유다현 과제

> 2022-05-19

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import {BrowserRouter} from "react-router-dom";

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
    <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

App.js

```
import React from "react";
import {Routes,Route} from "react-router-dom";
import List from "./pages/List";
import Add from "./pages/Add";
import Edit from "./pages/Edit";


function App() {
  return (
    <div>
        <Routes>
        <Route path="/" exapt={true} element={<List/>}></Route>
        <Route path="/add" element={<Add/>}></Route>
        <Route path="/edit/:id" element={<Edit/>}></Route>
      </Routes>
    </div>
  );
}

export default App;
```

pages/List.js

```
import React from 'react';
import useAxios from "axios-hooks";
import styled from "styled-components";
import {Link} from "react-router-dom";

import Spinner from "../component/Spinner";
import Table from "../component/Table";
import dayjs from "dayjs";
const LinkContainer = styled.div`
    position:sticky;
    top:0;
    background-color  : #fff;
    border-top:1px solid #eee;
    border-bottom:1px solid #eee;
    padding:10px 0;
`;

const TopLink  = styled(Link)`
    margin-right:15px;
    display:inline-block;
    font-size:16px;
    padding:7px 10px 5px 10px;
    border:1px solid #ccc;
    background-color: #fff;
    color:#000;
    text-decoration: none;
    &:hover{
        background-color: #06f2;
    }
`

const List = () => {

      /* 화면에 표시할 성적표 데이터를 저장하기 위한 상태 변수 */
      const [professor , setProfessor] = React.useState([]);

      /*백엔드로부터 데이터 불러오기 - 자체캐시기능 방지*/
      const [{data,loading1,error}]=
useAxios("http://localhost:3001/professor",{
          useCache:false
      });

      React.useEffect(() =>{
        //ajax의 결과를 화면에 표시하기 위한 상태값에 복사한다.
        setProfessor(data);
    },[data]);


    const [{loading2},sendDelete] = useAxios({
        method :"DELETE"
    },{
        useCache: false,
        manual :true
    });
  // 삭제버튼 클릭 시 호출될 이벤트 핸들러
  const onDeleteClick =(e)=>{
    e.preventDefault();
    //이벤트 발생한 대상 가져옴 (삭제하기 링크)
    const current = e.target;

    //클릭된 링크에 숨겨져 있는 일련번호와 학생이름 가져오기
    const id = parseInt(current.dataset.id);
    const name = current.dataset.name;


    if(window.confirm(`정말 ${name}교수의 정보를 삭제하시겠습니까?`)){

        //백엔드에 삭제 요청하기 - > 입력,수정,삭제는 async~await 문법으로 처리해야 함
        ( async ()=>{
                let json = null;
                try{
                    const response = await sendDelete({
                        method:"DELETE",
                        url:`http://localhost:3001/professor/${id}`
                    });
                    json = response.data;
                }catch(e){
                    console.error(e);

window.alert(`[${e.response.status}]${e.response.statusText}${e.message}}`
);

                }
                //삭제 잘될 경우
            if (json != null) {
                setProfessor(professorr => professor.filter((v, i) =>v.id
!== id));
            }
        })();
    }

}

return (
    <>
    <Spinner visible={loading1 || loading2}/>

    <LinkContainer>
        <TopLink to="add">
            교수 등록하기
        </TopLink>
    </LinkContainer>
    {
        error ? (
            <div>
                <h1>{error.code} Error.</h1>
                <p>{error.message}</p>
            </div>
        ) : (
            <Table>
                <thead>
                    <tr>
                        <td>No.</td>
                        <td>이름</td>
                        <td>아이디</td>
                        <td>직급</td>
                        <td>급여</td>
                        <td>입사일</td>
                        <td>보직수당</td>
                        <td>소속학과번호</td>
                        <td>수정</td>
                        <td>삭제</td>
                    </tr>
                </thead>
                <tbody>
                    {
                        professor &&
professor.map(({id,name,userid,position,sal,hiredate,comm,deptno},i)=>{
                            return(
                                <tr key={id}>
                                    <td>{id}</td>
                                    <td>{name}</td>
                                    <td>{userid}</td>
                                    <td>{position}</td>
                                    <td>{sal}만원</td>
                                    <td>
{dayjs(hiredate).format('YY/MM/DD')}</td>
                                    <td>{comm && `${comm}만원`}</td>
                                    <td>{deptno}</td>
                                    <td>
                                        <Link to={`edit/${id}`}>수정하기
</Link>
                                    </td>
                                    <td>
                                        <a href="#!" data-id={id} data-
name={name} onClick={onDeleteClick}>삭제하기</a>
                                    </td>
                                </tr>
                            )
                        })
                    }
                </tbody>
            </Table>
        )
    }
    </>
);
};

export default List;
```

