권채림 react crud 연습문제

> 2022-11-17

문제3 Traffic_Acc

App.js

```
import React, {memo} from 'react';
import {Routes, Route } from "react-router-dom";
import TrafficList from './pages/TrafficList';
import TrafficAdd from './pages/TrafficAdd';
import TrafficEdit from './pages/TrafficEdit';
import TrafficView from './pages/TrafficView';

const App = memo(() => {
 return (
   <div>
     <h1>Traffic_Acc</h1>

     <Routes>
       <Route path='/' exapt={true} element={<TrafficList/>}/>
       <Route path='/traffic_add' element={<TrafficAdd/>}/>
       <Route path='/traffic_view/:id' element={<TrafficView/>}/>
       <Route path='/traffic_edit/:id' element={<TrafficEdit/>}/>
     </Routes>
   </div>
 );
})

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
// /*
import App from './App';
/*/
import App from './test';
/**/
import { BrowserRouter } from 'react-router-dom';
import { Provider } from 'react-redux';
import store from './store';

const root = ReactDOM.createRoot(document.getElementById('root'));
```

```
root.render(
 <Provider store={store}>
   <BrowserRouter>
     <App />
   </BrowserRouter>
 </Provider>
);
```

store.js

```
import { configureStore } from "@reduxjs/toolkit";
import TrafficSlice from "./slices/TrafficSlice";
const store = configureStore({
   reducer: {
       TrafficSlice: TrafficSlice,
   }
});

export default store;
```

TrafficSlice.js

```
import { createSlice, createAsyncThunk } from '@reduxjs/toolkit'
import axios from 'axios';
import { pending, fulfilled, rejected } from '../helper/ReduxHelper';
import { cloneDeep } from 'lodash';


/** 다중행 데이터 조회를 위한 비동기 함수 */
export const getList = createAsyncThunk("TrafficSlice/getList", async (payload,
{rejectWithValue }) => {

   let result = null;
   const URL = process.env.REACT_APP_API_TRAFFICACC_LIST;

   try {
       const response = await axios.get(URL);
       result = response.data;
   } catch (err) {
       result = rejectWithValue(err.response);
   }
   return result;
});

/** 단일행 데이터 조회를 위한 비동기 함수 */
```

```
export const getItem = createAsyncThunk("TrafficSlice/getItem", async (payload,
{rejectWithValue }) => {

   let result = null;
   const URL = process.env.REACT_APP_API_TRAFFICACC_ITEM.replace(':id',
payload.id);
   console.log(URL);
   try {
       const response = await axios.get(URL);
       result = response.data;
   } catch (err) {
       result = rejectWithValue(err.response);
   }
   return result;
});

/** 데이터 저장을 위한 비동기 함수 */
export const postItem = createAsyncThunk("TrafficSlice/postItem", async (payload,
{rejectWithValue }) => {

   let result = null;
   const URL = process.env.REACT_APP_API_TRAFFICACC_LIST;

   try {
       const response = await axios.post(URL, {
           year: payload.year,
           month: payload.month,
           accident: payload.accident,
           death: payload.death,
           injury: payload.injury,
       });
       result = response.data;
   } catch (err) {
       result = rejectWithValue(err.response);
   }
   return result;
});

/** 데이터 수정을 위한 비동기 함수 */
export const putItem = createAsyncThunk("TrafficSlice/putItem", async (payload,
{rejectWithValue }) => {

   let result = null;
   const URL = process.env.REACT_APP_API_TRAFFICACC_ITEM.replace(':id',
payload.id);

   try {
       const response = await axios.put(URL, {
           year: payload.year,
           month: payload.month,
           accident: payload.accident,
           death: payload.death,
           injury: payload.injury,
       });
```

```
        result = response.data;
    } catch (err) {
        result = rejectWithValue(err.response);
    }
    return result;
});

/** 데이터 삭제을 위한 비동기 함수 */
export const deleteItem = createAsyncThunk("TrafficSlice/deleteItem", async
(payload, {rejectWithValue }) => {

    let result = null;
    const URL = process.env.REACT_APP_API_TRAFFICACC_ITEM.replace(':id',
payload.id);

    try {
        const response = await axios.delete(URL);
        result = response.data;
    } catch (err) {
        result = rejectWithValue(err.response);
    }
    return result;
});




const TrafficSlice = createSlice({
    name: 'TrafficSlice',

    initialState: {
        data: null,
        loading: false,
        error: null
    },
    reducers: {
        getCurrentData: (state, action) => {
            return state;
        }
    },
    extraReducers: {
        [getList.pending]: pending,
        [getList.fulfilled]: fulfilled,
        [getList.rejected]: rejected,

        [getItem.pending]: pending,
        [getItem.fulfilled]: (state, {meta, payload}) => {
            return {
                data: [payload],
                loading: false,
                error: null
            }
        },
        [getItem.rejected]: rejected,
        [postItem.pending]: pending,
        [postItem.fulfilled]: (state, {meta, payload}) => {
            const data = cloneDeep(state.data);
            console.log("data", data);

            data.push(payload);

            return {
                data:data,
                loading: false,
                error: null
            }
        },
        [postItem.rejected]: rejected,

        [deleteItem.pending]: pending,
        [deleteItem.fulfilled]: (state, {meta, payload}) => {
            console.log("meta",meta)
            const data = cloneDeep(state.data);
            const targetId = data.findIndex((v, i) => v.id ===
Number(meta.arg.id));
            console.log("targetId", targetId);

            data.splice(targetId, 1);

            return {
                data: data,
                loading: false,
                error: null
            }
        },
        [deleteItem.rejected]: rejected,

        [putItem.pending]: pending,
        [putItem.fulfilled]: (state, {meta, payload}) => {
            const data = cloneDeep(state.data);
            const targetId = data.findIndex((v, i) => v.id ===
Number(meta.arg.id));
            console.log("targetId", targetId);

            data.splice(targetId, 1, payload);

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

export const {getCurrentData} = TrafficSlice.actions;
export default TrafficSlice.reducer;
```

