

통합구현 시험보고서

공통코드

```
import React, {memo} from 'react';
import {Routes, Route} from 'react-router-dom';

import ProfessorList from './pages/ProfessorList';
import ProfessorEdit from './pages/ProfessorEdit';
import ProfessorAdd from './pages/ProfessorAdd';
import ProfessorDetail from './pages/ProfessorDetail';

function App() {
  return (
    <div>
      <h1>Redux-CRUD</h1>
      <hr/>
      <Routes>
        <Route path="/" exact element={<ProfessorList/>}/>
        <Route path="/professor_add" element={<ProfessorAdd/>}/>
        <Route path="/professor_detail/:profno" element={<ProfessorDetail/>}/>
        <Route path="/professor_edit/:profno" element={<ProfessorEdit/>}/>
      </Routes>
    </div>
  );
}
export default memo(App);
```

```
@fileName : Util.js
@author : 구본아
@Description : Slice파일에서 반복사용되는 코드 분리
```

```
const pending = (state, {payload})=>{
  return { ...state, loading:true }
}
const fulfilled = (state, {payload})=>{
  return {
    loading: false,
    data: payload?.data,
    error: null
  }
}
const rejected=(state, {payload})=>{
  return {
    ...state,
    loading: false,
    error: {
```

```

        code: payload?.data?.rt? payload?.data?.rt : (payload?.status ?
payload.status: 500),
        message: payload?.data?.rtmsg? payload?.data?.rtmsg:
(payload?.statusText? payload.statusText: 'Server Error')
    }
}
}
export {pending, fulfilled, rejected};

```

문제1. 페이지 번호와 키워드 검색 기능이 포함된 목록 조회 기능을 갖는 웹 페이지를 구현하고 이에 대한 소스코드와 실행 스크린샷을 제출하시오

Redux Slice

```

@fileName : ProfessorSlice.js
@author : 구본아
@Description : Professor 테이블 조회 기능, querystring 값으로 검색어, 페이지 번호, 한 페이지
에 보여줄 행 개수를 전달

const API_URL="http://localhost:3001/professor"

export const getList = createAsyncThunk(`${sliceName}/getList/`, async(payload,
{rejectWithValue})=>{
    let result = null;
    try {
        result= await axios.get(API_URL,{
            params:{
                query: payload?.query,
                page:payload?.page,
                rows:payload?.rows
            }
        })
    } catch (error) {
        result = rejectWithValue(error.response);
    }
    return result;
})

const ProfessorSlice = createSlice({
    name: ProfessorSlice,
    initialState: {
        loading: false,
        data: null,
        error: null
    },
    reducers: {

    },
    extraReducers:{
        [getList.pending]: pending,
        [getList.fulfilled]: fulfilled,

```

```

      [getList.rejected]:rejected,
    }
  })

  export default ProfessorSlice.reducer;

```

page

```

@fileName : ProfessorList.js
@author : 구본아
@Description : professor목록 조회페이지
               한페이지에 보여줄 데이터개수 드롭다운으로 선택
               검색기능
               페이지네이션

import React, { memo, useCallback, useRef } from 'react';
import styled from 'styled-components';
import { NavLink, useNavigate } from 'react-router-dom';
import { useQueryString } from '../hooks/useQueryString';

import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import Table from '../components/Table';

import dayjs from 'dayjs';
import { useSelector, useDispatch } from 'react-redux';
import { getList, deleteItem } from '../slices/ProfessorSlice';

const ControlContainer = styled.form`
  position: sticky;
  top: 0;
  background-color: #fff;
  border-top: 1px solid #eee;
  border-bottom: 1px solid #eee;
  padding: 10px 0;

  .controll {
    margin-right: 5px;
    display: inline-block;
    font-size: 16px;
    padding: 7px 10px 5px 10px;
    border: 1px solid #ccc;
  }

  .clickable {
    background-color: #fff;
    color: #000;
    text-decoration: none;
    cursor: pointer;
  }