pages/Add.js

```
import React from 'react';
import useAxios from "axios-hooks";
import styled from "styled-components";
import { useNavigate } from 'react-router-dom';
import dayjs from "dayjs";

import Spinner from "../component/Spinner";
import Table from "../component/Table";

import regexHelper from "../libs/RegexHelper";


const TableEx =styled(Table)`
    margin-top: 50px;
    margin-bottom: 15px;
    .inputWrapper{
        padding:0;
        position: relative;
        text-align: left;
        .field{
            box-sizing:border-box;
            display:block;
            position:absolute;
            top: 0;
            left:0;
            width:100%;
            height:100%;
            border:0;
            padding:0 10px;
            outline:none;
            font-size:14px;
        }
        label{
            margin-left:7px;
            margin-right:10px;
            input{
                margin-right: 10px;
            }
        }
    }
`;
```

```
const Add = () => {
     /* 저장 완료 후 목록으로 되돌아가기 위한 페이지 강제 이동 함수 생성*/
     const navigate = useNavigate();

     /* 드롭다운에 표시할 학과 목록 조회*/
     const [{data , loading:loading1 ,error}]=useAxios("http://localhost:3001/department");

     const [{ loading: loading2 }, refetch] = useAxios({
        url: "http://localhost:3001/professor",
        method: 'POST'
    }, { manual: true });

    const onSubmit = React.useCallback((e)=>{
        e.preventDefault();

        const current =e.target;

        try{
            regexHelper.value(current.name,"이름을 입력하세요");
            regexHelper.kor(current.name,"이름을 한글로 입력하세요");
            regexHelper.minLength(current.name,2,"이름은 최소 2글자 이상 입력해주세요.");

            regexHelper.value(current.userid,"아이디을 입력하세요");
            regexHelper.minLength(current.userid,2,"아이디은 최소 2글자 이상 입력해주세요.");
            regexHelper.maxLength(current.userid,20,"아이디은 최소 20글자 미만 입력해주세요.");

            regexHelper.check(current.position,"직급을 선택하세요");

            regexHelper.value(current.sal,"급여를 입력해주세요.");
            regexHelper.num(current.sal,"급여는 숫자만 가능합니다.");

            regexHelper.value(current.hiredate,"입사일을 입력하세요.");

            regexHelper.value(current.comm,"보직수당을 입력하세요.");
            regexHelper.num(current.comm,"보직 수당은 숫자만 입력가능합니다.");

            regexHelper.num(current.deptno,"소속학과를 선택합니다.");

        }catch(e){
            window.alert(e.message);
            e.field.focus();
            return;
        }

        let json = null;

        (async ()=>{
            try{
                const response = await refetch({
                    data:{
                        name:current.name.value,
                        userid:current.userid.value,
                        position:current.position.value,
                        sal:parseInt(current.sal.value),

hiredate:dayjs(current.hiredate.value).toISOString(),
                        comm: parseInt(current.comm.value),
                        deptno: parseInt(current.deptno.value)
                    }
                });
                json = response.data;
            }catch(e){
                console.error(e);
                window.alert(e);
            }
            if(json !== null){
                navigate("/");
            }
        })();
    },[refetch, navigate]);
    return (
        <>
           <Spinner visible={loading1 || loading2} />
           {error ? (
            <div>
                <h1>Oops~!!! {error.code} Error.</h1>
                <hr />
                <p>{error.message}</p>
            </div>) :(
              <form onSubmit={onSubmit}>
                     <TableEx>
                         <colgroup>
                            <col width="120"/>
                            <col/>
                         </colgroup>
                         <tbody>
                            <tr>
                                <th>이름</th>
                                <td className="inputWrapper">
                                    <input type="text" className="field"
name="name"/>
                                </td>
                            </tr>
                            <tr>
                                <th>아이디</th>
                                <td className="inputWrapper">
                                    <input type="text" className="field"
name="userid"/>
                                </td>
                            </tr>
                            <tr>
                                <th>직급</th>
                                <td className="inputWrapper">
```