TrafficList.js

```
import React, {memo, useCallback} from 'react';
import { useDispatch, useSelector } from 'react-redux';
import { NavLink, useNavigate } from 'react-router-dom';
import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import Table from '../components/Table';
import { deleteItem, getList } from '../slices/TrafficSlice';
import styled from 'styled-components';
import dayjs from 'dayjs';


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: #06fe06;
       }
       &:active {
           transform: scale(0.9, 0.9);
       }
   }
`;


const TrafficList = memo(() => {
   const dispatch = useDispatch();
   const {data, loading, error} = useSelector((state) => state.TrafficSlice);
```

```
    React.useEffect(()=>{
        dispatch(getList())
    },[]);

    const navigate = useNavigate();

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

        const current = e.currentTarget;
        const { id } = current.dataset;

        if(window.confirm(`정말 ${id}번(을)를 삭제하시겠습니까?`)) {
            dispatch(deleteItem({
                id: id
            }))
        }
    }, []);

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

        const current = e.currentTarget;
        const {id} = current.dataset;

        navigate(`/traffic_edit/${id}`);
    })


    return(
        <div>
            {/* 로딩바 */}
            <Spinner loading={loading}/>
            {/* 추가탭 */}
            <ControlContainer>
                <NavLink to="traffic_add" className="controll clickable">교통사고정
보 추가하기</NavLink>
            </ControlContainer>

            {/* 조회결과 */}
            {error ? (
                <ErrorView error={error}/>
            ) : (
                data && (
                    <Table>
                        <thead>
                            <tr>
                                <th>No.</th>
                                <th>년도</th>
                                <th>월</th>
                                <th>교통사고 건수</th>
                                <th>사망자 수</th>
                                <th>부상자 수</th>
                                <th>수정</th>
```

```
                                        <th>삭제</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    {
                                        data.length > 0 ? (
                                            data.map((v, i) => {
                                                return (
                                                    <tr key={v.id}>
                                                        <td>
                                                            <NavLink to=
{`/traffic_view/${v.id}`}>{v.id}</NavLink>
                                                        </td>
                                                        <td>{v.year}</td>
                                                        <td>{v.month}</td>
                                                        <td>{v.accident}</td>
                                                        <td>{v.death}</td>
                                                        <td>{v.injury}</td>
                                                        <td>
                                                            <button type='button' data-id =
{v.id} data-name={v.name} onClick={onTrafficEditClick} >
                                                                수정하기
                                                            </button>
                                                        </td>
                                                        <td>
                                                            <button type='button' data-id =
{v.id} data-name={v.name} onClick={onTrafficItemDelete}>
                                                                삭제하기
                                                            </button>
                                                        </td>
                                                    </tr>
                                                )
                                            })
                                        ):(
                                            <tr>
                                                <td colSpan='8' align='center'>검색결과가 없
습니다.</td>
                                            </tr>
                                        )
                                    }
                                </tbody>
                            </Table>
                        )
                    )}
            </div>
    )
})

export default TrafficList;
```