```

```
      &:hover {
        background-color: #06f2;
      }

      &:active {
        transform: scale(0.9, 0.9);
      }
    }
  `;

const Pagenation = styled.ul`
  list-style: none;
  padding: 0;
  margin: 20px 0;
  display: flex;
  justify-content: center;

  a {
    color: black;
    padding: 8px 12px;
    text-decoration: none;
    transition: background-color .3s;
    margin: 0 5px;

    &.current-page {
      background-color: #116688;
      color: white;
      border-radius: 5px;
    }

    &.disabled {
      color: #ccc;
      cursor: not-allowed;
    }

    &:hover:not(.current-page) {
      background-color: #ddd;
      border-radius: 5px;
    }
  }
`;

const ProfessorList = memo(() => {

  const dispatch = useDispatch();
  const { data, loading, error } = useSelector((state) => state.ProfessorSlice);

  const navigate = useNavigate();

  const { query, rows, page } = useQueryString({
    query: '',
    rows: 10,
    page: 1
  });
});
```

```
const refRowsDropdown = useRef();

const refTextInput = useRef();

React.useEffect(() => {
  dispatch(getList({
    query: query,
    rows: rows,
    page: page
  }));

  refRowsDropdown.current.value = rows;
  refTextInput.current.value = query;
}, [dispatch, query, rows, page]);

const onSearchSubmit = useCallback(e => {
  e.preventDefault();
  const dropdown = refRowsDropdown.current;
  const input = refTextInput.current;
  navigate(`/query=${input.value}&rows=${dropdown.value}`);
}, [navigate]);

const onEditClick = useCallback(e => {
  e.preventDefault();
  const current = e.target;
  const profno = current.dataset.profno;
  navigate(`/professor_edit/${profno}`);
}, [navigate]);

const onDeleteClick = useCallback(e => {
  e.preventDefault();

  const current = e.target;

  if (window.confirm(`정말 ${current.dataset.name}(을)를 삭제하시겠습니까?`))
  {
    dispatch(deleteItem({
      profno: current.dataset.profno
    }));
  }
}, [dispatch]);

const onMoveDetail = useCallback(e=>{
  const profno = e.currentTarget.dataset.profno;
  navigate(`/professor_detail/${profno}`)
},[navigate]);

return (
  <div>
    <Spinner visible={loading} />

    <ControlContainer onSubmit={onSearchSubmit}>
      <select name="rows" className="controll" onChange={onSearchSubmit}>
```

```

ref={refRowsDropdown}>
    <option value="10">10개씩 보기</option>
    <option value="20">20개씩 보기</option>
    <option value="30">30개씩 보기</option>
</select>
<input type="text" className="controll" ref={refTextInput} />
<button type="submit" className="controll clickable">검색</button>
<NavLink to="professor_add" className="controll clickable">교수정
보 추가하기</NavLink>
</ControlContainer>

{error ? (
    <ErrorView error={error} />
) : data && (
    <>
        <Table>
            <thead>
                <tr>
                    <th>교수번호</th>
                    <th>성명</th>
                    <th>아이디</th>
                    <th>직급</th>
                    <th>급여</th>
                    <th>입사일</th>
                    <th>comm</th>
                    <th>학과번호</th>
                    <th>수정</th>
                    <th>삭제</th>
                </tr>
            </thead>
            <tbody>
                {data.item.length > 0 ? (
                    data.item.map((item, index) => {
                        return (
                            <tr key={item.profno}>
                                <td onClick={onMoveDetail} data-
profno={item.profno}>{item.profno}</td>
                                <td>{item.name}</td>
                                <td>{item.userid}</td>
                                <td>{item.position}</td>
                                <td>{item.sal}</td>
                                <td>
{dayjs(item.hiredate).format('YYYY-MM-DD')}</td>
                                <td>{item.comm}</td>
                                <td>{item.deptno}</td>
                                <td>
                                    <button type='button' data-profno=
{item.profno} onClick={onEditClick}>
수정하기
                                    </button>
                                </td>
                                <td>
                                    <button type='button' data-profno=

