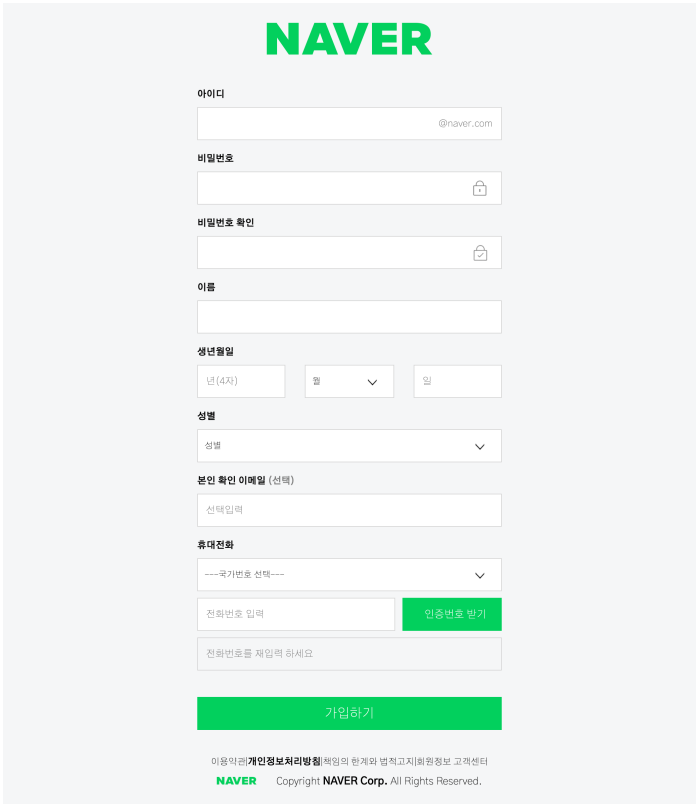
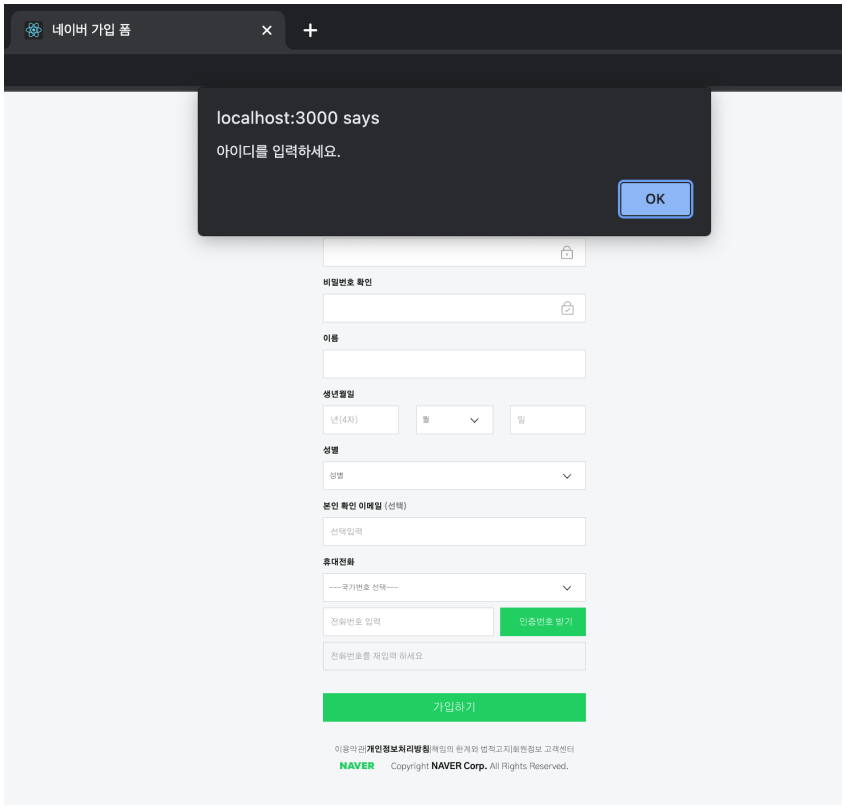
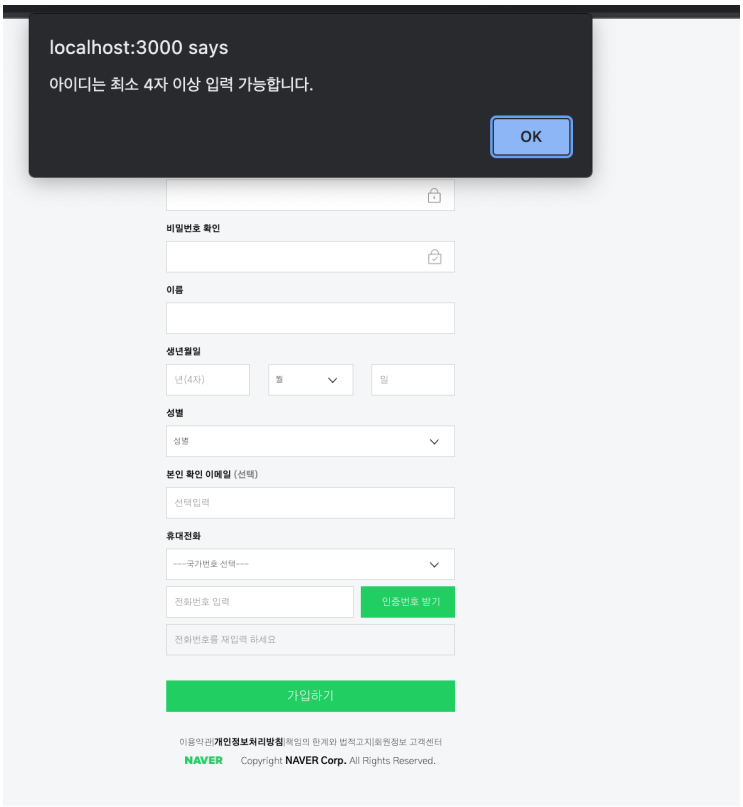
