

프로젝트 진행 순서

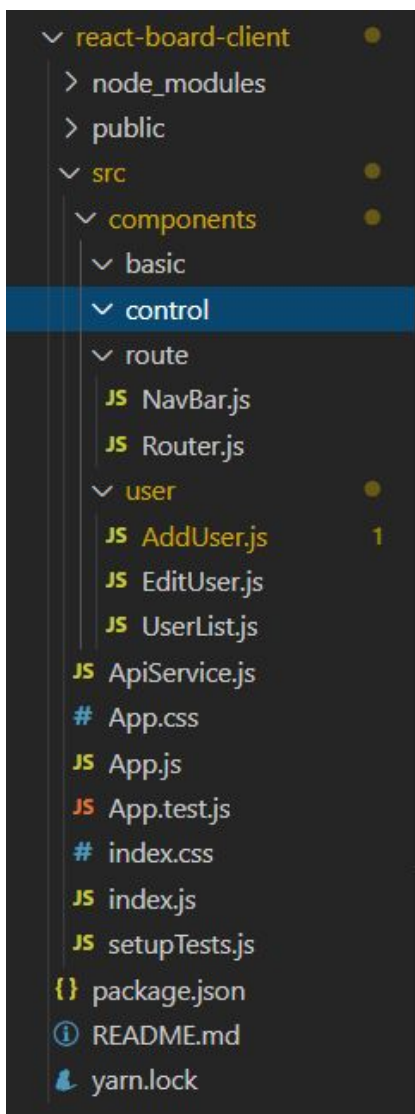
- A. 사용자 추가 기능 처리
- B. 로그인 처리
- C. 게시판 처리

A 사용자 추가 기능

- 1. 리액트 앱 설치

`npx create-react-app react-board-client`

리액트 프로젝트 모두 정리



폴더 생성 user, route

js 생성 ApiService, UserList, AddUser, EditUser, Router, NavBar

2. yarn 설치

<https://classic.yarnpkg.com/en/>

3. 노드 모듈 설치

```
> yarn add axios  
> yarn add @material-ui/core  
> yarn add @material-ui/icon  
> yarn add react-router-dom
```

axios : REST API 사용 라이브러리

material : 머티리얼 템플릿 UI 라이브러리

react-router-dom : 화면 전환 라우팅 라이브러리

4. ApiService.js

```
import axios from 'axios';  
  
const USER_API_BASE_URL = 'http://localhost:8090/users';  
  
class ApiService {  
  
  fetchUsers() {  
    return axios.get(USER_API_BASE_URL);  
  }  
  
  fetchUserByID(id) {  
    return axios.get(USER_API_BASE_URL + '/' + id);  
  }  
  
  addUser(user) {  
    return axios.post(USER_API_BASE_URL, user);  
  }  
  
  editUser(user) {  
    return axios.put(USER_API_BASE_URL + '/' + user.id, user);  
  }  
  
  deleteUser(id) {  
    return axios.delete(USER_API_BASE_URL + '/' + id);  
  }  
}
```

```
}  
  
export default new ApiService();
```

5. UserList.js

```
import React, { Component } from "react";  
import ApiService from '../../ApiService';  
  
import Table from '@material-ui/core/Table';  
import TableBody from '@material-ui/core/TableBody';  
import TableCell from '@material-ui/core/TableCell';  
import TableHead from '@material-ui/core/TableHead';  
import TableRow from '@material-ui/core/TableRow';  
import Button from '@material-ui/core/Button';  
import Typography from '@material-ui/core/Typography';  
import CreateIcon from '@material-ui/icons/Create';  
import DeleteIcon from '@material-ui/icons/Delete';  
  
class UserList extends Component {  
  constructor(props) {  
    super(props);  
  
    this.state = {  
      users: [],  
      message: null,  
    }  
  }  
  
  // https://www.zerocho.com/category/React/post/579b5ec26958781500ed9955  
  componentDidMount() {  
    console.log('UserList componentTableCellMount()');  
    this.reloadUserList();  
  }  
  
  reloadUserList = () => {  
    ApiService.fetchUsers()  
      .then(res => {  
        this.setState({ users: res.data })  
      })  
  }
```

```

        .catch(err => {
            console.error('reloadUserList() 에러!', err);
        })
    }

    addUser = () => {
        window.localStorage.removeItem('id');
        this.props.history.push('/add-user');
    }

    editUser = (id) => {
        window.localStorage.setItem('id', id);
        this.props.history.push('/edit-user')
    }

    deleteUser = (id) => {
        ApiService.deleteUser(id)
            .then(res => {
                this.setState({
                    message: '성공적으로 삭제했습니다',
                    users: this.state.users.filter(user => user.id !== id)
                });
            })
            .catch(err => {
                console.error('deleteUser() 에러!', err);
            })
    }

    render() {
        return (
            <div>
                <Typography variant="h4" style={style}>User List</Typography>
                <Button variant="contained" color="primary"
onClick={this.addUser}>Add User</Button>
                <Table>
                    <TableHead>
                        <TableRow>
                            <TableCell>Id</TableCell>