```

```

{item.profno} data-name={item.name} onClick={onDeleteClick}>
    삭제하기
</button>
</td>
</tr>
);
})
) : (
<tr>
<td colspan='9' align='center'>
    검색결과가 없습니다.
</td>
</tr>
</tbody>
</Table>

{data && (
    <Pagination>
        {data.pagenation.prevGroupLastPage > 0 ? (
            <li><NavLink to={`/?
query=${query}&rows=${rows}&page=${data.pagenation.prevGroupLastPage}`}>&laquo;
</NavLink></li>
        ) : (
            <li><NavLink to="#" className='disabled'>&laquo;
</NavLink></li>
        )}

        {(() => {
            const li = [];
            const start = data.pagenation.groupStart;
            const end = data.pagenation.groupEnd + 1;
            for (let i = start; i < end; i++) {
                if (i === data.pagenation.nowPage) {
                    li.push(<li key={i}><NavLink to="#"
className='current-page'>{i}</NavLink></li>)
                } else {
                    li.push(<li key={i}><NavLink to={`/?
query=${query}&rows=${rows}&page=${i}`}>{i}</NavLink></li>);
                }
            }

            return li;
        })()}

        {data.pagenation.nextGroupFirstPage > 0 ? (
            <li><NavLink to={`/?
query=${query}&rows=${rows}&page=${data.pagenation.nextGroupFirstPage}`}>&raquo;
</NavLink></li>
        ) : (
            <li><NavLink to="#" className='disabled'>&raquo;
</NavLink></li>
        )}
    </Pagination>

```

```

    })
  }
</div>
);
});

export default ProfessorList;

```

Redux-CRUD

10개씩 보기 ▾

검색 교수정보 추가하기

교수번호	성명	아이디	직급	급여	입사일	comm	학과번호	수정	삭제
9926	구아바	rndu456	전임강사	5000	2022-07-04	10	207	수정하기	삭제하기
9925	김춘추	chun789	교수	800	2021-12-31	20	208	수정하기	삭제하기
9924	김춘추	chun852	교수	800	2022-01-30	20	209	수정하기	삭제하기
9923	김춘추	chun852	부교수	800	2022-07-27	20	208	수정하기	삭제하기
9922	김둘리	dul75	전임강사	200	2022-07-13	12	208	수정하기	삭제하기
9921	고길동	gogil85	교수	500	2016-05-05	10	102	수정하기	삭제하기
9920	김라라	lala123	전임강사	500	2015-05-05	20	102	수정하기	삭제하기
9919	홍나나	hong123	전임강사	500	2020-03-05	50	102	수정하기	삭제하기
9918	홍나나	hong123	전임강사	300	2020-03-05	50	208	수정하기	삭제하기
9917	홍나나	hong123	전임강사	300	2020-03-05	50	102	수정하기	삭제하기

« 1 2 »

문제2.단일 항목에 대한 상세 조회 기능을 갖는 웹 페이지를 구현하고 이에 대한 소스코드와 실행 스크린샷을 제출하시오.

page

```

@fileName : ProfessorDetail.js
@author : 구본아
@Description : 교수 목록에서 교수 한명에 대한 상세페이지
               store의 교수 목록데이터에서 클릭된 교수번호에 해당하는 데이터를 활용하여 화면구현

import React, { memo, useEffect, useState } from 'react';
import { useParams } from 'react-router-dom';
import dayjs from 'dayjs';

import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import TableEx from '../components/TableEx';

import { useSelector } from 'react-redux';

const ProfessorDetail = memo(() => {
  const { profno } = useParams();

  const { data, loading, error } = useSelector((state) => state.ProfessorSlice);

```



```

const [ origin, setOrigin] = useState({
  name: '',
  userid: '',
  position: '',
  sal: '',
  hiredate: '',
  comm: '',
  deptno: '',
});

useEffect(() => {
  const index = data.item.findIndex(e => e.profno === parseInt(profno));

  setOrigin({
    name: data.item[index].name,
    userid: data.item[index].userid,
    position: data.item[index].position,
    sal: data.item[index].sal,
    hiredate: data.item[index].hiredate,
    comm: data.item[index].comm,
    deptno: data.item[index].deptno
  });
}, [data,profno]);

return (
  <>
    <Spinner visible={loading} />
    {error ? (
      <ErrorView error={error} />
    ) : (
      <form>
        <TableEx>
          <colgroup>
            <col width="120" />
            <col />
          </colgroup>
          <tbody>
            <tr>
              <th>교수이름</th>
              <td className='inputWrapper'>{origin.name}</td>
            </tr>
            <tr>
              <th>교수아이디</th>
              <td className='inputWrapper'>{origin.userid}</td>
            </tr>
            <tr>
              <th>직급</th>
              <td className="inputWrapper">{origin.position}
            </td>
            </tr>
            <tr>
              <th>급여</th>
              <td className='inputWrapper'>{origin.sal}</td>
            </tr>
          </tbody>
        </TableEx>
      </form>
    )}
  </>
)