TrafficAdd.js

```
import React, {memo, useCallback} from 'react';
import TableEx from '../components/TableEx';
import { useNavigate } from 'react-router-dom';
import { useSelector, useDispatch } from 'react-redux';
import { postItem } from '../slices/TrafficSlice';
import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';


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

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

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

       const current = e.currentTarget;

       dispatch(postItem({
           year: current.year.value,
           month: current.month.value,
           accident: current.accident.value,
           death: current.death.value,
           injury: current.injury.value,
       })).then((result) => {
           console.log("result:",result);
           navigate(`/traffic_view/${result.payload.id}`);
       })
   }, [])

   return(
       <div>
           <Spinner loading={loading}/>
           {error ? (
               <ErrorView error={error}/>
           ) : (
           <form onSubmit={onTrafficSubmit}>
               <TableEx>
                   <colgroup>
                       <col width="120"/>
                       <col/>
                   </colgroup>
                   <tbody>
                       <tr>
                           <th>년도</th>
                           <td className='inputWrapper'>
                               <input className="field" type="text" name="year"/>
                           </td>
                       </tr>
                       <tr>
```

```
                        <th>월</th>
                            <td className='inputWrapper'>
                            <input className="field" type="text" name="month"/>
                        </td>
                    </tr>
                    <tr>
                        <th>교통사고 건수</th>
                            <td className='inputWrapper'>
                            <input className="field" type="text"
name="accident"/>
                        </td>
                    </tr>
                    <tr>
                        <th>사망자 수</th>
                            <td className='inputWrapper'>
                            <input className="field" type="text" name="death"/>
                        </td>
                    </tr>
                    <tr>
                        <th>부상자 수</th>
                            <td className='inputWrapper'>
                            <input className="field" type="text"
name="injury"/>
                        </td>
                    </tr>

                </tbody>
            </TableEx>

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

        )}
    </div>
  )
})

export default TrafficAdd;
```

TrafficEdit.js

```
import React, {memo, useCallback, useEffect, useMemo} from 'react';
import TableEx from '../components/TableEx';
import { useNavigate, useParams } from 'react-router-dom';
import { useSelector, useDispatch } from 'react-redux';
import { getCurrentData, getItem, putItem } from '../slices/TrafficSlice';
import Spinner from '../components/Spinner';
import ErrorView from '../components/ErrorView';
import dayjs from 'dayjs';

const TrafficEdit = memo(() => {
   const { id } = useParams();

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

   useEffect(()=> {
       dispatch(getCurrentData());
   },[]);

   const item = useMemo(() => {
       if(data) {
           return data.find((v, i) => v.id == id);
       } else {
           dispatch(getItem({id:id}));
       }
   }, [data])

   const navigate = useNavigate();

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

       const current = e.currentTarget;

       dispatch(putItem({
           id: current.id.value,
           year: current.year.value,
           month: current.month.value,
           accident: current.accident.value,
           death: current.death.value,
           injury: current.injury.value,
       })).then((result) => {
           console.log("result:",result);
           navigate(`/traffic_view/${result.payload.id}`);
       })
   },[])

   return(
       <div>
           <Spinner loading={loading}/>
           {error ? (
               <ErrorView error={error}/>
           ) : (
           <form onSubmit={onTrafficSubmit}>
               <input type="hidden" name="id" defaultValue={item?.id} />
               <TableEx>
                   <colgroup>
                       <col width="120"/>
                       <col/>
                   </colgroup>
                   <tbody>
```

