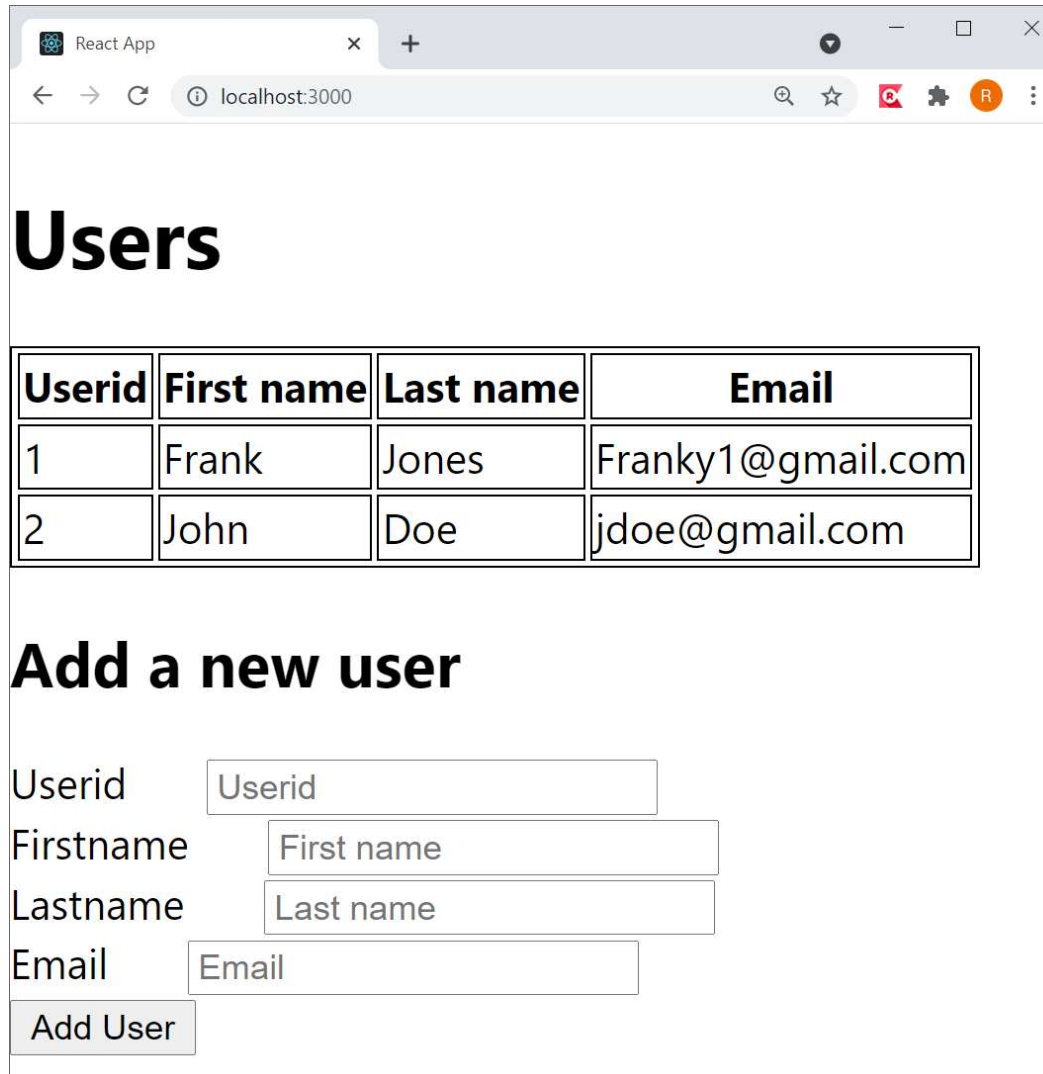
Userid is the key

Users

Userid	First name	Last name	Email
1	Frank	Jones	Franky1@gmail.com
2	John	Doe	jdoe@gmail.com
3	Mary	Brown	marybrown@gmail.com