```

```

        <tr>
          <th>입사일</th>
          <td className='inputWrapper'>
            {dayjs(origin.hiredate).format('YYYY-MM-DD')}
          </td>
        </tr>
        <tr>
          <th>수수료</th>
          <td className='inputWrapper'>{origin.comm}</td>
        </tr>
        <tr>
          <th>학과번호</th>
          <td className='inputWrapper'>{origin.deptno}</td>
        </tr>
      </tbody>
    </TableEx>
  </form>
  )}
</>
);
});

export default ProfessorDetail;

```

Redux-CRUD

교수이름	김교수
교수아이디	kim789
직급	교수
급여	600
입사일	2022-07-03
수수료	80
학과번호	207

문제3. 신규 데이터를 등록할 수 있는 기능을 갖는 웹 페이지를 구현하고 이에 대한 소스코드와 실행 스크린샷을 제출하시오 관련된 . Redux hook 참조키는 가 아닌 을 사용하여 백엔드로부터

데이터를 받아온 후 Dropdown을 통해 선택할 수 있어야 합니다.

Redux Slice

```
@fileName : ProfessorSlice.js
@author : 구본아
@Description : 새로운 교수데이터 추가를 위한 post요청 함수 정의

const API_URL="http://localhost:3001/professor"

export const postItem = createAsyncThunk(`${sliceName}/postItem/`, async(payload,
{rejectWithValue})=>{
  let result = null;
  try {
    result= await axios.post(API_URL,{
      name: payload.name,
      userid: payload.userid,
      position: payload.position,
      sal: payload.sal,
      hiredate: payload.hiredate,
      comm: payload.comm,
      deptno: payload.deptno
    })
  } catch (error) {
    result = rejectWithValue(error.response);
  }
  return result;
})

const ProfessorSlice = createSlice({
  name: ProfessorSlice,
  initialState: {
    loading: false,
    data: null,
    error: null
  },
  reducers: {

  },
  extraReducers:{
    [postItem.pending]:pending,
    [postItem.fulfilled]: (state, {meta,payload})=>{
      const data = cloneDeep(state.data);

      data.item.unshift(payload.data.item);

      data.item.pop();

      return{
        data: data,
        loading: false,
        error: null
      }
    }
  }
})
```

```

    }
  },
  [postItem.rejected]:rejected,
}
}))

export default ProfessorSlice.reducer;

```

page

```

@fileName : ProfessorAdd.js
@author : 구본아
@Description : 새로운 교수데이터 추가 페이지

import React, { memo, useCallback, useEffect } from 'react';
import {useNavigate} from 'react-router-dom';
import axios from 'axios-hooks';

import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import TableEx from '../components/TableEx';

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

import {useSelector, useDispatch} from 'react-redux';
import {postItem} from '../slices/ProfessorSlice';

const ProfessorAdd = memo(() => {
  const navigate = useNavigate();

  const [{data}] = axios("http://localhost:3001/department/deptnolist")

  const dispatch = useDispatch();
  const {loading, error} = useSelector((state)=>state.ProfessorSlice);

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

    const current = e.target;

    try {
      regexHelper.value(current.name, '교수이름을 입력하세요');
      regexHelper.value(current.userid, '아이디를 입력하세요');
      regexHelper.engNum(current.userid, '아이디는 영문,숫자만 가능합니다.');
      regexHelper.value(current.position, '직급을 선택하세요');
      regexHelper.value(current.hiredate, '입사일을 입력하세요');
      regexHelper.value(current.sal, '급여를 입력하세요');
      regexHelper.num(current.sal, '급여는 숫자입력만 가능합니다.');
      regexHelper.num(current.comm, '수수료는 숫자입력만 가능합니다.');
      regexHelper.value(current.deptno, '학과번호를 입력하세요');
    }
  }

```

```

        regexHelper.num(current.deptno, '학과번호는 숫자입니다.');
```

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

dispatch(postItem({
 name: current.name.value,
 userid: current.userid.value,
 position: current.position.value,
 sal: current.sal.value,
 hiredate: current.hiredate.value,
 comm: current.comm.value,
 deptno: current.deptno.value