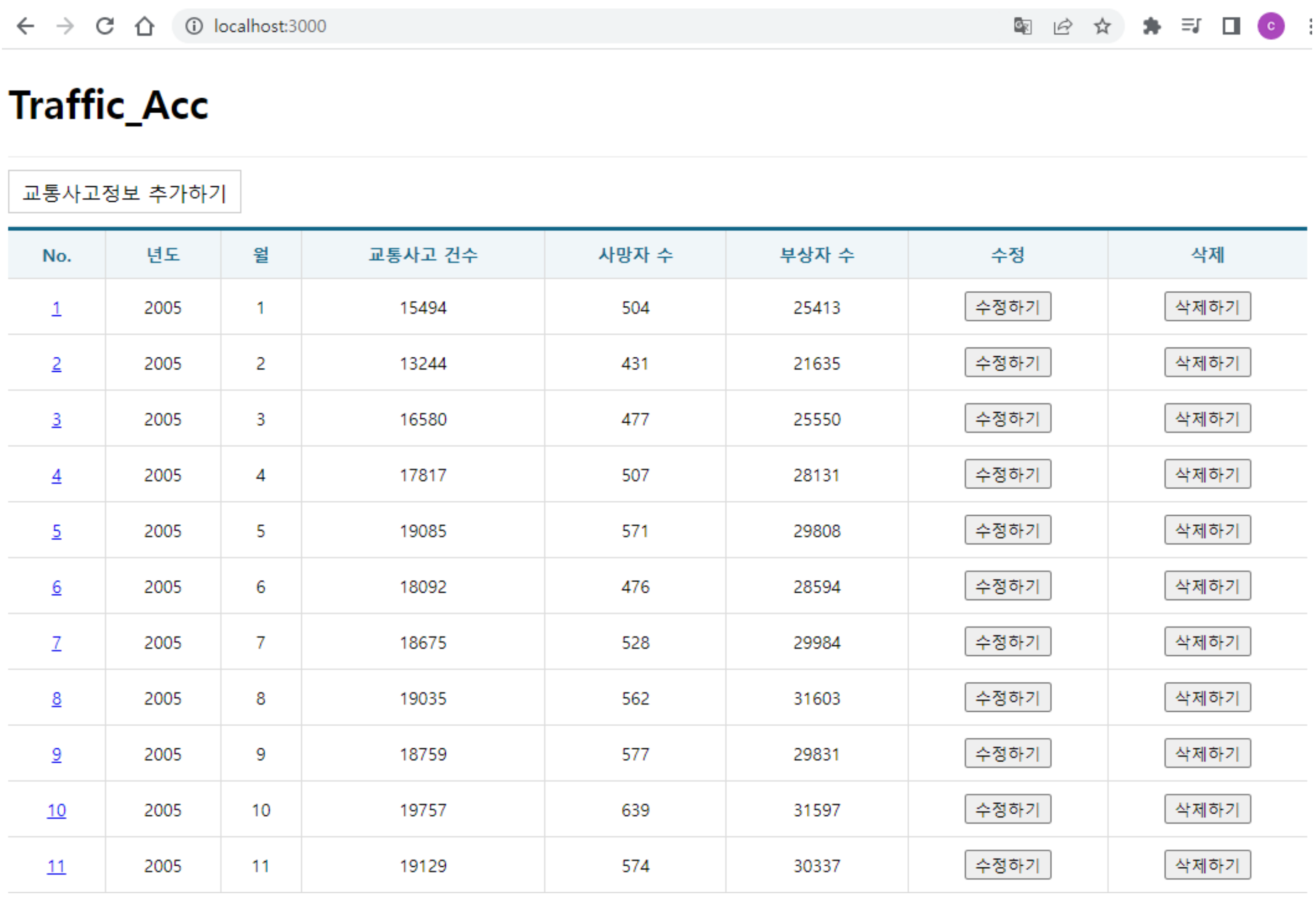
```
                        <tr>
                            <th>년도</th>
                            <td className='inputWrapper'>
                                <input className="field" type="text" name="year"
defaultValue={item?.year}/>
                            </td>
                        </tr>
                        <tr>
                            <th>월</th>
                                <td className='inputWrapper'>
                                <input className="field" type="text" name="month"
defaultValue={item?.month}/>
                            </td>
                        </tr>
                        <tr>
                            <th>교통사고 건수</th>
                                <td className='inputWrapper'>
                                <input className="field" type="text"
name="accident" defaultValue={item?.accident}/>
                            </td>
                        </tr>
                        <tr>
                            <th>사망자 수</th>
                                <td className='inputWrapper'>
                                <input className="field" type="text" name="death"
defaultValue={item?.death}/>
                            </td>
                        </tr>
                        <tr>
                            <th>부상자 수</th>
                                <td className='inputWrapper'>
                                <input className="field" type="text" name="injury"
defaultValue={item?.injury}/>
                            </td>
                        </tr>
                    </tbody>
                </TableEx>

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

            )}
        </div>
    )
})

export default TrafficEdit;
```