```

```

        <TableCell>UserName</TableCell>
        <TableCell>FirstName</TableCell>
        <TableCell>LastName</TableCell>
        <TableCell>Gender</TableCell>
        <TableCell>Salary</TableCell>
        <TableCell align="center">Edit</TableCell>
        <TableCell align="center">Delete</TableCell>
    </TableRow>
</TableHead>
<TableBody>
    {this.state.users.map(user =>
        <TableRow key={user.id}>
            <TableCell component="th"
socpe="user">{user.id}</TableCell>
            <TableCell
align="left">{user.userName}</TableCell>
            <TableCell
align="left">{user.firstName}</TableCell>
            <TableCell
align="left">{user.lastName}</TableCell>
            <TableCell align="left">{user.gender}</TableCell>
            <TableCell align="left">{user.salary}</TableCell>
            <TableCell align="center" onClick={() =>
this.editUser(user.id)}>
                <CreateIcon />
            </TableCell>
            <TableCell align="center" onClick={() =>
this.deleteUser(user.id)}>
                <DeleteIcon />
            </TableCell>
        </TableRow>
    )}
    </TableBody>
</Table>
</div>
);
}
}

```

```
const style = {
  display: 'flex',
  justifyContent: 'center',
}

export default UserList;
```

6. AddUser.js

```
import React, { Component } from "react";
import ApiService from '../ApiService';

import TextField from '@material-ui/core/TextField';
import InputLabel from '@material-ui/core/InputLabel';
import Button from '@material-ui/core/Button';
import Select from '@material-ui/core/Select';
import Typography from '@material-ui/core/Typography';

class AddUser extends Component {
  constructor(props) {
    super(props);

    this.state = {
      userName: '',
      password: '',
      firstName: '',
      lastName: '',
      gender: '',
      salary: '',
      message: null,
    }
  }

  onChange = (e) => {
    this.setState({
      [e.target.name]: e.target.value
    });
  }
}
```

```

saveUser = (e) => {
  e.preventDefault();
  // https://gist.github.com/LeoHeo/7c2a2a6dbcf80becaaa1e61e90091e5d
  let user = {
    userName: this.state.userName,
    password: this.state.password,
    firstName: this.state.firstName,
    lastName: this.state.lastName,
    gender: this.state.gender,
    salary: this.state.salary,
  }

  ApiService.addUser(user)
    .then(res => {
      this.setState({
        message: user.userName + '님이 성공적으로 등록되었습니다'
      })
      console.info(this.state.message);
      this.props.history.push('/users');
    })
    .catch(err => {
      console.error('saveUser() 에러', err);
    })
  }

render() {
  return (
    <div>
      <Typography variant="h4" style={style}>Add User</Typography>
      <form sytle={formContainer}>
        <TextField type="text" id="userName" name="userName"
label="Input your username"
          fullWidth margin="normal" value={this.state.userName}
onChange={this.onChange} />

        <TextField type="password" id="password" name="password"
label="Input your password"
          fullWidth margin="normal" value={this.state.password}
onChange={this.onChange} />