})).then(()=>{
 navigate("/");
})
},[dispatch,navigate]);

return (
 <>
 <Spinner visible={loading}/>

 {error? (
 <ErrorView error={error}/>
):(data &&
 <form onSubmit={onSubmit}>
 <TableEx>
 <colgroup>
 <col width="120"/>
 <col/>
 </colgroup>
 <tbody>
 <tr>
 <th>교수이름</th>
 <td className='inputWrapper'><input
className='field' type="text" name="name"/></td>
 </tr>
 <tr>
 <th>교수아이디</th>
 <td className='inputWrapper'><input
className='field' type="text" name="userid"/></td>
 </tr>
 <tr>
 <th>직급</th>
 <td style={{textAlign:'left'}}>
 <input type="radio" id="position"
name="position" value="교수"/>
 <label htmlFor="position">교수</label>
 <input type="radio" id="position"
name="position" value="부교수"/>

```

        <label htmlFor="position">부교수</label>
        <input type="radio" id="position"
name="position" value="조교수"/>
        <label htmlFor="position">조교수</label>
        <input type="radio" id="position"
name="position" value="전임강사"/>
        <label htmlFor="position">전임강사</label>
      </td>
    </tr>
    <tr>
      <th>급여</th>
      <td className='inputWrapper'><input
className='field' type="text" name="sal"/></td>
    </tr>
    <tr>
      <th>입사일</th>
      <td className='inputWrapper'>
        <input className='field' type="date"
name="hiredate"/>
      </td>
    </tr>
    <tr>
      <th>수수료</th>
      <td className='inputWrapper'><input
className='field' type="text" name="comm"/></td>
    </tr>
    <tr>
      <th>학과번호</th>
      <td className='inputWrapper'>
        <select className='field' name="deptno">
          <option>--학과번호--</option>
          {
            data.item.map((v,i)=>{
              return <option key={i}>{v}
</option>
            })
          }
        </select>
      </td>
    </tr>
  </tbody>
</TableEx>

  <div style={{textAlign:'center'}}>
    <button type="submit">저장하기</button>
  </div>
</form>
  )}
</>
);
});
export default ProfessorAdd;

```

Redux-CRUD

교수이름	김교수 추가추가추가
교수아이디	kin2235
직급	<input checked="" type="radio"/> 교수 <input type="radio"/> 부교수 <input type="radio"/> 조교수 <input type="radio"/> 전임강사
급여	800
입사일	2022-07-19 📅
수수료	10
학과번호	--학과번호-- ▼
	<div> --학과번호-- 102 202 207 208 </div>

문제4.데이터를 수정할 수 있는 기능을 갖는 웹 페이지를 구현하고 이에 대한 소스코드와 실행 스크린샷을 제출하시오 관련된 . Redux hook 참조키는 가 아닌 을 사용하여 백엔드로부터 데이터를 받아온 후 Dropdown을 통해 선택할 수 있어야 합니다.

Redux Slice

```
@fileName : ProfessorSlice.js
@author : 구본아
@Description : 교수 데이터 수정을 위한 put요청 함수정의

const API_URL="http://localhost:3001/professor"

export const putItem = createAsyncThunk(`${sliceName}/putItem/`, async(payload,
{rejectWithValue})=>{
  let result = null;
  try {
    result= await axios.put(`${API_URL}/${payload.profno}`, {
      name: payload.name,
      userid: payload.userid,
      position: payload.position,
      sal: payload.sal,
      hiredate: payload.hiredate,
      comm: payload.comm,
      deptno: payload.deptno
    })
  } catch (error) {
    result = rejectWithValue(error.response);
  }
  return result;
});

const ProfessorSlice = createSlice({
  name: ProfessorSlice,
  initialState: {
    loading: false,
    data: null,
```

```

        error: null
      },
      reducers: {

      },
      extraReducers:{
        [putItem.pending]:pending,
        [putItem.fulfilled]: (state,{meta,payload})=>{
          const data = cloneDeep(state.data);

          const index = data.item.findIndex(element=>element.deptno ===
parseInt(meta.arg.profno));

          if(index !== undefined){
            data.item.splice(index,1,payload.data.item);
          }

          return{
            data: data,
            loading: false,
            error: null
          }
        },
        [putItem.rejected]:rejected
      }
    })
  export default ProfessorSlice.reducer;