TrafficView.js

```
import React, {memo, useCallback, useEffect, useMemo} from 'react';
import { NavLink, useParams, useNavigate } from 'react-router-dom';
import { useSelector, useDispatch } from 'react-redux';
import { getCurrentData, deleteItem, getItem } from '../slices/TrafficSlice';

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

const TrafficView = memo(() => {
   const {id} = useParams();

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

   useEffect(()=>{
       dispatch(getCurrentData());
   }, [])


   const item = useMemo(() => {
       if(data) {
           return data.find((v,i)=> v.id == id);
       } else {
           dispatch(getItem({id: id}));
       }
   }, [data]);

   const navigate = useNavigate();

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

       const current = e.currentTarget;
       const { id } = current.dataset;

       if(window.confirm(`정말 ${id}번(을)를 삭제하시겠습니까?`)) {
           dispatch(deleteItem({
               id: id
           })).then(({meta, payload}) => {
               navigate('/');
           })
       }
   }, []);


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

```
            {error ? (
                <ErrorView error={error}/>
            ) : (
                item && (
                    <div>
                        <Table>
                            <colgroup>
                                <col width="120"/>
                                <col/>
                            </colgroup>
                            <tbody>
                                <tr>
                                    <th>No.</th>
                                    <td>{item.id}</td>
                                </tr>
                                <tr>
                                    <th>년도</th>
                                    <td>{item.year}</td>
                                </tr>
                                <tr>
                                    <th>월</th>
                                    <td>{item.month}</td>
                                </tr>
                                <tr>
                                    <th>교통사고 건수</th>
                                    <td>{item.accident}</td>
                                </tr>
                                <tr>
                                    <th>사망자 수</th>
                                    <td>{item.death}</td>
                                </tr>
                                <tr>
                                    <th>부상자 수</th>
                                    <td>{item.injury}</td>
                                </tr>

                            </tbody>
                        </Table>

                        <div style={{textAlign: 'center'}}>
                            <NavLink to="/">목록</NavLink>
                            &nbsp;|&nbsp;
                            <NavLink to="/traffic_add">등록</NavLink>
                            &nbsp;|&nbsp;
                            <NavLink to={`/traffic_edit/${item.id}`}>수정</NavLink>
                            &nbsp;|&nbsp;
                            <NavLink to="#!" data-id = {item.id} data-name=
{item.name} onClick={onTrafficItemDelete}>삭제</NavLink>

                        </div>
                    </div>
                )
            )}
        </div>
```

```
    )
})