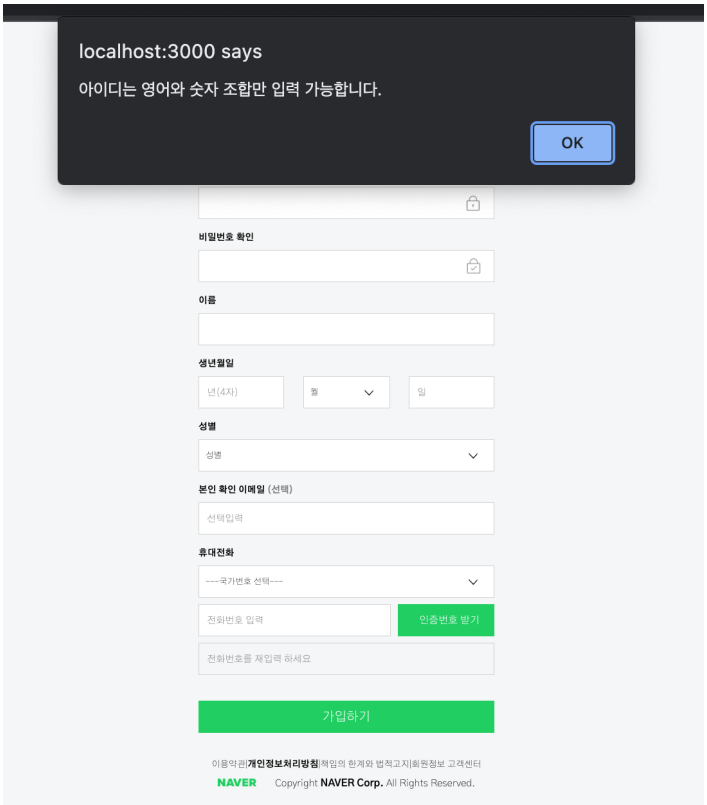
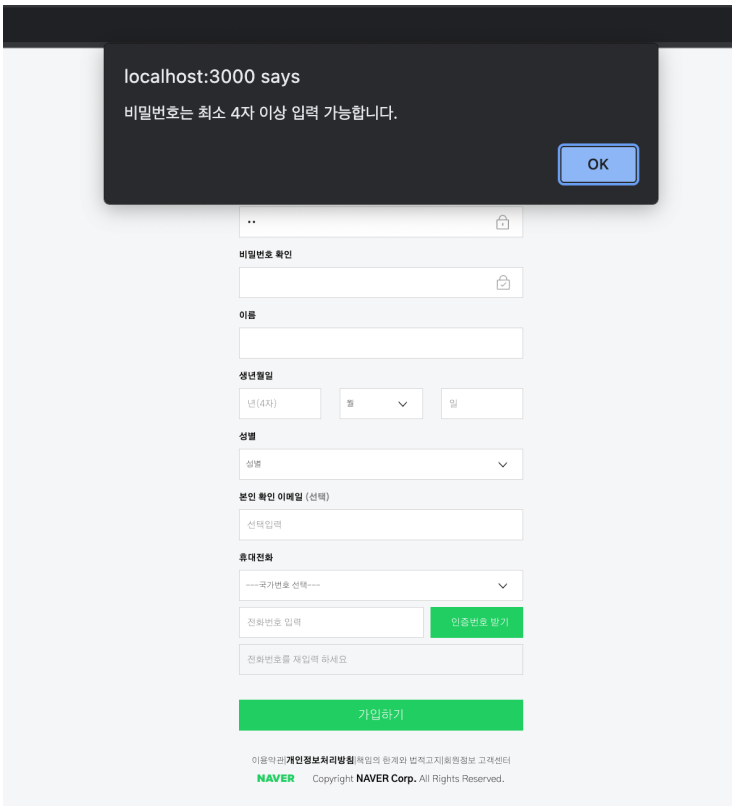
네이버 회원가입 리액트 클론코딩