```

```

        <TextField type="text" id="firstName" name="firstName"
label="Input your firstName"
        fullWidth margin="normal" value={this.state.firstName}
onChange={this.onChange} />
        <TextField type="text" id="lastName" name="lastName"
label="Input your lastName"
        fullWidth margin="normal" value={this.state.lastName}
onChange={this.onChange} />
        <InputLabel
htmlFor="gender-native-helper">Gender</InputLabel>
        <Select
        native fullWidth
        value={this.state.gender}
        onChange={this.onChange}
        label="Select your gender"
        inputProps={{
            name: 'gender',
            id: 'gender-native-helper',
        }}>
        <option aria-label="None" value=""></option>
        <option value={'M'}>Male</option>
        <option value={'F'}>Female</option>
        </Select>
        <TextField type="number" id="salary" name="salary"
label="Input your salary"
        fullWidth margin="normal" value={this.state.salary}
onChange={this.onChange} />
        <Button variant="contained" color="primary"
onClick={this.saveUser}>Save</Button>
    </form>
</div>
    );
}
}

const style = {
    display: 'flex',
    justifyContent: 'center',

```



```

}

const formContainer = {
  display: 'flex',
  flexFlow: 'row wrap',
}

export default AddUser;

```

7. EditUser.js

```

import React, { Component } from "react";
import ApiService from '../../ApiService';

import TextField from '@material-ui/core/TextField';
import InputLabel from '@material-ui/core/InputLabel';
import Button from '@material-ui/core/Button';
import Select from '@material-ui/core/Select';
import Typography from '@material-ui/core/Typography';

class EditUser extends Component {
  constructor(props) {
    super(props);

    this.state = {
      id: '',
      //userName: '',
      password: '',
      firstName: '',
      lastName: '',
      gender: '',
      salary: '',
      message: null,
    }
  }

  componentDidMount() {
    this.loadUser();
  }
}

```

```

loadUser = () => {
  ApiService.fetchUserByID(window.localStorage.getItem("id"))
    .then(res => {
      let user = res.data;
      this.setState({
        id: user.id,
        password: user.password,
        userName: user.userName,
        firstName: user.firstName,
        lastName: user.lastName,
        gender: user.gender,
        salary: user.salary,
      })
    })
    .catch(err => {
      console.error('loadUser() 에러', err);
    })
}

onChange = (e) => {
  this.setState({
    [e.target.name]: e.target.value
  });
}

saveUser = (e) => {
  e.preventDefault();
  // https://gist.github.com/LeoHeo/7c2a2a6dbcf80becaaa1e61e90091e5d
  let user = {
    id: this.state.id,
    //userName: this.state.userName,
    password: this.state.password,
    firstName: this.state.firstName,
    lastName: this.state.lastName,
    gender: this.state.gender,
    salary: this.state.salary,
  }

  ApiService.editUser(user)

```

```

        .then(res => {
            this.setState({
                message: user.userName + '님 정보가 수정되었습니다'
            })
            this.props.history.push('/users');
        })
        .catch(err => {
            console.error('saveUser() 에러', err);
        })
    }

    render() {
        return (
            <div>
                <Typography variant="h4" style={style}>Edit User</Typography>
                <form>
                    <TextField type="text" id="userName" name="userName"
aria-readonly
                        fullWidth margin="normal" value={this.state.userName} />

                    <TextField type="password" id="password" name="password"
                        fullWidth margin="normal" value={this.state.password}
onChange={this.onChange} />
                    <TextField type="text" id="firstName" name="firstName"
                        fullWidth margin="normal" value={this.state.firstName}
onChange={this.onChange} />
                    <TextField type="text" id="lastName" name="lastName"
                        fullWidth margin="normal" value={this.state.lastName}
onChange={this.onChange} />
                    <InputLabel
htmlFor="gender-native-helper">Gender</InputLabel>
                    <Select
                        native fullWidth
                        value={this.state.gender}
                        onChange={this.onChange}
                        label="Select your gender"
                        inputProps={{
                            name: 'gender',