export default TrafficView;
```

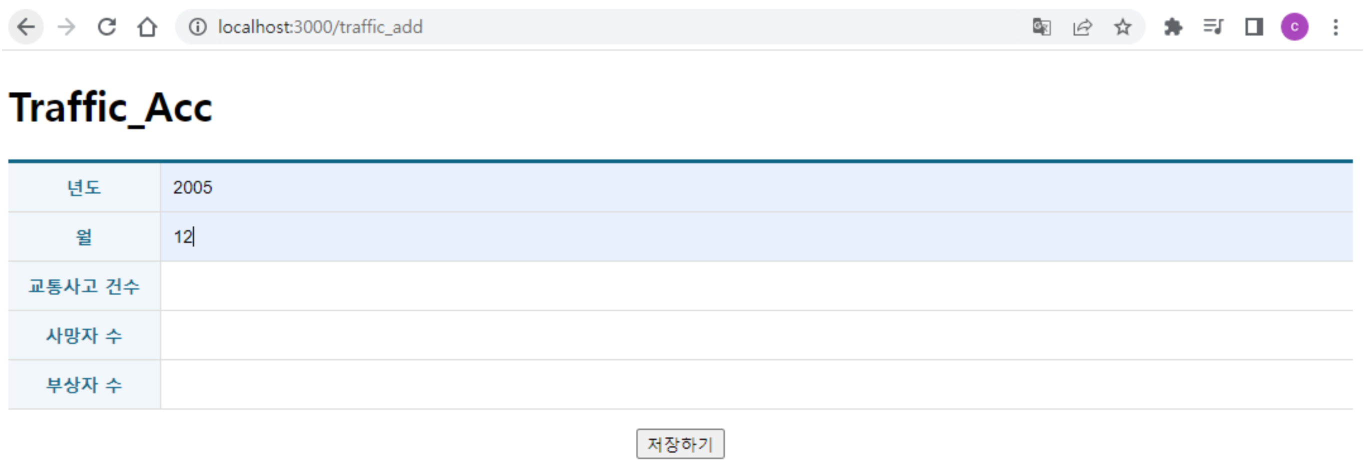


localhost:3000

Traffic_Acc

교통사고정보 추가하기

No.	년도	월	교통사고 건수	사망자 수	부상자 수	수정	삭제
1	2005	1	15494	504	25413	수정하기	삭제하기
2	2005	2	13244	431	21635	수정하기	삭제하기
3	2005	3	16580	477	25550	수정하기	삭제하기
4	2005	4	17817	507	28131	수정하기	삭제하기
5	2005	5	19085	571	29808	수정하기	삭제하기
6	2005	6	18092	476	28594	수정하기	삭제하기
7	2005	7	18675	528	29984	수정하기	삭제하기
8	2005	8	19035	562	31603	수정하기	삭제하기
9	2005	9	18759	577	29831	수정하기	삭제하기
10	2005	10	19757	639	31597	수정하기	삭제하기
11	2005	11	19129	574	30337	수정하기	삭제하기