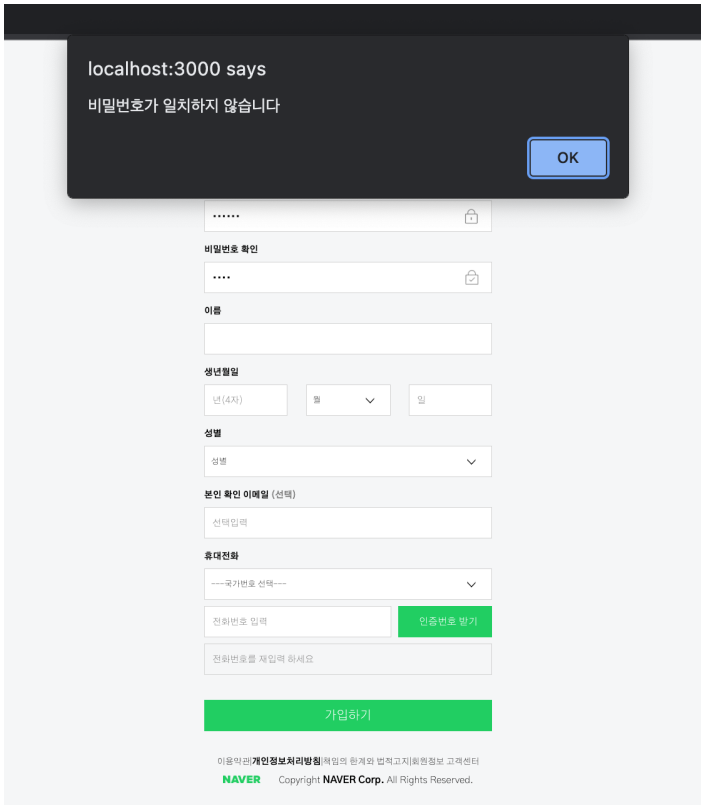


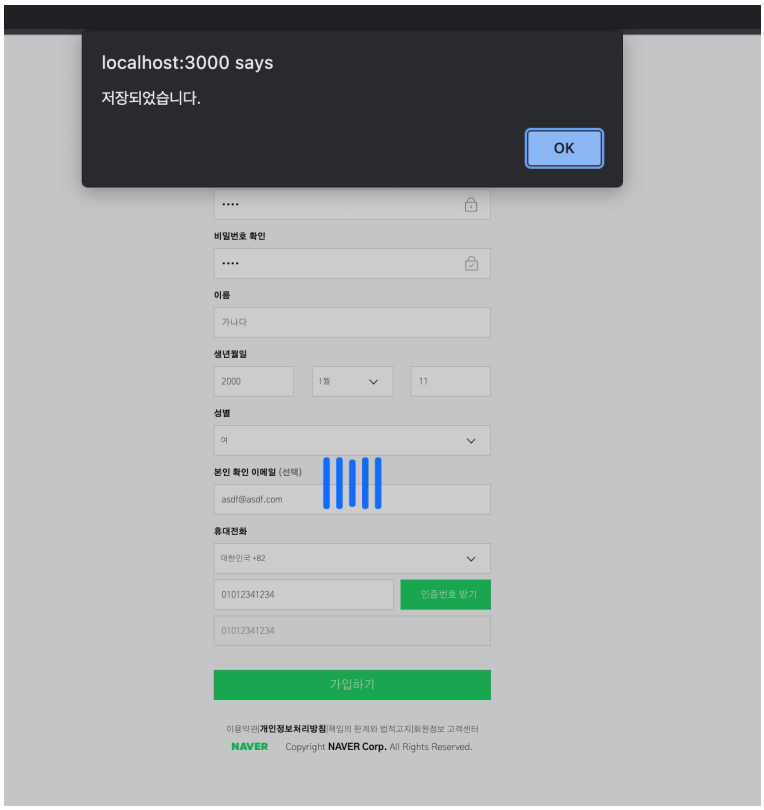



localhost:3000 says
아이디는 최소 4자 이상 입력 가능합니다.
OK
비밀번호 확인
이름
생년월일
년(4자)
월
일
성별
성별
본인 확인 이메일 (선택)
선택입력
휴대전화
---국가번호 선택---
전화번호 입력
인증번호 받기
전화번호를 재입력 하세요
가입하기
이용약관 개인정보처리방침 책임의 한계와 법적고지 회원정보 고객센터
NAVER Copyright NAVER Corp. All Rights Reserved.


localhost:3000 says
아이디는 영어와 숫자 조합만 입력 가능합니다.
OK
비밀번호 확인
이름
생년월일
년(4자)
월
일
성별
성별
본인 확인 이메일 (선택)
선택입력
휴대전화
---국가번호 선택---
전화번호 입력
인증번호 받기
전화번호를 재입력 하세요
가입하기
이용약관 개인정보처리방침 책임의 한계와 법적고지 회원정보 고객센터
NAVER Copyright NAVER Corp. All Rights Reserved.


localhost:3000 says
비밀번호는 최소 4자 이상 입력 가능합니다.
OK
비밀번호 확인
이름
생년월일
년(4자)
월
일
성별
성별
본인 확인 이메일 (선택)
선택입력
휴대전화
---국가번호 선택---
전화번호 입력
인증번호 받기
전화번호를 재입력 하세요
가입하기
이용약관 개인정보처리방침 책임의 한계와 법적고지 회원정보 고객센터
NAVER Copyright NAVER Corp. All Rights Reserved.