```

page

```

@fileName : ProfessorEdit.js
@author : 구본아
@Description : 교수 데이터 수정 페이지
              (useAxios_참조키 department테이블의 deptno목록 드롭다운으로 구현)
              (redux_교수 목록데이터에서 클릭된 교수번호의 데이터를 기본값으로 보여준다.)

import React, { memo, useEffect, useState, useCallback } from 'react';
import { useNavigate, useParams } from 'react-router-dom';
import dayjs from 'dayjs';

import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import TableEx from '../components/TableEx';
import regexHelper from '../libs/RegexHelper';

import { useSelector, useDispatch } from 'react-redux';
import useAxios from 'axios-hooks';
import { putItem } from '../slices/ProfessorSlice';

const ProfessorEdit = memo(() => {
  const {profno} = useParams();

```



```
const navigate = useNavigate();
const [checked, setChecked] = useState("");

const [{data:deptnoData,loading:deptnoLoading,error:deptnoError},refetch] =
useAxios("http://localhost:3001/department/deptnolist")

const dispatch = useDispatch();
const { data, loading, error } = useSelector((state) => state.ProfessorSlice);
const [ origin, setOrigin] = useState({
  name: '',
  userid: '',
  position: '',
  sal: '',
  hiredate: '',
  comm: '',
  deptno: '',
});

useEffect(() => {
  const index = data.item.findIndex(e => e.profno === parseInt(profno));

  setOrigin({
    name: data.item[index].name,
    userid: data.item[index].userid,
    position: data.item[index].position,
    sal: data.item[index].sal,
    hiredate: data.item[index].hiredate,
    comm: data.item[index].comm,
    deptno: data.item[index].deptno
  });

  setChecked(data.item[index].position);

}, [data,profno]);

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

  try {
    regexHelper.value(current.name, '교수이름을 입력하세요');
    regexHelper.value(current.userid, '아이디를 입력하세요');
    regexHelper.engNum(current.userid, '아이디는 영문,숫자만 가능합니다.')
    regexHelper.value(current.position, '직급을 선택하세요');
    regexHelper.value(current.hiredate, '입사일을 입력하세요');
    regexHelper.value(current.sal, '급여를 입력하세요');
    regexHelper.num(current.sal, '급여는 숫자입력만 가능합니다.')
    regexHelper.num(current.comm, '수수료는 숫자입력만 가능합니다.')
    regexHelper.value(current.deptno, '학과번호를 입력하세요');
    regexHelper.num(current.deptno, '학과번호는 숫자입니다.');

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

```

        return;
    }

    dispatch(putItem({
        profno: profno,
        name: current.name.value,
        userid: current.userid.value,
        position: current.position.value,
        sal: current.sal.value,
        hiredate: current.hiredate.value,
        comm: current.comm.value,
        deptno: current.deptno.value
    })).then(() => {
        navigate("/");
    });

}, [dispatch, profno, navigate]);

return (
    <>
        <Spinner visible={deptnoLoading} />
        {error && deptnoError ? (
            <ErrorView error={error} />
        ) : (
            (checked && origin && deptnoData) &&(

                <form onSubmit={onSubmit}>
                    <TableEx>
                        <colgroup>
                            <col width="120" />
                            <col />
                        </colgroup>
                        <tbody>
                            <tr>
                                <th>교수이름</th>
                                <td className='inputWrapper'><input
className='field' type="text" name="name" defaultValue={origin.name}/></td>
                            </tr>
                            <tr>
                                <th>교수아이디</th>
                                <td className='inputWrapper'><input
className='field' type="text" name="userid" defaultValue={origin.userid}/></td>
                            </tr>
                            <tr>
                                <th>직급</th>
                                <td style={{textAlign: 'left'}}>
                                    <input type="radio" id="position"
name="position" value="교수" defaultChecked={checked === "교수"}/>
                                    <label htmlFor="position">교수</label>
                                    <input type="radio" id="position"
name="position" value="부교수" defaultChecked={checked === "부교수"}/>
                                    <label htmlFor="position">부교수</label>
                                    <input type="radio" id="position"
name="position" value="조교수" defaultChecked={checked === "조교수"}/>

```