MODIFY THE LIST IN THE SAME PAGE

Add to a list of objects

UsersList.js(1/4)

```javascript
import React, { useState } from 'react';

export const UsersList = () => {
  const cleanuser = { userid: "", firstname: "", lastname: "", email: "" };
  const [user, setUser] = useState(cleanuser);
  const initialList = [
    { userid: "1", firstname: "Frank", lastname: "Jones", email: "Franky1@gmail.com" },
    { userid: "2", firstname: "John", lastname: "Doe", email: "jdoe@gmail.com" },
  ];
  const [userlist, setUserlist] = useState(initialList);

  const handleSubmit = (e) => {
    if (user) {
      setUserlist(userlist.concat(user)); //add user to the list
    }
    //clear user
    setUser(cleanuser);
    //prevent POST request
    e.preventDefault();
  }

  const handleFieldChange = (e) => {
    setUser({ ...user, [e.target.name]: e.target.value });
  }
```

One users

List of users

Called when you click the "Add user" button

Do not send a POST request

Called when you change one of the form fields

Users

Userid	First name	Last name	Email
1	Frank	Jones	Franky1@gmail.com
2	John	Doe	jdoe@gmail.com

Add a new user

Userid   [Userid]
Firstname   [First name]
Lastname   [Last name]
Email   [Email]
[Add User]

UsersList.js(2/4)

```
return (
  <div>
    <h1>Users</h1>

    <table>
      <tr><th>Userid</th><th>First name</th><th>Last name</th><th>Email</th></tr>
      {userlist.map(user => (
        <tr><td>{user.userid}</td>
          <td>{user.firstname}</td>
          <td>{user.lastname}</td>
          <td>{user.email}</td></tr>
      ))}
    </table>
```

Show table of users

Users

Userid	First name	Last name	Email
1	Frank	Jones	Franky1@gmail.com
2	John	Doe	jdoe@gmail.com

Add a new user

Userid   [Userid]
Firstname   [First name]
Lastname   [Last name]
Email   [Email]
[Add User]

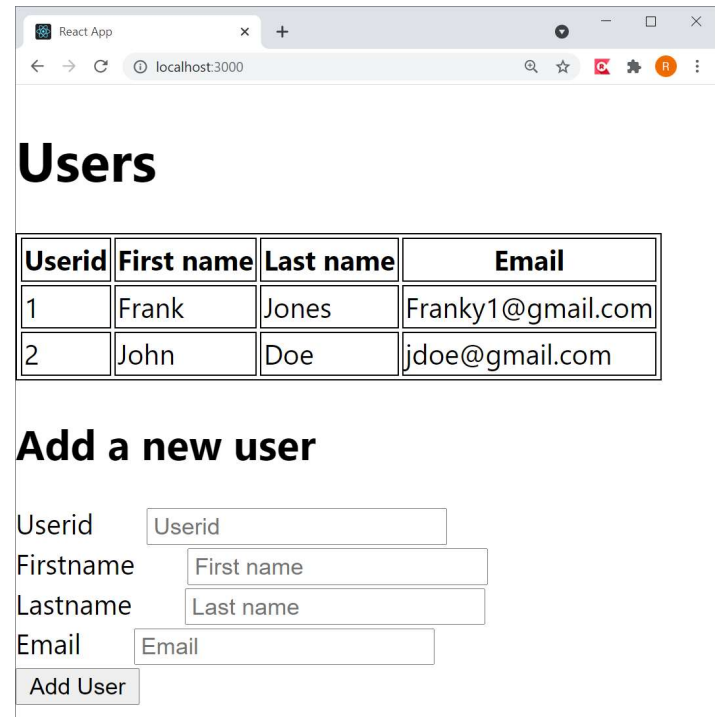# UsersList.js(3/4)

```
<p><h2>Add a new user</h2></p>
<form onSubmit={handleSubmit}>
  <div>
    Userid
    <input
      type="text"
      placeholder="Userid"
      name="userid"
      value={user.userid}
      onChange={handleFieldChange} />
  </div>
  <div>
    Firstname
    <input
      type="text"
      placeholder="First name"
      name="firstname"
      value={user.firstname}
      onChange={handleFieldChange} />
  </div>
```

Call handleSubmit when you click the "Add user" button

Users

Userid	First name	Last name	Email
1	Frank	Jones	Franky1@gmail.com
2	John	Doe	jdoe@gmail.com