```

```

        id: 'gender-native-helper',
      }}>
      <option aria-label="None" value=""></option>
      <option value={ 'M' }>Male</option>
      <option value={ 'F' }>Female</option>
    </Select>
    <TextField type="number" id="salary" name="salary"
      fullWidth margin="normal" value={this.state.salary}
onChange={this.onChange} />
    <Button variant="contained" color="primary"
onClick={this.saveUser}>Save</Button>
  </form>
</div>
);
}
}

const style = {
  display: 'flex',
  justifyContent: 'center',
}

export default EditUser;

```

8. Router.js

```

import React from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom';
import UserList from '../user/UserList';
import AddUser from '../user/AddUser';
import EditUser from '../user/EditUser';

const Router = () => {
  return (
    <div>
      <BrowserRouter>
        <div style={style}>
          <Switch>
            <Route exact path="/" component={UserList} />
            <Route path="/users" component={UserList} />

```

```

        <Route path="/add-user" component={AddUser} />
        <Route path="/edit-user" component={EditUser} />
      </Switch>
    </div>
  </BrowserRouter>
</div>
);
}

const style = {
  margintop: '20px',
}

export default Router;

```

9. NavBar.js

```

import React from 'react';

import AppBar from '@material-ui/core/AppBar';
import ToolBar from '@material-ui/core/Toolbar';
import Typography from '@material-ui/core/Typography';
import Button from '@material-ui/core/Button';
import IconButton from '@material-ui/core/IconButton';
import MenuIcon from '@material-ui/icons/Menu';

const NavBar = () => {
  return (
    <div>
      <AppBar position="static">
        <ToolBar>
          <IconButton edge="start" color="inherit" aria-label="Menu">
            <MenuIcon />
          </IconButton>
          <Typography variant="h6" style={style}>
            React User Application
          </Typography>
          <Button color="inherit">Login</Button>
        </ToolBar>
      </AppBar>
    </div>
  );
}

```

```
        </div>
      );
    }

    const style = {
      flexGrow: 1
    }

    export default NavBar;
```

10. App.js 수정

```
import './App.css';
import AppRouter from './components/route/Router';
import NavBar from './components/route/NavBar';

import Container from '@material-ui/core/Container';

function App() {
  console.log('App component render()');
  return (
    <div className="App">
      <NavBar />
      <Container>
        <AppRouter />
      </Container>
    </div>
  );
}

export default App;
```

React Board

x +

localhost:3000/users

React User Application

LOGIN

User List

ADD USER

Id	UserName	FirstName	LastName	Gender	Salary	Edit	Delete
----	----------	-----------	----------	--------	--------	------	--------

React Board

x +

localhost:3000/add-user

React User Application

LOGIN

Add User

Input your username

Input your password

Input your firstName

Input your lastName

Gender

Input your salary

SAVE

React Board x +

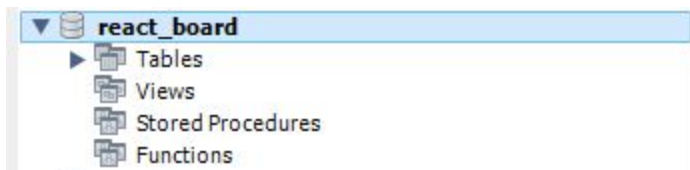
localhost:3000/edit-user

React User Application LOGIN

Edit User

Gender

11. MySQL 준비



```
1 • CREATE TABLE users (  
2     id int NOT NULL AUTO_INCREMENT,  
3     userName varchar(12) NOT NULL,  
4     password varchar(20) NOT NULL,  
5     firstName varchar(20) NOT NULL,  
6     lastName varchar(20) NOT NULL,  
7     gender char(1) DEFAULT NULL,  
8     salary int DEFAULT NULL,  
9     PRIMARY KEY (id)  
10 );
```

react_board 스키마(데이터베이스) 생성
users 테이블 생성

12. Spring Boot 프로젝트 생성 STS 실행, New Spring Starter Project 실행

New Spring Starter Project

⚠ A project with name 'react-board-server' already exists in the workspace.