```

        <label htmlFor="position">조교수</label>
        <input type="radio" id="position"
name="position" value="전임강사" defaultChecked={checked === "전임강사"}/>
        <label htmlFor="position">전임강사</label>

        </td>
      </tr>
      <tr>
        <th>급여</th>
        <td className='inputWrapper'><input
className='field' type="text" name="sal" defaultValue={origin.sal}/></td>
      </tr>
      <tr>
        <th>입사일</th>
        <td className='inputWrapper'>
          <input className='field' type="date"
name="hiredate" defaultValue={dayjs(origin.hiredate).format('YYYY-MM-DD')}/>
        </td>
      </tr>
      <tr>
        <th>수수료</th>
        <td className='inputWrapper'><input
className='field' type="text" name="comm" defaultValue={origin.comm}/></td>
      </tr>
      <tr>
        <th>학과번호</th>
        <td className='inputWrapper'>
          <select className='field' name="deptno">
            {
              deptnoData.item.map((v,i)=>{
                return <option key={i}>{v}
              })
            }
          </select>
        </td>
      </tr>
    </tbody>
  </TableEx>

  <div style={{ textAlign: 'center' }}>
    <button type='submit'>저장하기</button>
  </div>
</form>
)
  })
</>
);
});
export default ProfessorEdit;

```

10개씩 보기 ▾

검색

교수정보 추가하기

교수번호	성명	아이디	직급	급여	입사일	comm	학과번호	수정	삭제
9927	김교수	kim789	교수	600	2022-07-03	80	207	수정하기	삭제하기
9926	구아바	mdu456	전임강사	5000	2022-07-04	10	207	수정하기	삭제하기
9925	김춘추	chun789	교수	800	2021-12-31	20	208	수정하기	삭제하기
9924	김춘추	chun852	교수	800	2022-01-30	20	209	수정하기	삭제하기
9923	김춘추	chun852	부교수	800	2022-07-27	20	208	수정하기	삭제하기
9922	김둘리	dul75	전임강사	200	2022-07-13	12	208	수정하기	삭제하기
9921	고길동	gogil85	교수	500	2016-05-05	10	102	수정하기	삭제하기
9920	김라라	lala123	전임강사	500	2015-05-05	20	102	수정하기	삭제하기
9919	홍나나	hong123	전임강사	500	2020-03-05	50	102	수정하기	삭제하기

«

1

2

»

Redux-CRUD

교수이름	김춘추		
교수아이디	chun789		
직급	<input checked="" type="radio"/> 교수 <input type="radio"/> 부교수 <input type="radio"/> 조교수 <input type="radio"/> 전임강사		
급여	800		
입사일	2021-12-31		📅
수수료	20		
학과번호	102 ▾		

저장하기

문제5. 목록 페이지의 삭제버튼을 통해 데이터를 삭제할 수 있는 기능을 갖는 웹 페이지를 구현 하고 이에 대한 소스코드와 실행 스크린샷을 제출하시오

Redux Slice

```

@fileName : ProfessorSlice.js
@author : 구본아
@Description : 교수데이터 삭제를 위한 delete요청함수 정의
              클릭된 교수번호(profno)값을 path parameter로 전송

const API_URL="http://localhost:3001/professor"

export const deleteItem = createAsyncThunk(`${sliceName}/deleteItem/`,
  async(payload, {rejectWithValue})=>{
    let result = null;
    try {
      result= await axios.delete(`${API_URL}/${payload.profno}`)
    } catch (error) {
      result = rejectWithValue(error.response);
    }
    return result;
  })

const ProfessorSlice = createSlice({
  name: ProfessorSlice,
  initialState: {

```

```

        loading: false,
        data: null,
        error: null
      },
      reducers: {

      },
      extraReducers: {
        [deleteItem.pending]: pending,
        [deleteItem.fulfilled]: (state, {meta, payload}) => {
          const data = cloneDeep(state.data);

          const index = data.item.findIndex(element => element.deptno ===
parseInt(meta.arg.profno));
          console.log('index = ' + index);

          if(index !== undefined){
            data.item.splice(index, 1);
          }

          return {
            data: data,
            loading: false,
            error: null
          }
        },
        [deleteItem.rejected]: rejected
      }
    })
  export default ProfessorSlice.reducer;

```

page