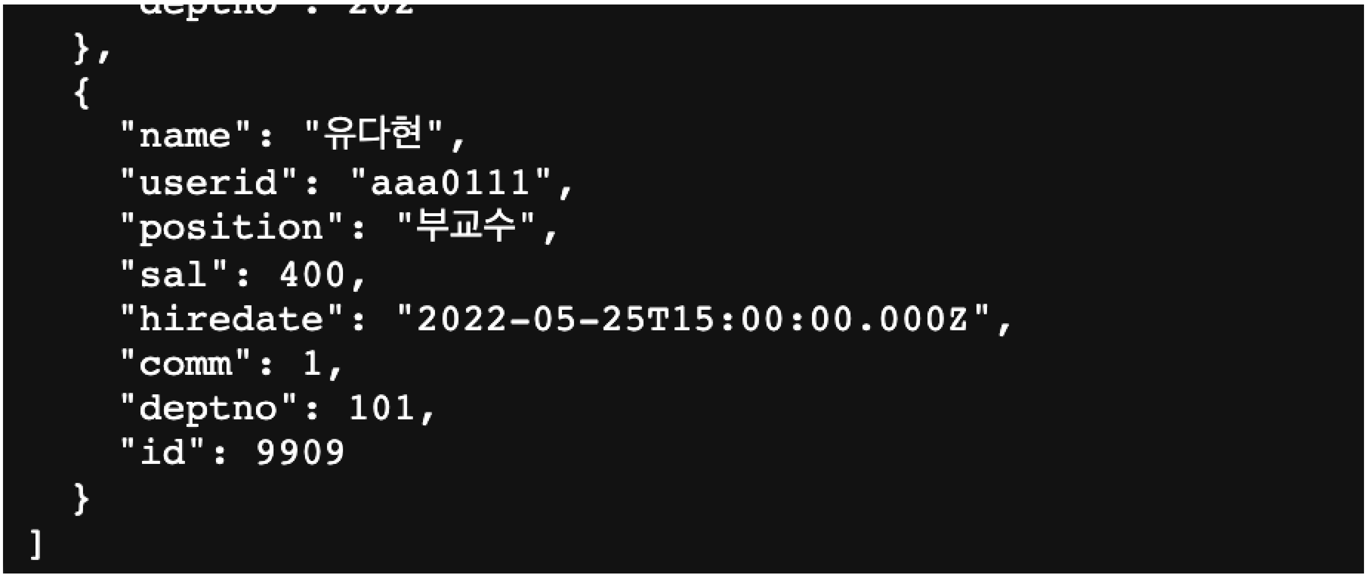
```
                                    <lable><input type="radio"
name="position" value="교수"/>교수</lable>
                                    <lable><input type="radio"
name="position" value="부교수"/>부교수</lable>
                                    <lable><input type="radio"
name="position" value="조교수"/>조교수</lable>
                                    <lable><input type="radio"
name="position" value="전임강사"/>전임강사</lable>
                                </td>
                            </tr>
                            <tr>
                                <th>급여(만원)</th>
                                <td className="inputWrapper">
                                    <input type="number" className="field"
name="sal" placeholder="숫자로 입력하세요"/>
                                </td>
                            </tr>
                            <tr>
                                <th>입사일</th>
                                <td className="inputWrapper">
                                    <input type="date" className="field"
name="hiredate"/>
                                </td>
                            </tr>
                            <tr>
                                <th>보직수당</th>
                                <td className="inputWrapper">
                                    <input type="text" className="field"
name="comm"/>
                                </td>
                            </tr>
                            <tr>
                                <th>소속학과</th>
                                <td className="inputWrapper">
                                  <select name="deptno">
                                      <option value="">선택하세요</option>
                                      {data && data.map ((v,i)=><option
key={i} value={v.id}>{v.dname}</option>)}
                                  </select>
                                </td>
                            </tr>

                         </tbody>
                     </TableEx>
                     <div style={{ textAlign: 'center' }}>
                        <button type='submit'>저장하기</button>
                    </div>

                 </form>
                 )
                 }
        </>
    );
};
```

```
export default Add;
```