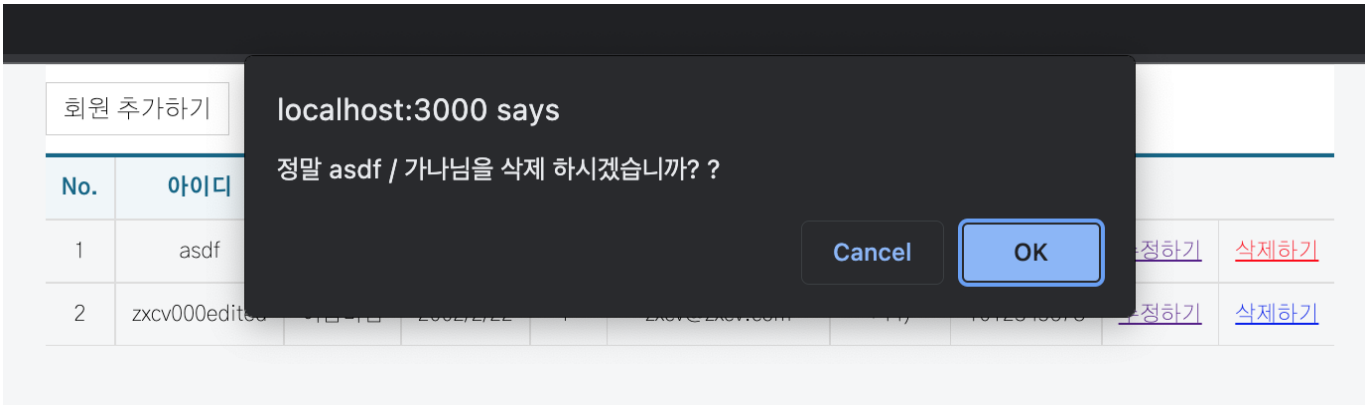

localhost:3000 says
비밀번호가 일치하지 않습니다
OK
비밀번호 확인
이름
생년월일
년(4자)
월
일
성별
성별
본인 확인 이메일 (선택)
선택입력
휴대전화
---국가번호 선택---
전화번호 입력
인증번호 받기
전화번호를 재입력 하세요
가입하기
이용약관 개인정보처리방침 책임의 한계와 법적고지 회원정보 고객센터
NAVER Copyright NAVER Corp. All Rights Reserved.



회원 추가하기

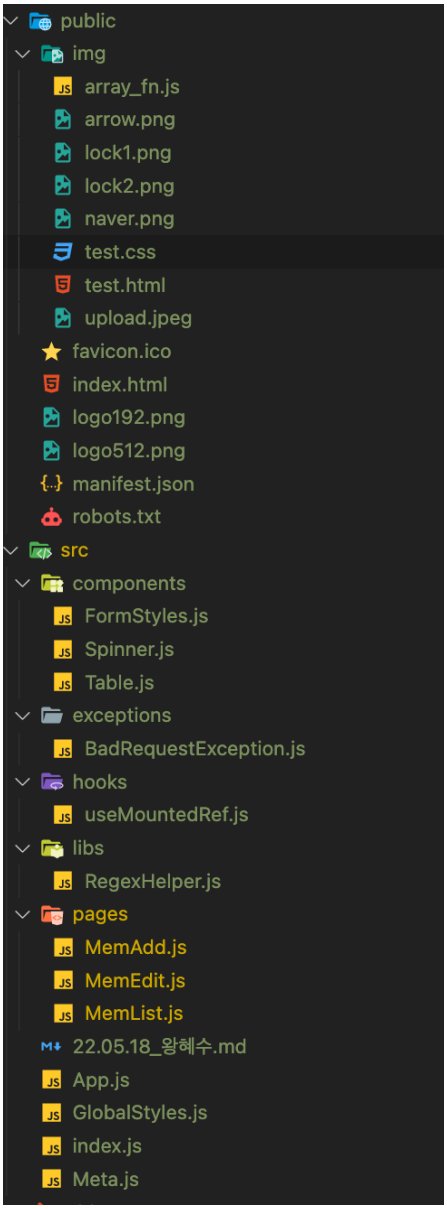
No.	아이디	이름	생년월일	성별	이메일	국가번호	전화번호		
1	asdf	가나	2000/1/11	M	yago29@naver.com	+82)	1012345678	수정하기	삭제하기
2	zxcv	다라	2002/2/22	F	zxcv@zxcv.com	+44)	1012345678	수정하기	삭제하기


회원 추가하기
No. 아이디 이름 생년월일 성별 이메일 국가번호 전화번호
1 asdf 가나 2000/1/11 M yago29@naver.com +82) 1012345678 수정하기 삭제하기
2 zxcv000edited 이름바꿈 2002/2/22 F zxcv@zxcv.com +44) 1012345678 수정하기 삭제하기


회원 추가하기
localhost:3000 says
정말 asdf / 가나님을 삭제 하시겠습니까? ?
Cancel
OK
No. 아이디
1 asdf
2 zxcv000edited
수정하기 삭제하기
수정하기 삭제하기

회원 추가하기

No.	아이디	이름	생년월일	성별	이메일	국가번호	전화번호		
2	zxcv000edited	이름바꿈	2002/2/22	F	zxcv@zxcv.com	+44)	1012345678	수정하기	삭제하기



src/index.js

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

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Meta title={'네이버 가입 폼'} />
    <GlobalStyles />
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

---

src/App.js

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

import MemList from './pages/MemList';
import MemAdd from './pages/MemAdd';
import MemEdit from './pages/MemEdit';

function App() {
  return (
    <>
      <Routes>
        <Route path="/" exapt={true} element={<MemList />} />
        <Route path="/add" element={<MemAdd />} />
        <Route path="/edit/:id" element={<MemEdit />} />
      </Routes>
    </>
  );
}

export default App;
```

---

src/components/FormStyles.js MemList, MemEdit에서 사용한 스타일컴포넌트

```
import styled from 'styled-components';

const FormStyles = styled.div`
  margin: 25px auto;
  width: 460px;
  .logo_b {
    width: 250px;
    height: 50px;
```