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

☒ Add project to working sets

Working sets:

선택 Dependencies - Spring Boot DevTools, Lombok, MyBatis Framework, MySQL Driver, Spring Web

13. application.properties 설정

서버 설정

```
server.port=8090
```

#MySQL 설정

```
spring.datasource.url=jdbc:mysql://localhost:3306/react_board?useUnicode=true&characterEncoding=utf8&verifyServerCertificate=false&useSSL=false&serverTimezone=UTC
```

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

```
spring.datasource.username=root
```

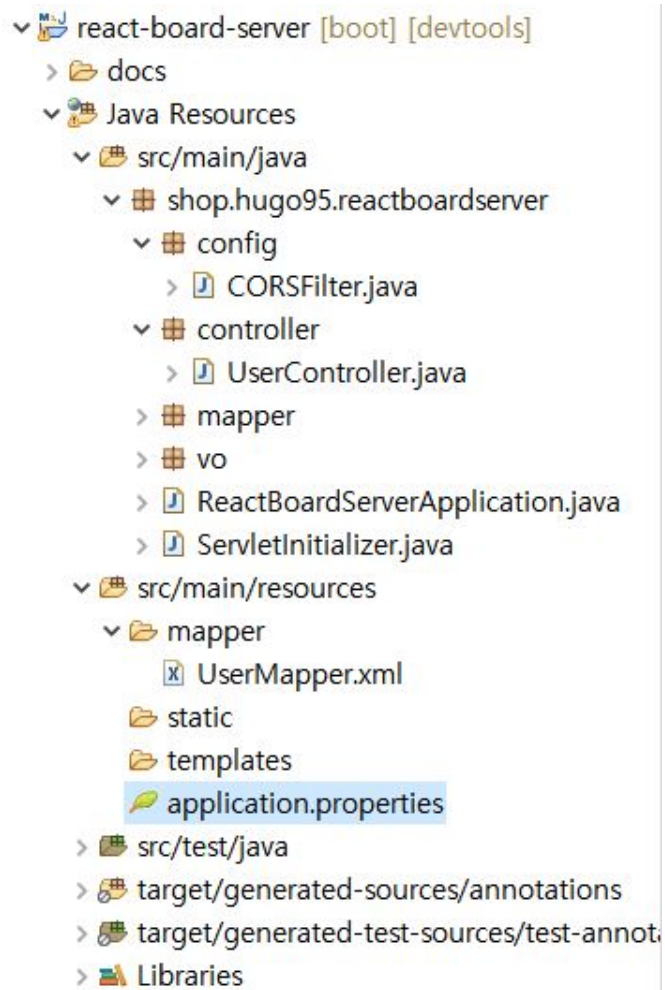
```
spring.datasource.password=mysql_p@ssw0rd
```

#MyBatis 설정

```
mybatis.mapper-locations=classpath*:mapper/*.xml
```

14. 프로젝트 폴더 구성

config, controller, mapper,,vo 순으로 구성



```
react-board-server [boot] [devtools]
├── docs
├── Java Resources
│   ├── src/main/java
│   │   ├── shop.hugo95.reactboardserver
│   │   │   ├── config
│   │   │   │   └── CORSFilter.java
│   │   │   ├── controller
│   │   │   │   └── UserController.java
│   │   │   ├── mapper
│   │   │   ├── vo
│   │   │   ├── ReactBoardServerApplication.java
│   │   │   └── ServletInitializer.java
│   │   └── src/main/resources
│   │       ├── mapper
│   │       │   └── UserMapper.xml
│   │       ├── static
│   │       ├── templates
│   │       └── application.properties
│   ├── src/test/java
│   ├── target/generated-sources/annotations
│   ├── target/generated-test-sources/test-annot
│   └── Libraries
```