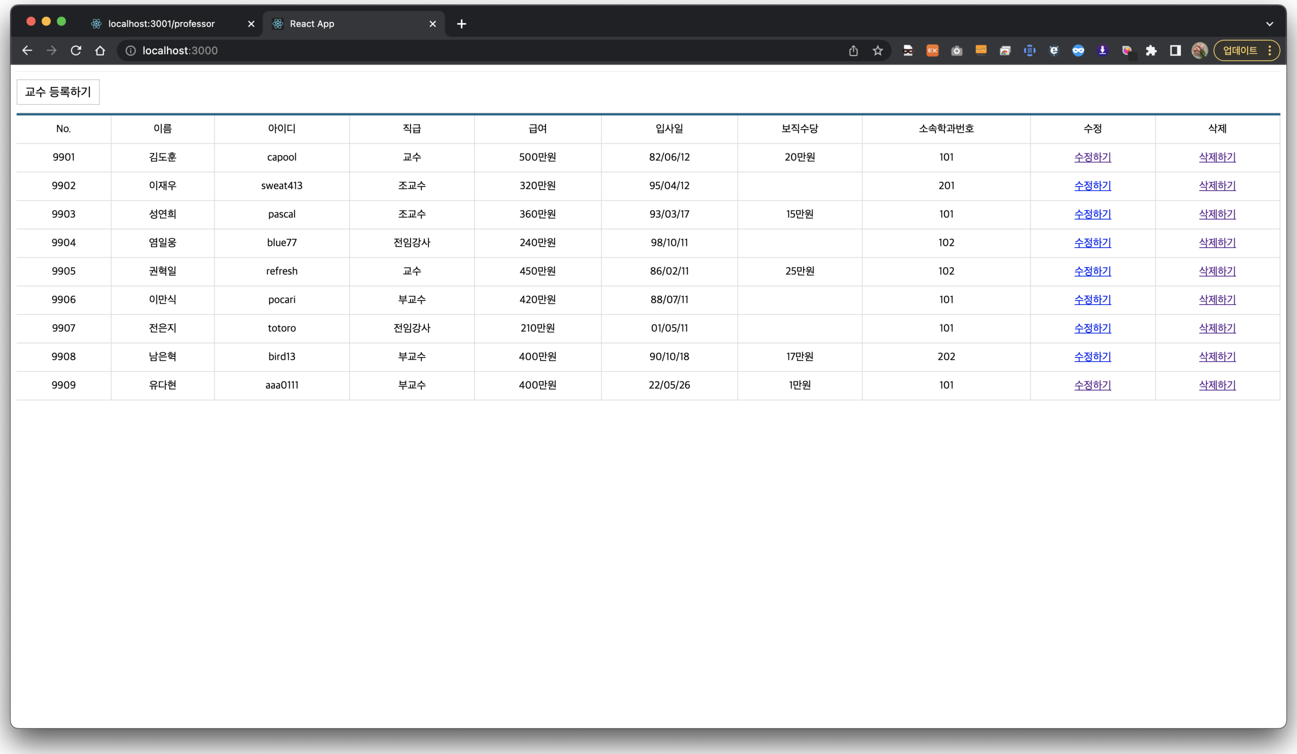
pages/Edit.js

```
import React from 'react';
import useAxios from "axios-hooks";
import styled from "styled-components";
import { useNavigate,useParams } from 'react-router-dom';
import dayjs from "dayjs";

import Spinner from "../component/Spinner";
import Table from "../component/Table";

import regexHelper from "../libs/RegexHelper";



const TableEx =styled(Table)`
    margin-top: 50px;
    margin-bottom: 15px;
    .inputWrapper{
        padding:0;
        position: relative;
        text-align: left;
        .field{
            box-sizing:border-box;
            display:block;
            position:absolute;
            top: 0;
            left:0;
            width:100%;
            height:100%;
            border:0;
            padding:0 10px;
            outline:none;
            font-size:14px;
        }
        label{
            margin-left:7px;
            margin-right:10px;
            input{
                margin-right: 10px;
            }
        }
    }
`;


const Edit = () => {
     /* 저장 완료 후 목록으로 되돌아가기 위한 페이지 강제 이동 함수 생성*/
     const {id} = useParams();

    const navigate = useNavigate();


    const [{data:department,loading:
loading1,error:error1}]=useAxios("http://localhost:3001/department");

    const [{data:professor,loading:loading2 ,
error:error2},refetch]=useAxios(`http://localhost:3001/professor/${id}`);

    const onSubmit = React.useCallback((e)=>{
        e.preventDefault();

        const current =e.target;

        try{
            regexHelper.value(current.name,"이름을 입력하세요");
            regexHelper.kor(current.name,"이름을 한글로 입력하세요");
            regexHelper.minLength(current.name,2,"이름은 최소 2글자 이상 입력해주
세요.");

            regexHelper.value(current.userid,"아이디을 입력하세요");
            regexHelper.minLength(current.userid,2,"아이디은 최소 2글자 이상 입력
해주세요.");
            regexHelper.maxLength(current.userid,20,"아이디은 최소 20글자 미만 입
력해주세요.");

            regexHelper.check(current.position,"직급을 선택하세요");

            regexHelper.value(current.sal,"급여를 입력해주세요.");
            regexHelper.num(current.sal,"급여는 숫자만 가능합니다.");

            regexHelper.value(current.hiredate,"입사일을 입력하세요.");

            regexHelper.value(current.comm,"보직수당을 입력하세요.");
            regexHelper.num(current.comm,"보직 수당은 숫자만 입력가능합니다.");

            regexHelper.num(current.deptno,"소속학과를 선택합니다.");

        }catch(e){
            window.alert(e.message);
            e.field.focus();
            return;
        }

        let json = null;

        (async ()=>{
            try{
                const response = await refetch({
                    method:"PUT",
                    data:{
                        name:current.name.value,
                        userid:current.userid.value,
                        position:current.position.value,
                        sal:parseInt(current.sal.value),

hiredate:dayjs(current.hiredate.value).toISOString(),
                        comm: parseInt(current.comm.value),
                        deptno: parseInt(current.deptno.value)
                    }
                });
                json = response.data;
            }catch(e){
                console.error(e);
                window.alert(e);
            }
            if(json !== null){
                window.alert("수정되었습니다.");
                navigate("/");
            }
        })();
    },[refetch, navigate]);

    return (
        <>
                <Spinner visible={loading1 || loading2} />