```
    margin: 30px auto;
  }
  .logo_s {
    width: 60px;
    height: 10px;
  }
  form {
    display: flex;
    flex-direction: column;
    justify-content: space-evenly;
  }
  .form-group {
    display: flex;
    flex-direction: column;
    justify-content: left;
    > input {
      width: 460px;
    }
  }

  input {
    height: 50px;
    border: 1px solid lightgray;
    text-indent: 10px;
    font-size: 15px;
    &:focus {
      outline: none;
      background-size: 0;
      border: 1px solid #02d05d;
    }
  }

  .gray {
    color: gray;
  }
  label {
    margin-top: 20px;
    margin-bottom: 10px;
    font-weight: 600;
    font-size: 14px;
  }
  .user_id::placeholder {
    text-align: right;
    color: gray;
  }
  .user_pw {
    background-image: url('/img/lock1.png');
    background-repeat: no-repeat;
    background-position: right 1rem center;
    background-size: 2rem;
  }
  .user_pw_re {
    background-image: url('/img/lock2.png');
    background-repeat: no-repeat;
```

```
    background-position: right 1rem center;
    background-size: 2rem;
  }
  select {
    width: 100%;
    border: 1px solid lightgray;
    height: 50px;
    text-indent: 10px;
    appearance: none;
    background-image: url('/img/arrow.png');
    background-repeat: no-repeat;
    background-position: right 1rem center;
    background-size: 2rem;
    &:focus {
      outline: none;
      background-size: 0;
      border: 1px solid #02d05d;
    }
  }

  .month {
    width: 135px;
  }
  .birth {
    width: 460px;
    display: flex;
    justify-content: space-between;
  }
  .tel {
    width: 460px;
    display: flex;
    justify-content: space-between;

    > input {
      width: 65%;
    }
  }

  .verify_btn {
    width: 150px;
    text-indent: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #02d05d;
    color: white;
  }
  .verify_num {
    width: 460px;
    height: 50px;
    border: 1px solid lightgray;
    background-color: #f5f6f7;
    color: gray;
    text-indent: 10px;
```

```
    display: flex;
    align-items: center;
  }
  .phone-box {
    height: 170px;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
  }
  button {
    border-style: none;
    width: 460px;
    height: 50px;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #02d05d;
    color: white;
    margin: 40px 0;
    font-size: 20px;
  }
  .term {
    font-size: 14px;
    font-weight: 100;
    text-align: center;
  }
  .copyr {
    display: flex;
    justify-content: space-evenly;
    font-weight: 100;
    margin: 10px 0;
  }
`;

export default FormStyles;
```

---

src/pages/MemList.js

```
import React from 'react';
import useAxios from 'axios-hooks';
import styled from 'styled-components';
import { NavLink } from 'react-router-dom';
import { useNavigate } from 'react-router-dom';
import Spinner from '../components/Spinner';
import Table from '../components/Table';

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(NavLink)`
  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 MemList = () => {
  const [member, setMember] = React.useState([]);

  const navigate = useNavigate();

  const [{ data, loading1, error }, refetch] = useAxios(
    'http://localhost:3001/member',
    {
      useCache: false,
    }
  );

  React.useEffect(() => {

    setMember(data);
  }, [data]);


  const [{ loading2 }, sendDelete] = useAxios(
    {
      method: 'DELETE',
    },
    {
      useCache: false,
      manual: true,
    }
  );

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


    const current = e.target;
    const id = parseInt(current.dataset.id);
```

```
      const name = current.dataset.name;
      const userid = current.dataset.userid;

      if (window.confirm(`정말 ${userid} / ${name}님을 삭제 하시겠습니까? ?`)) {

        (async () => {
          let json = null;

          try {
            const response = await sendDelete({
              method: 'DELETE',
              url: `http://localhost:3001/member/${id}`,
            });

            json = response.data;
          } catch (e) {
            console.error(e);
            window.alert(
              `[${e.response.status}]
${e.response.statusText}\n${e.message}`
            );
          }
          if (json != null) {

            setMember((member) => member.filter((v, i) => v.id !== id));
          }
        })();
      }
    };

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

        <LinkContainer>
          <TopLink to="add">회원 추가하기</TopLink>
        </LinkContainer>
        {error ? (
          <div>
            <h1>Oops~!!! {error.code} Error.</h1>
            <hr />
            <p>{error.message}</p>
          </div>
        ) : (
          <Table>
            <thead>
              <tr>
                <th>No.</th>
                <th>아이디</th>
                <th>이름</th>
                <th>생년월일</th>
                <th>성별</th>
                <th>이메일</th>
                <th>국가번호</th>
```

```
            <th>전화번호</th>
          </tr>
        </thead>
        <tbody>
          {member &&
            member.map(
              (
                {
                  id,
                  userid,
                  name,
                  year,
                  month,
                  date,
                  sex,
                  email,
                  country,
                  tel,
                },
                i
              ) => {
                return (
                  <tr key={id}>
                    <td>{id}</td>
                    <td>{userid}</td>
                    <td>{name}</td>
                    <td>{`${year}/${month}/${date}`}</td>
                    <td>{sex}</td>
                    <td>{email}</td>
                    <td>{country}</td>
                    <td>{tel}</td>
                    <td>
                      <NavLink to={`edit/${id}`}>수정하기</NavLink>
                    </td>
                    <td>
                      <a
                        href="#!"
                        data-id={id}
                        data-userid={userid}
                        data-name={name}
                        onClick={onDeleteClick}
                      >
                        삭제하기
                      </a>
                    </td>
                  </tr>
                );
              }
            )}
        </tbody>
      </Table>
    )}
  </>
);
```