Add a new user

Userid   Userid

Firstname   First name

Lastname   Last name

Email   Email

Add User

UsersList.js(4/4)

```jsx
      <div>
        Lastname
        <input
          type="text"
          placeholder="Last name"
          name="lastname"
          value={user.lastname}
          onChange={handleFieldChange} />
      </div>
      <div>
        Email
        <input
          type="text"
          placeholder="Email"
          name="email"
          value={user.email}
          onChange={handleFieldChange} />
      </div>
      <button type="submit">Add User</button>
    </form>
  </div>
  );
};
```
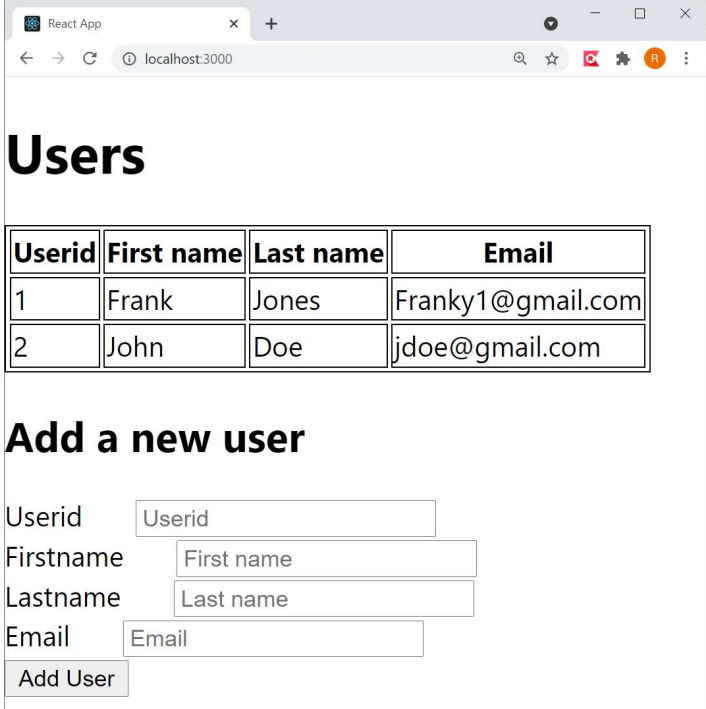
Submit button

Remove from a list of objects

UsersList.js

```javascript
const removeUser = (e) => {
    const newUserlist = userlist.filter((user) => user.userid !== e.target.value);
    setUserlist(newUserlist);
}
```

Create a new list and filter out the user to be removed

```html
    <table>
       <tr><th>Userid</th><th>First name</th><th>Last name</th><th>Email</th></tr>
       {userlist.map(user => (
         <tr><td>{user.userid}</td>
            <td>{user.firstname}</td>
            <td>{user.lastname}</td>
            <td>{user.email}</td>
            <td><button onClick={removeUser} value={user.userid} >Remove</button></td>
         </tr>
       ))}
    </table>
```

Add button to the table

Call removeUser function when clicked

Pass userid as value in event

Userid	First name	Last name	Email	
1	Frank	Jones	Franky1@gmail.com	Remove
2	John	Doe	jdoe@gmail.com	Remove