```

@fileName : ProfessorList.js
@author : 구분아
@Description : 교수 목록 조회 페이지에서 삭제기능관련 코드(삭제기능 클릭이벤트, 삭
제버튼)

const onDeleteClick = useCallback(e => {
  e.preventDefault();

  const current = e.target;

  if (window.confirm(`정말 ${current.dataset.name}(을)를 삭제하시겠습니까?`))
  {
    dispatch(deleteItem({
      profno: current.dataset.profno
    }));
  }
}, [dispatch]);

```

```

<Table>
  <thead>
    <tr>
      <th>교수번호</th>
      <th>성명</th>
      <th>아이디</th>
      <th>직급</th>
      <th>급여</th>
      <th>입사일</th>
      <th>comm</th>
      <th>학과번호</th>
      <th>수정</th>
      <th>삭제</th>
    </tr>
  </thead>
  <tbody>
    {data.item.length > 0 ? (
      data.item.map((item, index) => {
        return (
          <tr key={item.profno}>
            <td onClick={onMoveDetail} data-profno=
{item.profno}>{item.profno}</td>
            <td>{item.name}</td>
            <td>{item.userid}</td>
            <td>{item.position}</td>
            <td>{item.sal}</td>
            <td>{dayjs(item.hiredate).format('YYYY-MM-DD')}
</td>
            <td>{item.comm}</td>
            <td>{item.deptno}</td>
            <td>
              <button type='button' data-profno=
{item.profno} onClick={onEditClick}>
                수정하기
              </button>
            </td>
            <td>
              <button type='button' data-profno=
{item.profno} data-name={item.name} onClick={onDeleteClick}>
                삭제하기
              </button>
            </td>
          </tr>
        );
      })
    )}
  </tbody>
</Table>

```

기술면접수업github강사githubnotiongithubleekh4232/n

localhost:3000 내용:
정말 김교수(을)를 삭제하시겠습니까?

확인취소

디자인유용한들NFT

10개씩 보기

검색

교수정보 추가하기

교수번호	성명	아이디	직급	급여	입사일	comm	학과번호	수정	삭제
9927	김교수	kim789	교수	600	2022-07-03	80	207	수정하기	삭제하기
9926	구아바	rndu456	전임강사	5000	2022-07-04	10	207	수정하기	삭제하기
9925	김춘추	chun789	교수	800	2021-12-31	20	208	수정하기	삭제하기
9924	김춘추	chun852	교수	800	2022-01-30	20	209	수정하기	삭제하기
9923	김춘추	chun852	부교수	800	2022-07-27	20	208	수정하기	삭제하기
9922	김둘리	dul75	전임강사	200	2022-07-13	12	208	수정하기	삭제하기
9921	고길동	gogil85	교수	500	2016-05-05	10	102	수정하기	삭제하기
9920	김라라	lala123	전임강사	500	2015-05-05	20	102	수정하기	삭제하기
9919	홍나나	hong123	전임강사	500	2020-03-05	50	102	수정하기	삭제하기
9918	홍나나	hong123	전임강사	300	2020-03-05	50	208	수정하기	삭제하기

«12»

10개씩 보기

검색

교수정보 추가하기

교수번호	성명	아이디	직급	급여	입사일	comm	학과번호	수정	삭제
9927	김교수	kim789	교수	600	2022-07-03	80	207	수정하기	삭제하기
9926	구아바	rndu456	전임강사	5000	2022-07-04	10	207	수정하기	삭제하기
9925	김춘추	chun789	교수	800	2021-12-31	20	208	수정하기	삭제하기
9924	김춘추	chun852	교수	800	2022-01-30	20	209	수정하기	삭제하기
9923	김춘추	chun852	부교수	800	2022-07-27	20	208	수정하기	삭제하기
9922	김둘리	dul75	전임강사	200	2022-07-13	12	208	수정하기	삭제하기
9921	고길동	gogil85	교수	500	2016-05-05	10	102	수정하기	삭제하기
9920	김라라	lala123	전임강사	500	2015-05-05	20	102	수정하기	삭제하기
9919	홍나나	hong123	전임강사	500	2020-03-05	50	102	수정하기	삭제하기

«12»