15. src/main/java/project-name/config/CORSFilter.java

```
package shop.hugo95.reactboardserver.config;

import java.io.IOException;

import javax.servlet.Filter;
import javax.servlet.FilterChain;
import javax.servlet.FilterConfig;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequestResponse;

import org.springframework.context.annotation.Configuration;

@Configuration
public class CORSFilter implements Filter {

    @Override
    public void doFilter(ServletRequest request, ServletResponse
response, FilterChain chain)
        throws IOException, ServletException {

        System.out.println("on the way of Filtering...");
        HttpServletRequestResponse res = (HttpServletRequestResponse) response;
        res.setHeader("Access-Control-Allow-Origin", "*");
        res.setHeader("Access-Control-Allow-Credentials", "true");
        res.setHeader("Access-Control-Allow-Methods", "POST, GET, PUT,
DELETE, PATCH, OPTIONS");
        res.setHeader("Access-Control-Allow-Max-Age", "3600");
        res.setHeader("Access-Control-Allow-Headers", "X-PINGOTHER,
X-Requested-With, Content-Type, "
            + "Authorization, Origin, Accept,
Access-Control-Request-Method, Access-Control-Request-Headers");

        chain.doFilter(request, response);
    }

    public void init(FilterConfig filterConfig) {}

    public void destroy() {}
}
```

16. src/main/java/project-name/vo/UserVO.java

```
package shop.hugo95.reactboardserver.vo;

import lombok.Data;

@Data
public class UserVO {
    Long id;
    String userName;
    String password;
    String firstName;
    String lastName;
    String gender;
    int salary;
}
```

17. src/main/resources/mapper/UserMapper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTDMapper3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="shop.hugo95.reactboardserver.mapper.UserMapper">
    <insert id="insertUser"
parameterType="shop.hugo95.reactboardserver.vo.UserVO">
        INSERT INTO users
            (id, userName, password, firstName, lastName, gender,
salary)
        VALUES
            (#{id}, #{userName}, #{password}, #{firstName},
#{lastName}, #{gender}, #{salary})
    </insert>

    <select id="userList"
resultType="shop.hugo95.reactboardserver.vo.UserVO">
        SELECT * FROM users
        ORDER BY id
    </select>

    <select id="fetchUserByID"
resultType="shop.hugo95.reactboardserver.vo.UserVO"
parameterType="Long">
        SELECT * FROM users
        WHERE id = #{id}
    </select>
```

```

        <update id="updateUser"
parameterType="shop.hugo95.reactboardserver.vo.UserVO">
            UPDATE users
            SET
                password = #{password},
                firstName = #{firstName},
                lastName = #{lastName},
                gender = #{gender},
                salary = #{salary}
            WHERE id = #{id}
        </update>

        <delete id="deleteUser"
parameterType="shop.hugo95.reactboardserver.vo.UserVO">
            DELETE FROM users
            WHERE id = #{id}
        </delete>
    </mapper>

```

18. src/main/java/project-name/mapper/UserMapper.java

```

package shop.hugo95.reactboardserver.mapper;

import java.util.List;

import org.apache.ibatis.annotations.Mapper;

import shop.hugo95.reactboardserver.vo.UserVO;

@Mapper
public interface UserMapper {
    List<UserVO> userList();
    UserVO fetchUserByID(Long id);
    void updateUser(UserVO user);
    void insertUser(UserVO user);
    void deleteUser(Long id);
}

```

19. src/main/java/project-name/controller/UserController.java

```

package shop.hugo95.reactboardserver.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.CrossOrigin;

```

```

import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import shop.hugo95.reactboardserver.mapper.UserMapper;
import shop.hugo95.reactboardserver.vo.UserVO;

@CrossOrigin(origins = "*", maxAge = 3600)
@RestController
@RequestMapping("/users")
public class UserController {

    @Autowired
    UserMapper userMapper;

    @GetMapping
    public List<UserVO> userList() {
        System.out.println(userMapper.userList());
        return userMapper.userList();
    }

    @PostMapping
    void insertUser(@RequestBody UserVO user) {
        userMapper.insertUser(user);
        System.out.println("유저 데이터 저장");
    }

    @GetMapping("/{id}")
    public UserVO fetchUserByID(@PathVariable Long id) {
        System.out.println(userMapper.fetchUserByID(id));
        UserVO fetchUser = userMapper.fetchUserByID(id);
        return fetchUser;
    }