        {(error1 || error2) ? (
            <div>
                <h1>Oops~!!! {(error1 || error2).code} Error.</h1>
                <hr />
                <p>{(error1 || error2).message}</p>
            </div>
            ) : (
                professor && ( <form onSubmit={onSubmit}>
                 <TableEx>
                     <colgroup>
                        <col width="120"/>
                        <col/>
                     </colgroup>
                     <tbody>
                        <tr>
                            <th>이름</th>
                            <td className="inputWrapper">
                                <input type="text" className="field"
defaultValue={professor.name} name="name"/>
                            </td>
                        </tr>
                        <tr>
                            <th>아이디</th>
                            <td className="inputWrapper">
                                <input type="text" className="field"
defaultValue={professor.userid} name="userid"/>
                            </td>
                        </tr>
                        <tr>
                            <th>직급</th>
                            <td className="inputWrapper">
                                <lable><input type="radio" defaultChecked={professor.position ==="교수"} name="position" value="교수"/>교수</lable>
                                <lable><input type="radio" defaultChecked={professor.position ==="부교수"} name="position" value="부교수"/>부교수</lable>
                                <lable><input type="radio" defaultChecked={professor.position ==="조교수"} name="position" value="조교수"/>조교수</lable>
                                <lable><input type="radio" defaultChecked={professor.position ==="전임강사"} name="position" value="전임강사"/>전임강사</lable>
                            </td>
                        </tr>
                        <tr>
                            <th>급여(만원)</th>
                            <td className="inputWrapper">
                                <input type="number" className="field" name="sal" defaultValue={professor.sal} placeholder="숫자로 입력하세요"/>
                            </td>
                        </tr>
                        <tr>
                            <th>입사일</th>
                            <td className="inputWrapper">
                                <input type="text" className="field" defaultValue={professor.hiredate} name="hiredate"/>
                            </td>
                        </tr>
                        <tr>
                            <th>보직수당</th>
                            <td className="inputWrapper">
                                <input type="text" className="field" defaultValue={professor.comm} name="comm"/>
                            </td>
                        </tr>
                        <tr>
                            <th>소속학과</th>
                            <td className="inputWrapper">
                              <select name="deptno" defaultValue={professor.deptno}>
                                  <option value="">선택하세요</option>
                                  {department && department.map ((v,i)=> <option key={i} value={v.id}>{v.dname}</option>)}
                              </select>
                            </td>
                        </tr>

                     </tbody>
                 </TableEx>
                 <div style={{ textAlign: 'center' }}>
                    <button type='submit'>저장하기</button>
```