```
};

export default React.memo(MemList);
```

---

src/pages/MemAdd.js

```
import React from 'react';
import useAxios from 'axios-hooks';

import { useNavigate } from 'react-router-dom';

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

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

const MemAdd = () => {
  const navigate = useNavigate();

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

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

    const current = e.target;

    try {
      regexHelper.value(current.user_id, '아이디를 입력하세요.');
      regexHelper.minLength(
        current.user_id,
        4,
        '아이디는 최소 4자 이상 입력 가능합니다.'
      );
      regexHelper.maxLength(
        current.user_id,
        20,
        '아이디는 최대 20자 까지 입력 가능합니다.'
      );
      regexHelper.engNum(
        current.user_id,
        '아이디는 영어와 숫자 조합만 입력 가능합니다.'
      );

      /*비밀번호 검사*/
```

```
        regexHelper.value(current.user_pw, '비밀번호를 입력하세요.');
        regexHelper.minLength(
          current.user_pw,
          4,
          '비밀번호는 최소 4자 이상 입력 가능합니다.'
        );
        regexHelper.maxLength(
          current.user_pw,
          20,
          '비밀번호는 최대 20자 까지 입력 가능합니다.'
        );
        regexHelper.compareTo(
          current.user_pw,
          current.user_pw_re,
          '비밀번호가 일치하지 않습니다'
        );

        /*이름 검사*/
        regexHelper.value(current.user_name, '이름을 입력하세요.');
        regexHelper.kor(current.user_name, '이름은 한글로 입력하세요.');
        regexHelper.minLength(
          current.user_name,
          2,
          '이름은 최소 2글자 이상입력해야 합니다.'
        );
        regexHelper.maxLength(
          current.user_name,
          10,
          '이름은 최대 10글자 까지 입력가능.'
        );

        /*생년월일*/
        regexHelper.value(current.user_birth_y, '년도를 입력하세요');
        regexHelper.year(current.user_birth_y, '정확한 년도를 입력하세요');

        regexHelper.select(current.month, '정확한 달을 선택하세요');

        regexHelper.value(current.user_birth_d, '날짜를 입력하세요');
        regexHelper.date(current.user_birth_d, '정확한 날짜를 입력하세요');

        /*성별검사*/
        regexHelper.select(current.gender, '성별을 선택하세요');

        /*이메일 검사*/
        regexHelper.value(current.email, '이메일을 입력하세요');
        regexHelper.email(current.email, '이메일 주소가 잘못되었습니다');

        /*연락처 검사*/
        regexHelper.select(current.phone_country, '국가번호를 선택하세요');
        regexHelper.value(current.tel, '연락처를 입력하세요');
        regexHelper.cellphone(current.tel, '연락처가 잘못되었습니다');

        regexHelper.compareTo(
          current.tel,
```

```
          current.verify,
          '잘못된 인증번호입니다.'
        );
      } catch (e) {
        window.alert(e.message);
        e.field.focus();
        return;
      }

      const userid = current.user_id.value;
      const pw = current.user_pw.value;
      const name = current.user_name.value;
      const year = current.user_birth_y.value;
      const month = current.month.value;
      const date = current.user_birth_d.value;
      const sex = current.gender.value;
      const email = current.email.value;
      const country = current.phone_country.value;
      const tel = current.tel.value;

      let json = null;

      (async () => {
        try {
          const response = await refetch({
            data: {
              userid: userid,
              pw: pw,
              name: name,
              year: parseInt(year),
              month: parseInt(month),
              date: parseInt(date),
              sex: sex,
              email: email,
              country: country,
              tel: parseInt(tel),
            },
          });

          json = response.data;
        } catch (e) {
          console.error(e);
          window.alert(
            `[${e.response.status}] ${e.response.statusText}\n${e.message}`
          );
        }

        if (json !== null) {
          window.alert('저장되었습니다.');
          navigate('/');
        }
      })();
    }, []);
```

```
    return (
      <>
        <Spinner visible={loading} />
        <FormStyles className="frame">
          <div className="logo_b">
            <img src="/img/naver.png" alt="" />
          </div>
          <form onSubmit={onSubmit} name="join-form">
            <div className="form-group">
              <label htmlFor="user_id">아이디</label>
              <input
                type="text"
                name="user_id"
                id="user_id"
                className="form-control user_id"
                placeholder="@naver.com   "
              />
            </div>
            <div className="form-group">
              <label htmlFor="user_pw">비밀번호</label>
              <input
                type="password"
                name="user_pw"
                id="user_pw"
                className="form-control user_pw"
              />
            </div>
            <div className="form-group">
              <label htmlFor="user_pw_re">비밀번호 확인</label>
              <input
                type="password"
                name="user_pw_re"
                id="user_pw_re"
                className="form-control user_pw_re"
              />
            </div>
            <div className="form-group">
              <label htmlFor="user_name">이름</label>
              <input
                type="text"
                name="user_name"
                id="user_name"
                className="form-control"
              />
            </div>
            <div className="form-group">
              <label htmlFor="user_birth">생년월일</label>
              <div className="birth">
                <input
                  type="text"
                  name="user_birth_y"
                  id="user_birth_y"
                  placeholder="년(4자)"
                />
```