MODIFY THE LIST IN SEPARATE PAGES

Add to a list of objects

App.js

```
function App() {
  const [userlist, setUserlist] = useState([]);

  return (
    <div >
      <Router>
        <Route
          exact path='/'
          render={(props) => (
            <UsersList {...props} userlist={userlist} setUserlist={setUserlist} />
          )}
        />
        <Route
          path='/adduser'
          render={(props) => (
            <AddUser {...props} userlist={userlist} setUserlist={setUserlist} />
          )}
        />
      </Router>
    </div>
  );
}

export default App;
```
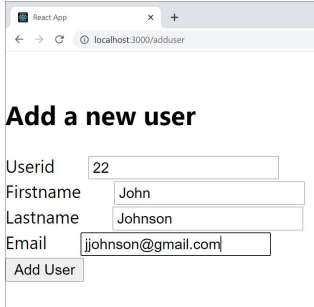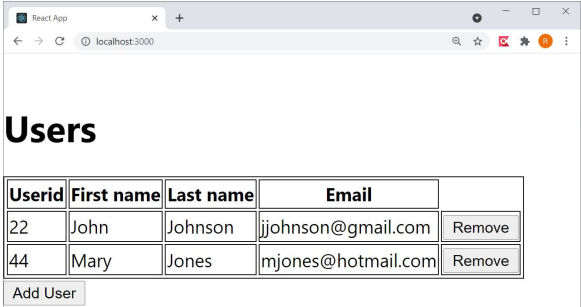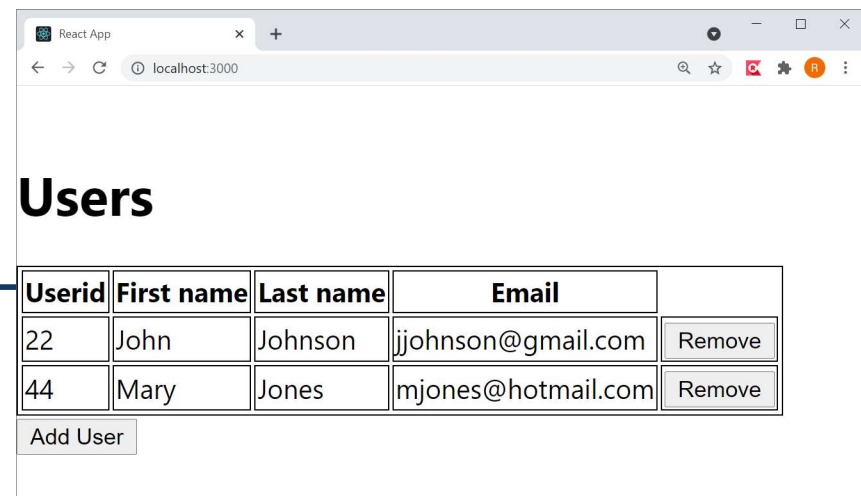
List of users is in the top component

Pass **userlist** and **setUserlist** function to the child components

UsersList.js(1/2)

```javascript
import React, { useState } from 'react';

export const UsersList = (props) => {
  const cleanuser = { userid: "", firstname: "", lastname: "", email: "" };
  const [user, setUser] = useState(cleanuser);


  const handleAddUser = () => {
    props.history.push("/adduser");//navigate to add user
  }


  const removeUser = (e) => {
    const newUserlist = props.userlist.filter((user) => user.userid !== e.target.value);
    props.setUserlist(newUserlist);
  }
```
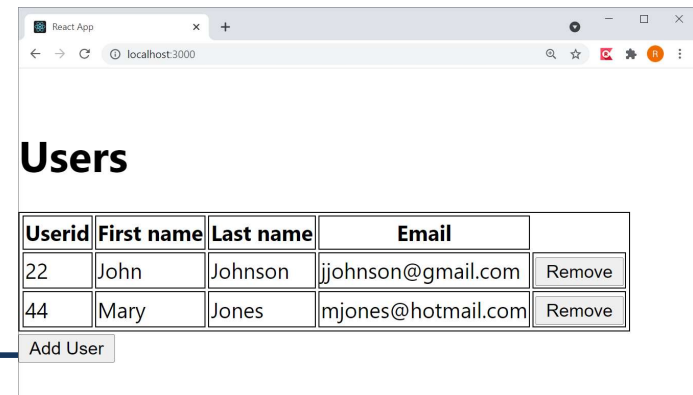
Users

Userid	First name	Last name	Email	
22	John	Johnson	jjohnson@gmail.com	Remove
44	Mary	Jones	mjones@hotmail.com	Remove