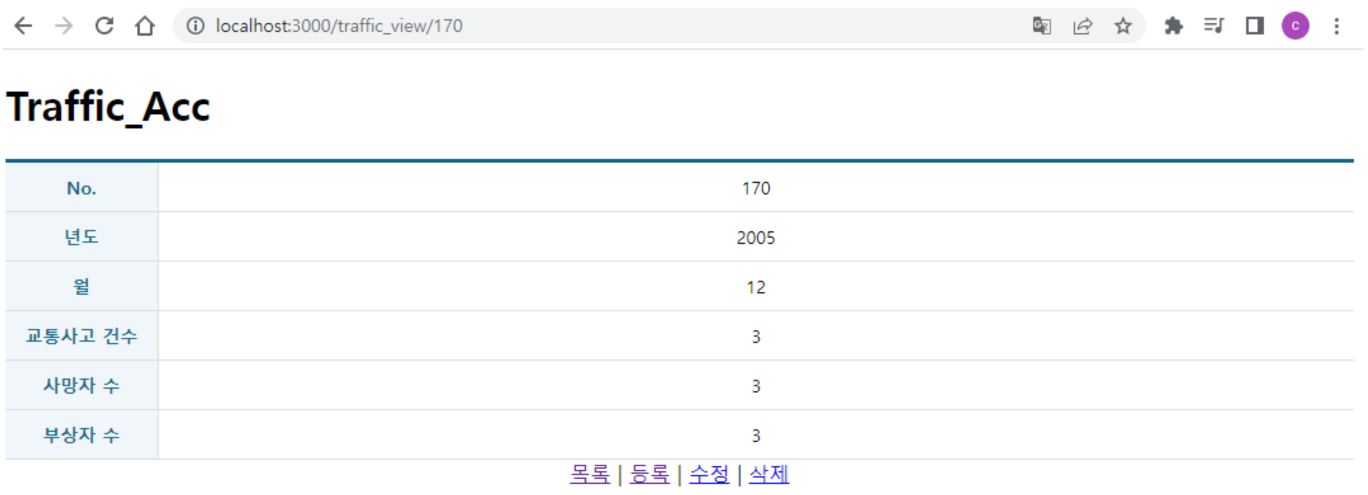


localhost:3000/traffic_add

Traffic_Acc

년도	2005
월	12
교통사고 건수	
사망자 수	
부상자 수	

저장하기

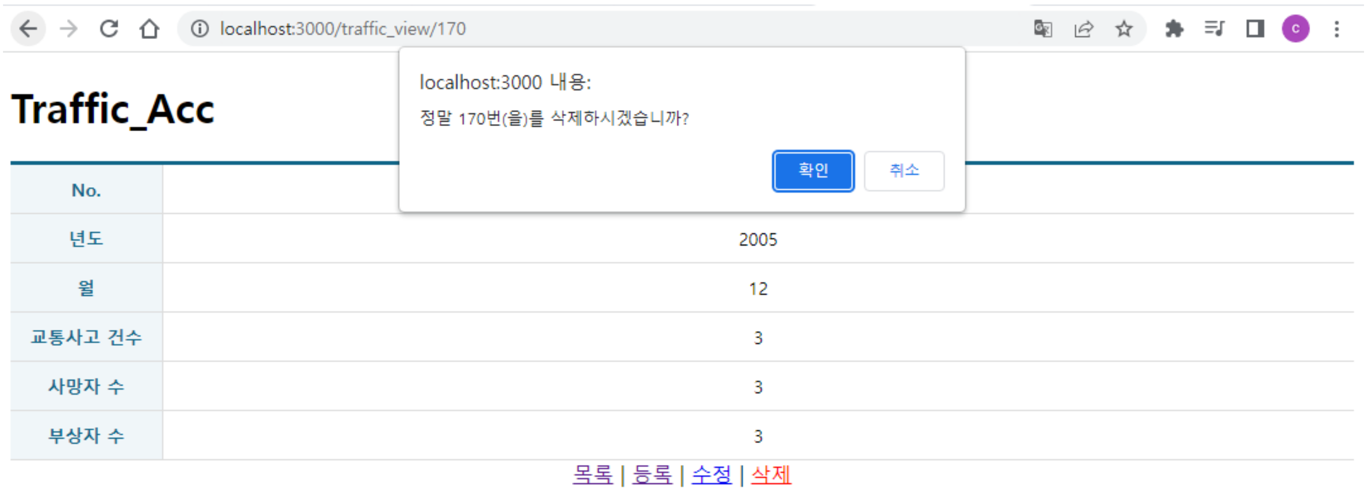


localhost:3000/traffic_view/170

Traffic_Acc

No.	170
년도	2005
월	12
교통사고 건수	3
사망자 수	3
부상자 수	3

목록 | 등록 | 수정 | 삭제



localhost:3000/traffic_view/170

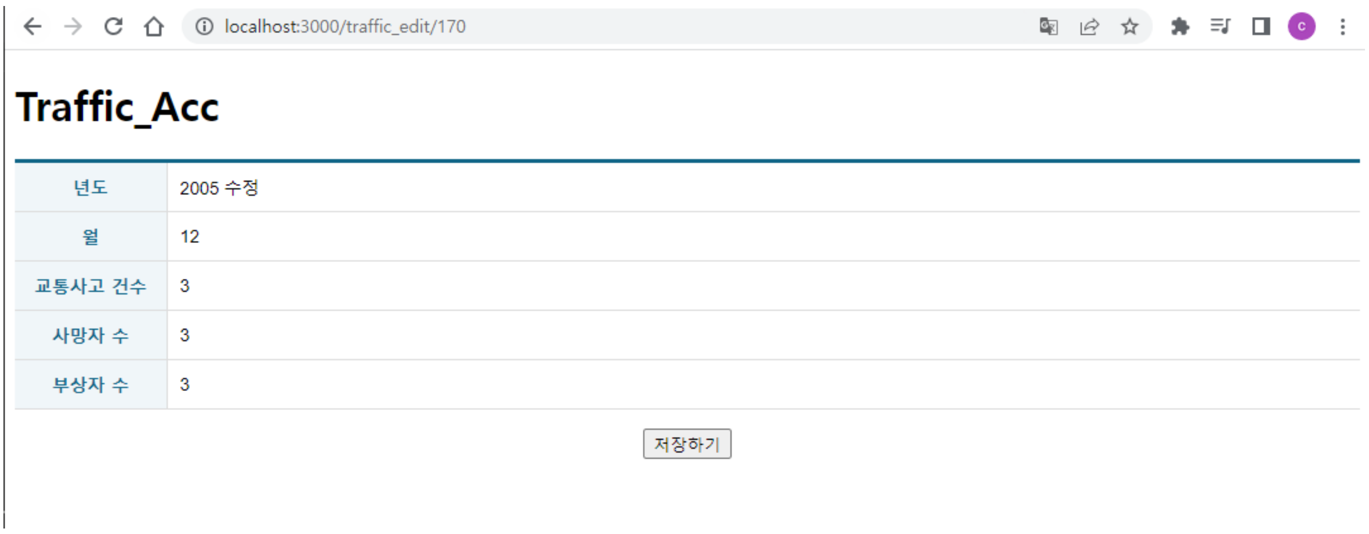
localhost:3000 내용:

정말 170번(을)를 삭제하시겠습니까?

Traffic_Acc

No.	
년도	2005
월	12
교통사고 건수	3
사망자 수	3
부상자 수	3

목록 | 등록 | 수정 | 삭제



localhost:3000/traffic_edit/170

Traffic_Acc

년도	2005 수정
월	12
교통사고 건수	3
사망자 수	3
부상자 수	3