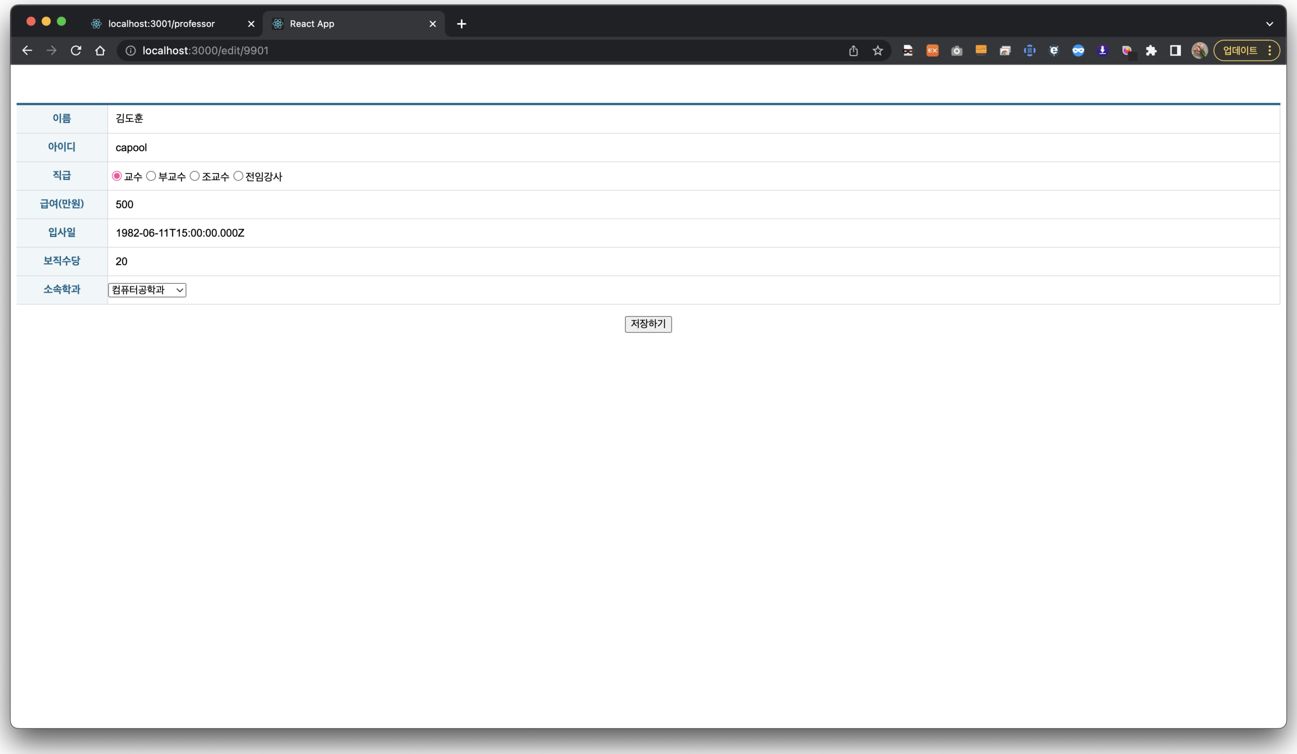
```
                </div>

             </form>
             )
        )}
    </>
    );
};

export default Edit;
```