Add User

UsersList.js(1/2)

```javascript
return (
  <div>
    <h1>Users</h1>
    <table>
      <thead><tr><th>Userid</th><th>First name</th><th>Last name</th><th>Email</th>
       </tr></thead>
      <tbody>
        {props.userlist.map(user => (
          <tr key={user.userid}>
            <td>{user.userid}</td>
            <td>{user.firstname}</td>
            <td>{user.lastname}</td>
            <td>{user.email}</td>
            <td><button onClick={removeUser} value={user.userid} >Remove</button></td
>
          </tr>
        ))}
      </tbody>
    </table>
    <button onClick={handleAddUser}>Add User</button>
  </div>
);
};
```
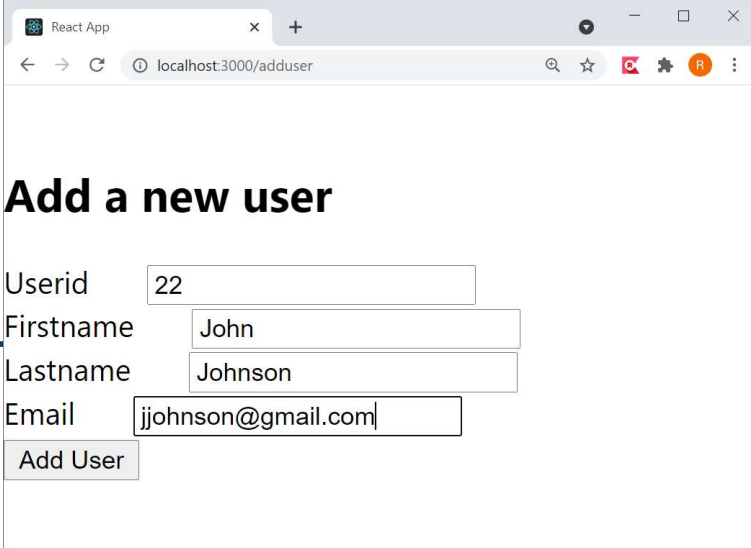

Users

Userid	First name	Last name	Email	
22	John	Johnson	jjohnson@gmail.com	Remove
44	Mary	Jones	mjones@hotmail.com	Remove

Add User

AddUser.js(1/2)

```javascript
export const AddUser = (props) => {
  const cleanuser = { userid: "", firstname: "", lastname: "", email: "" };
  const [user, setUser] = useState(cleanuser);

  const handleSubmit = (e) => {
    props.setUserlist(props.userlist.concat(user)); //add user to the list
    props.history.push("/"); //navigate to userlist
  }

  const handleFieldChange = (e) => {
    setUser({ ...user, [e.target.name]: e.target.value });
  }
```

AddUser.js(2/2)
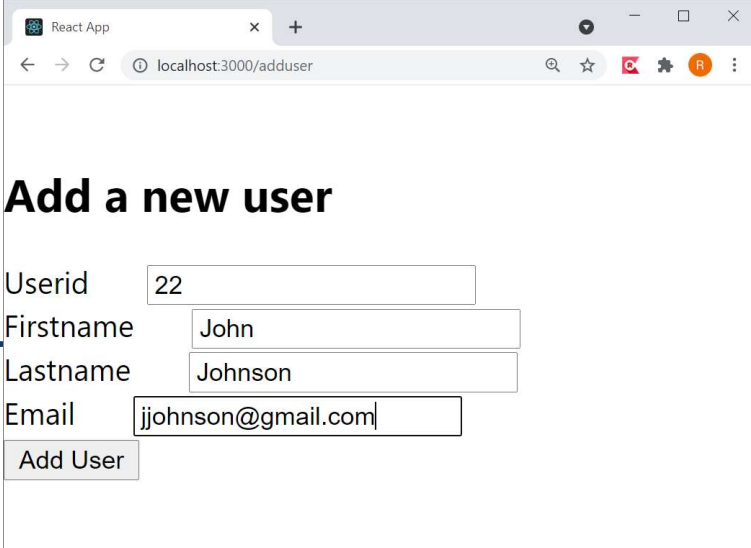
```
return (
    <div>
      <h2>Add a new user</h2>
      <form onSubmit={handleSubmit}>
        <div>
          Userid
          <input
            type="text"
            placeholder="Userid"
            name="userid"
            value={user.userid}
            onChange={handleFieldChange} />
        </div>
          …
        <button type="submit">Add User</button>
      </form>
    </div>
  );
};
```

Add a new user

Userid	22
Firstname	John
Lastname	Johnson
Email	jjohnson@gmail.com

Add User