```
            <select name="month" className="month">
              <option value="">월</option>
              <option value="1">1월</option>
              <option value="2">2월</option>
              <option value="3">3월</option>
              <option value="4">4월</option>
              <option value="5">5월</option>
              <option value="6">6월</option>
              <option value="7">7월</option>
              <option value="8">8월</option>
              <option value="9">9월</option>
              <option value="10">10월</option>
              <option value="11">11월</option>
              <option value="12">12월</option>
            </select>
            <input
              type="text"
              name="user_birth_d"
              id="user_birth_d"
              placeholder="일"
            />
          </div>
        </div>
        <div className="form-group">
          <label htmlFor="gender">성별</label>
          <select name="gender" id="gender">
            <option value="">성별</option>
            <option id="gender1" value="F">
              여
            </option>
            <option id="gender2" value="M">
              남
            </option>
          </select>
        </div>
        <div className="form-group">
          <label htmlFor="email">
            본인 확인 이메일 <span className="gray">(선택)</span>
          </label>
          <input
            type="email"
            name="email"
            id="email"
            className="form-control"
            placeholder="선택입력"
          />
        </div>
        <div className="form-group">
          <label htmlFor="phone">휴대전화</label>
          <div className="phone-box">
            <select name="phone_country" id="phone_country">
              <option value="">---국가번호 선택---</option>
              <option value="+82)">대한민국 +82</option>
              <option value="+44)">영국 +44</option>
```

```
                <option value="+61)">호주 +61</option>
              </select>
              <div className="tel">
                <input
                  type="tel"
                  name="tel"
                  id="tel"
                  placeholder="전화번호 입력"
                />
                <div className="verify_btn">인증번호 받기</div>
              </div>
              <input
                id="verify"
                name="verify"
                className="verify_num"
                placeholder="전화번호를 재입력 하세요"
              ></input>
            </div>
          </div>
          <button type="sumit">가입하기</button>
        </form>
        <div className="term">
          <span>이용약관</span>
          <span>|</span>
          <span>
            <b>개인정보처리방침</b>
          </span>
          <span>|</span>
          <span>책임의 한계와 법적고지</span>
          <span>|</span>
          <span>회원정보 고객센터</span>
        </div>
        <div className="copyr">
          <div className="logo_s">
            <img src="/img/naver.png" alt="" />
          </div>
          <span>
            Copyright<b> NAVER Corp. </b>All Rights Reserved.
          </span>
        </div>
      </FormStyles>
    </>
  );
};