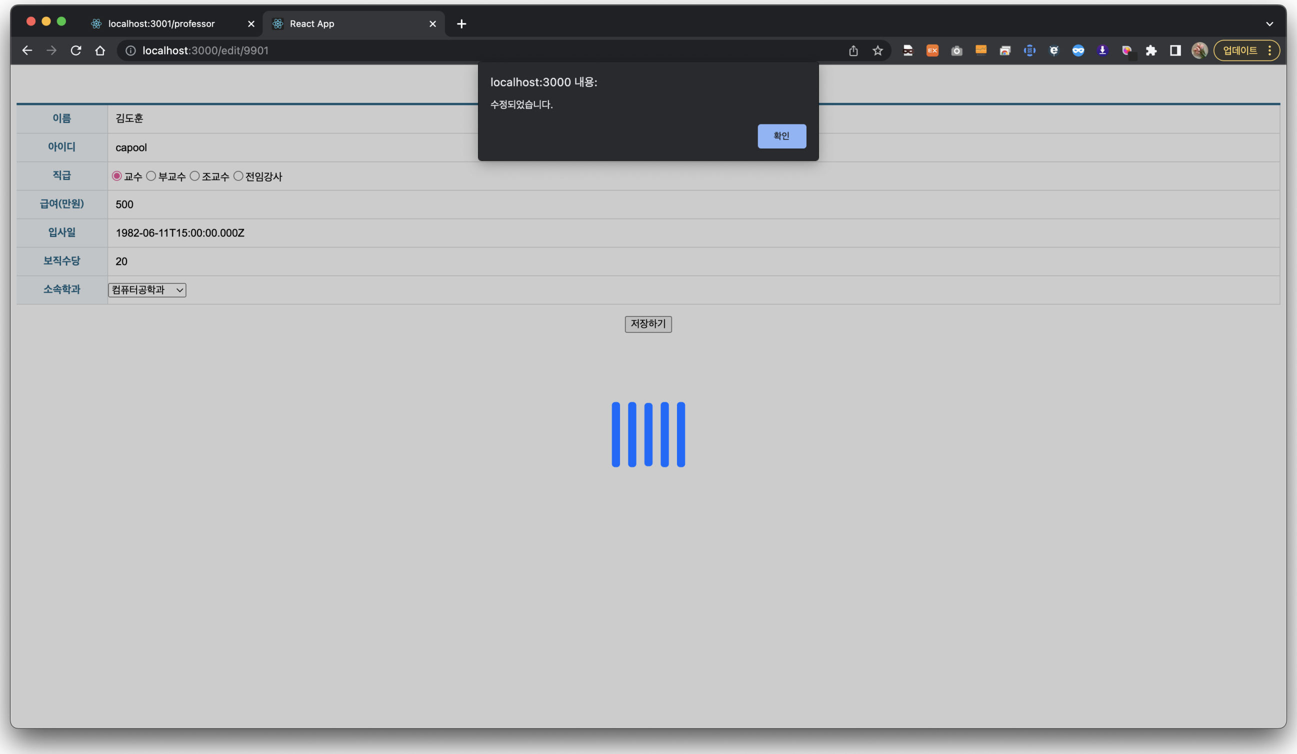
```
      "deptno": 202
  },
  {
    "name": "유다현",
    "userid": "aaa0111",
    "position": "부교수",
    "sal": 400,
    "hiredate": "2022-05-25T15:00:00.000Z",
    "comm": 1,
    "deptno": 101,
    "id": 9909
  }
]
```