    @PutMapping("/{id}")
    public void updateUser(@PathVariable Long id, @RequestBody UserVO
user) {
        UserVO updateUser = user;
        System.out.println("업데이트 유저 => " + updateUser);
        updateUser.setPassword(user.getPassword());
        updateUser.setFirstName(user.getFirstName());
        updateUser.setLastName(user.getLastName());
        updateUser.setGender(user.getGender());
        updateUser.setSalary(user.getSalary());
    }
}

```

```

        userMapper.updateUser(updateUser);
    }

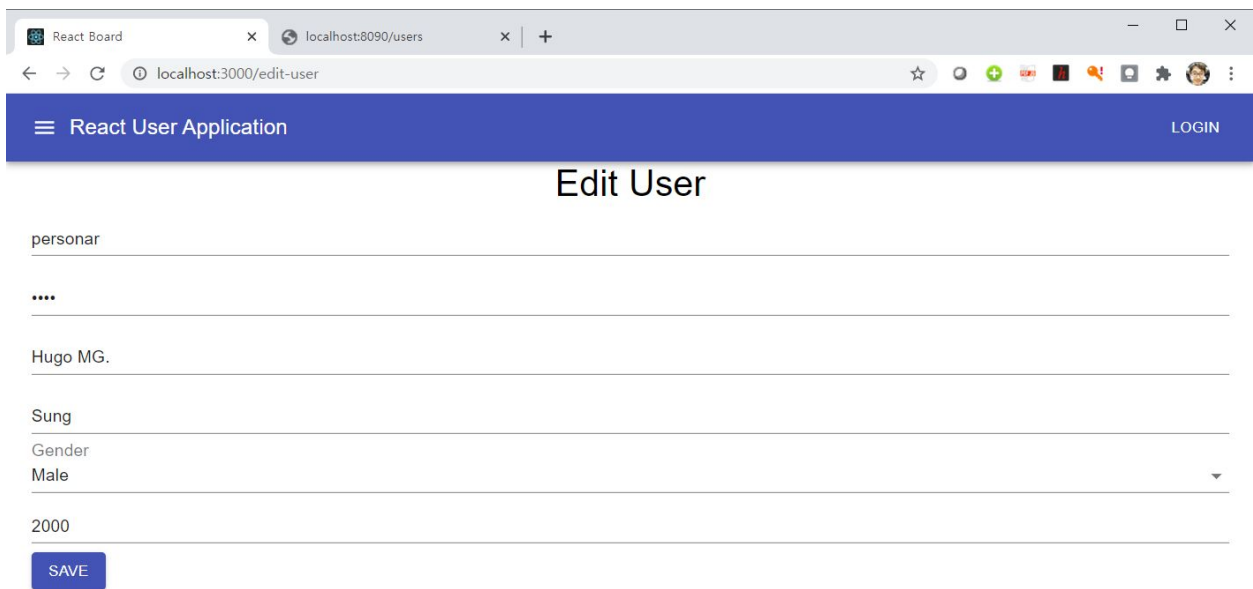
    @DeleteMapping("/{id}")
    public void deleteUser(@PathVariable Long id) {
        userMapper.deleteUser(id);
        System.out.println("유저 삭제!");
    }
}

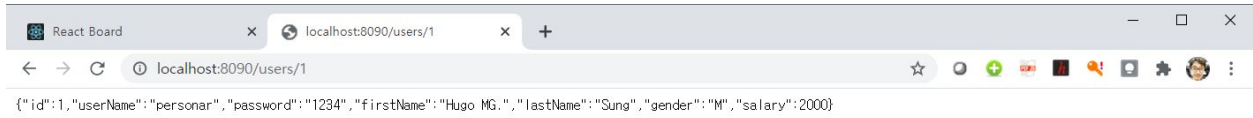
```

20. 실행결과

The screenshot shows a web browser window with the URL `localhost:3000`. The page title is "React User Application" and there is a "LOGIN" link in the top right. The main content area is titled "User List". On the left, there is a blue button labeled "ADD USER". Below this, there is a table with the following data:

Id	UserName	FirstName	LastName	Gender	Salary	Edit	Delete
1	personar	Hugo MG.	Sung	M	2000		
3	test	test	test	M	2222		
4	Ashely	ES	Park	F	5000		





完