export default React.memo(MemAdd);
```

---

src/pages/MemEdit.js

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

import Spinner from '../components/Spinner';
import regexHelper from '../libs/RegexHelper';
import FormStyles from '../components/FormStyles';

const MemEdit = () => {
  const { id } = useParams();

  const navigate = useNavigate();

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

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

    const current = e.target;

    try {
      /*아이디 검사*/
      regexHelper.value(current.user_id, '아이디를 입력하세요.');
      regexHelper.minLength(
        current.user_id,
        4,
        '아이디는 최소 4자 이상 입력 가능합니다.'
      );
      regexHelper.maxLength(
        current.user_id,
        20,
        '아이디는 최대 20자 까지 입력 가능합니다.'
      );
      regexHelper.engNum(
        current.user_id,
        '아이디는 영어와 숫자 조합만 입력 가능합니다.'
      );

      /*비밀번호 검사*/
      regexHelper.value(current.user_pw, '비밀번호를 입력하세요.');
      regexHelper.minLength(
        current.user_pw,
        4,
        '비밀번호는 최소 4자 이상 입력 가능합니다.'
      );
      regexHelper.maxLength(
        current.user_pw,
        20,
        '비밀번호는 최대 20자 까지 입력 가능합니다.'
      );
      regexHelper.compareTo(
```

```
        current.user_pw,
        current.user_pw_re,
        '비밀번호 확인이 잘못되었습니다'
      );

      /*이름 검사*/
      regexHelper.value(current.user_name, '이름을 입력하세요.');
      regexHelper.kor(current.user_name, '이름은 한글로 입력하세요.');
      regexHelper.minLength(
        current.user_name,
        2,
        '이름은 최소 2글자 이상입력해야 합니다.'
      );
      regexHelper.maxLength(
        current.user_name,
        10,
        '이름은 최대 10글자 까지 입력가능.'
      );

      /*생년월일*/
      regexHelper.value(current.user_birth_y, '년도를 입력하세요');
      regexHelper.year(current.user_birth_y, '정확한 년도를 입력하세요');

      regexHelper.select(current.month, '정확한 달을 선택하세요');

      regexHelper.value(current.user_birth_d, '날짜를 입력하세요');
      regexHelper.date(current.user_birth_d, '정확한 날짜를 입력하세요');

      /*성별검사*/
      regexHelper.select(current.gender, '성별을 선택하세요');

      /*이메일 검사*/
      regexHelper.value(current.email, '이메일을 입력하세요');
      regexHelper.email(current.email, '이메일 주소가 잘못되었습니다');

      /*연락처 검사*/
      regexHelper.select(current.phone_country, '국가번호를 선택하세요');
      regexHelper.value(current.tel, '연락처를 입력하세요');
      regexHelper.cellphone(current.tel, '연락처가 잘못되었습니다');

      regexHelper.compareTo(
        current.tel,
        current.verify,
        '잘못된 인증번호입니다.'
      );
    } catch (e) {
      window.alert(e.message);
      e.field.focus();
      return;
    }

    const userid = current.user_id.value;
    const pw = current.user_pw.value;
    const name = current.user_name.value;
```

```
      const year = current.user_birth_y.value;
      const month = current.month.value;
      const date = current.user_birth_d.value;
      const sex = current.gender.value;
      const email = current.email.value;
      const country = current.phone_country.value;
      const tel = current.tel.value;

      let json = null;

      (async () => {
        try {
          const response = await refetch({
            method: 'PUT',
            data: {
              userid: userid,
              pw: pw,
              name: name,
              year: parseInt(year),
              month: parseInt(month),
              date: parseInt(date),
              sex: sex,
              email: email,
              country: country,
              tel: parseInt(tel),
            },
          });

          json = response.data;
        } catch (e) {
          console.error(e);
          window.alert(
            `[${e.response.status}] ${e.response.statusText}\n${e.message}`
          );
        }


        if (json !== null) {
          window.alert('저장되었습니다.');

          navigate('/');
        }
      })();
    }, []);

    return (
      <>
        <Spinner visible={loading} />
        <FormStyles className="frame">
          {error ? (
            <div>
              <h1>Oops~!!! {error.code} Error.</h1>
              <hr />
              <p>{error.message}</p>
```

```
            </div>
          ) : (
            data && (
              <form onSubmit={onSubmit} name="join-form">
                <div className="form-group">
                  <label htmlFor="user_id">아이디</label>
                  <input
                    type="text"
                    name="user_id"
                    id="user_id"
                    className="form-control user_id"
                    placeholder="@naver.com   "
                    defaultValue={data.userid}
                  />
                </div>
                <div className="form-group">
                  <label htmlFor="user_pw">비밀번호</label>
                  <input
                    type="password"
                    name="user_pw"
                    id="user_pw"
                    className="form-control user_pw"
                    defaultValue={data.pw}
                  />
                </div>
                <div className="form-group">
                  <label htmlFor="user_pw_re">비밀번호 확인</label>
                  <input
                    type="password"
                    name="user_pw_re"
                    id="user_pw_re"
                    className="form-control user_pw_re"
                    defaultValue={data.pw}
                  />
                </div>
                <div className="form-group">
                  <label htmlFor="user_name">이름</label>
                  <input
                    type="text"
                    name="user_name"
                    id="user_name"
                    className="form-control"
                    defaultValue={data.name}
                  />
                </div>
                <div className="form-group">
                  <label htmlFor="user_birth">생년월일</label>
                  <div className="birth">
                    <input
                      type="text"
                      name="user_birth_y"
                      id="user_birth_y"
                      placeholder="년(4자)"
                      defaultValue={data.year}
```

```
                    />
                    <select
                      name="month"
                      className="month"
                      defaultValue={data.month}
                    >
                      <option value="">월</option>
                      <option value="1">1월</option>
                      <option value="2">2월</option>
                      <option value="3">3월</option>
                      <option value="4">4월</option>
                      <option value="5">5월</option>
                      <option value="6">6월</option>
                      <option value="7">7월</option>
                      <option value="8">8월</option>
                      <option value="9">9월</option>
                      <option value="10">10월</option>
                      <option value="11">11월</option>
                      <option value="12">12월</option>
                    </select>
                    <input
                      type="text"
                      name="user_birth_d"
                      id="user_birth_d"
                      placeholder="일"
                      defaultValue={data.date}
                    />
                  </div>
                </div>
                <div className="form-group">
                  <label htmlFor="gender">성별</label>
                  <select name="gender" id="gender" defaultValue={data.sex}>
                    <option value="">성별</option>
                    <option id="gender1" value="F">
                      여
                    </option>
                    <option id="gender2" value="M">
                      남
                    </option>
                  </select>
                </div>
                <div className="form-group">
                  <label htmlFor="email">
                    본인 확인 이메일 <span className="gray">(선택)</span>
                  </label>
                  <input
                    type="email"
                    name="email"
                    id="email"
                    className="form-control"
                    placeholder="선택입력"
                    defaultValue={data.email}
                  />
                </div>
```

```
              <div className="form-group">
                <label htmlFor="phone">휴대전화</label>
                <div className="phone-box">
                  <select
                    name="phone_country"
                    id="phone_country"
                    defaultValue={data.country}
                  >
                    <option value="">---국가번호 선택---</option>
                    <option value="+82)">대한민국 +82</option>
                    <option value="+44)">영국 +44</option>
                    <option value="+61)">호주 +61</option>
                  </select>
                  <div className="tel">
                    <input
                      type="tel"
                      name="tel"
                      id="tel"
                      placeholder="전화번호 입력"
                      defaultValue={data.tel}
                    />
                    <div className="verify_btn">인증번호 받기</div>
                  </div>
                  <input
                    id="verify"
                    name="verify"
                    className="verify_num"
                    placeholder="전화번호를 재입력 하세요"
                  ></input>
                </div>
              </div>
              <button type="sumit">수정하기</button>
            </form>
          )
        )}
      </FormStyles>
    </>
  );
};

export default React.memo(MemEdit);
```

> 예제와 일치하는 코드는 참조하지 않았습니다