No.	이름	아이디	직급	급여	입사일	보직수당	소속학과번호	수정	삭제
9901	김도훈	capool	교수	500만원	82/06/12	20만원	101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	유다현	aaa0111	부교수	400만원	22/05/26	1만원	101	수정하기	삭제하기



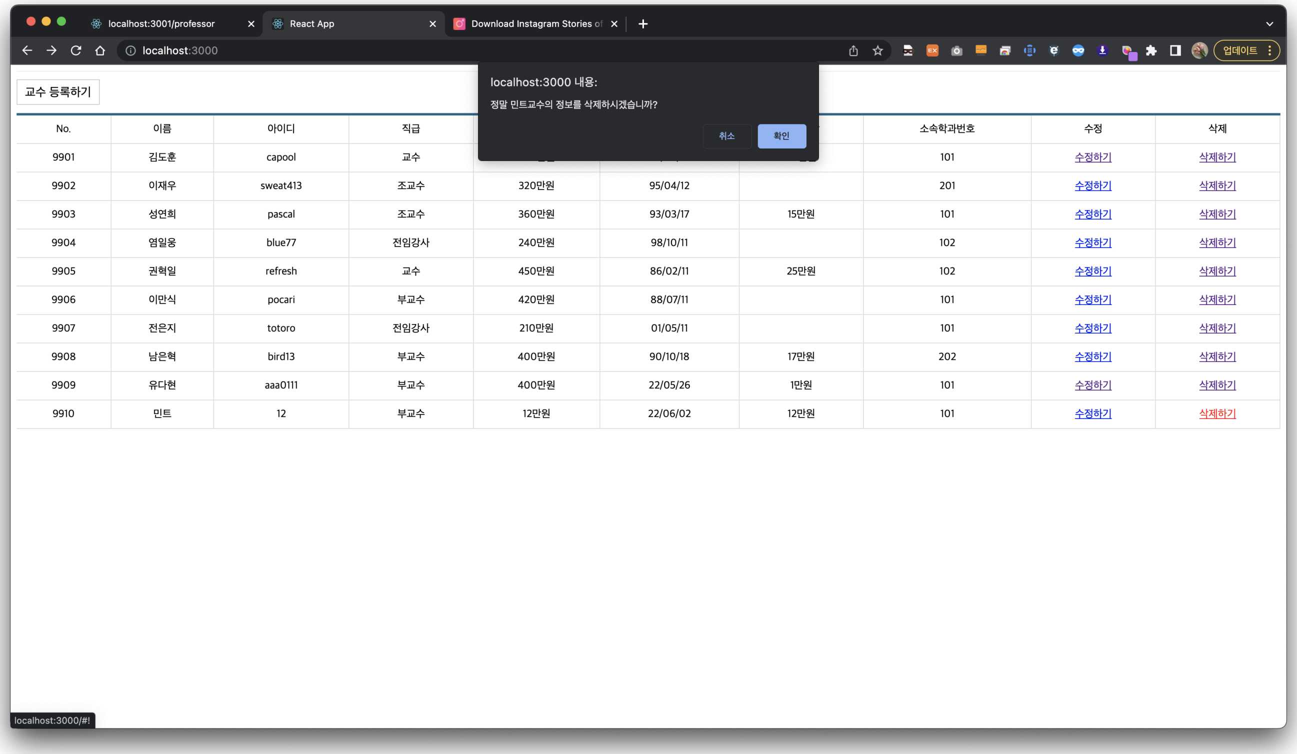
이름	김도훈
아이디	capool
직급	● 교수 ○ 부교수 ○ 조교수 ○ 전임강사
급여(만원)	500
입사일	1982-06-11T15:00:00.000Z
보직수당	20
소속학과	컴퓨터공학과



localhost:3000 내용:

수정되었습니다.

이름	김도훈
아이디	capool
직급	● 교수 ○ 부교수 ○ 조교수 ○ 전임강사
급여(만원)	500
입사일	1982-06-11T15:00:00.000Z
보직수당	20
소속학과	컴퓨터공학과



localhost:3000 내용:

정말 민트교수의 정보를 삭제하시겠습니까?

취소 확인

No.	이름	아이디	직급				소속학과번호	수정	삭제
9901	김도훈	capool	교수				101	수정하기	삭제하기
9902	이재우	sweat413	조교수	320만원	95/04/12		201	수정하기	삭제하기
9903	성연희	pascal	조교수	360만원	93/03/17	15만원	101	수정하기	삭제하기
9904	염일웅	blue77	전임강사	240만원	98/10/11		102	수정하기	삭제하기
9905	권혁일	refresh	교수	450만원	86/02/11	25만원	102	수정하기	삭제하기
9906	이만식	pocari	부교수	420만원	88/07/11		101	수정하기	삭제하기
9907	전은지	totoro	전임강사	210만원	01/05/11		101	수정하기	삭제하기
9908	남은혁	bird13	부교수	400만원	90/10/18	17만원	202	수정하기	삭제하기
9909	유다현	aaa0111	부교수	400만원	22/05/26	1만원	101	수정하기	삭제하기
9910	민트	12	부교수	12만원	22/06/02	12만원	101	수정하기	삭제하기