

# REPORT



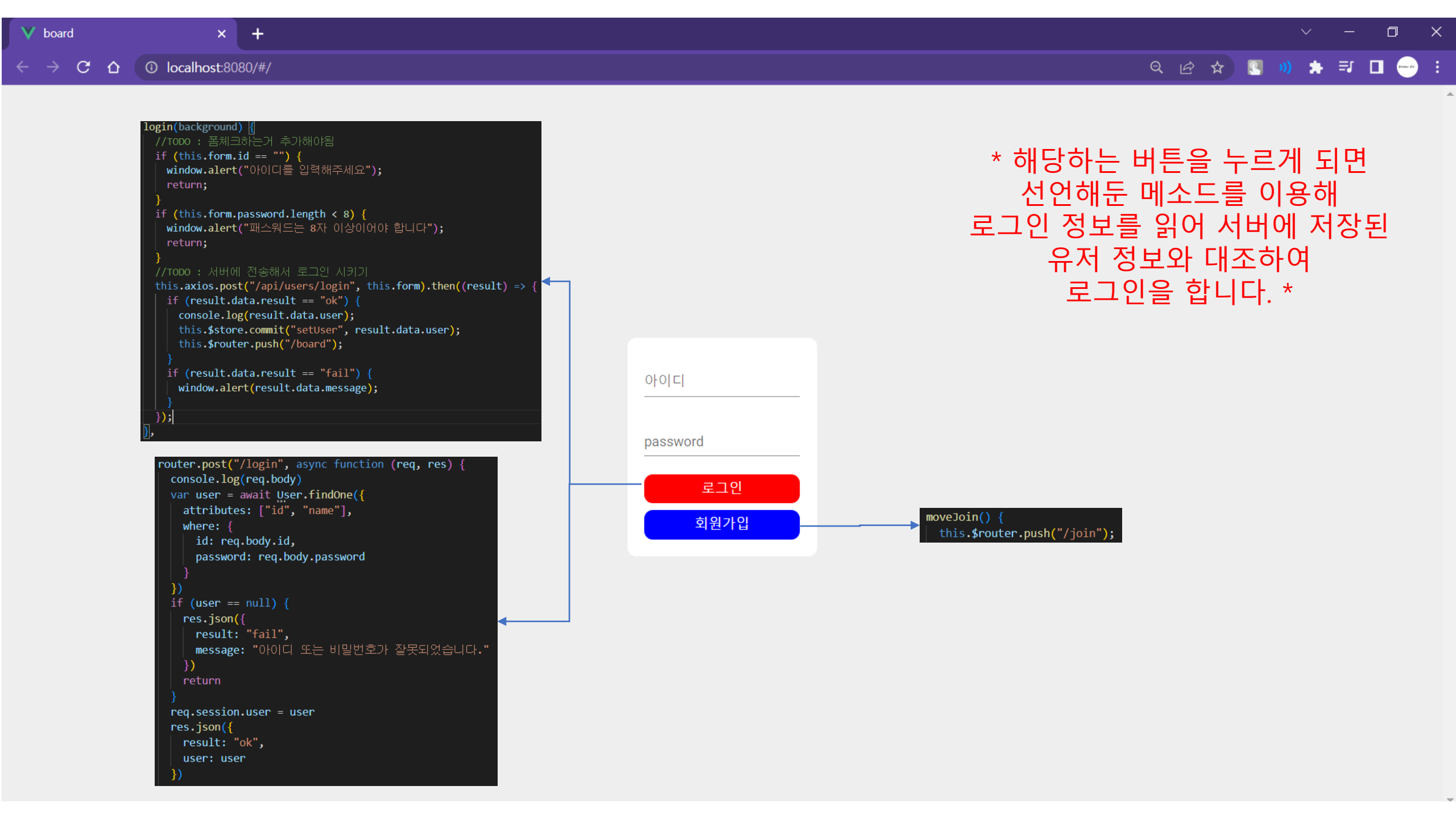
과목명	스마트 로봇 프로그래밍
담당 교수	진용화 교수님
학과	정보통신공학과
반	전공심화과정
이름	안형진
학번	2022541009

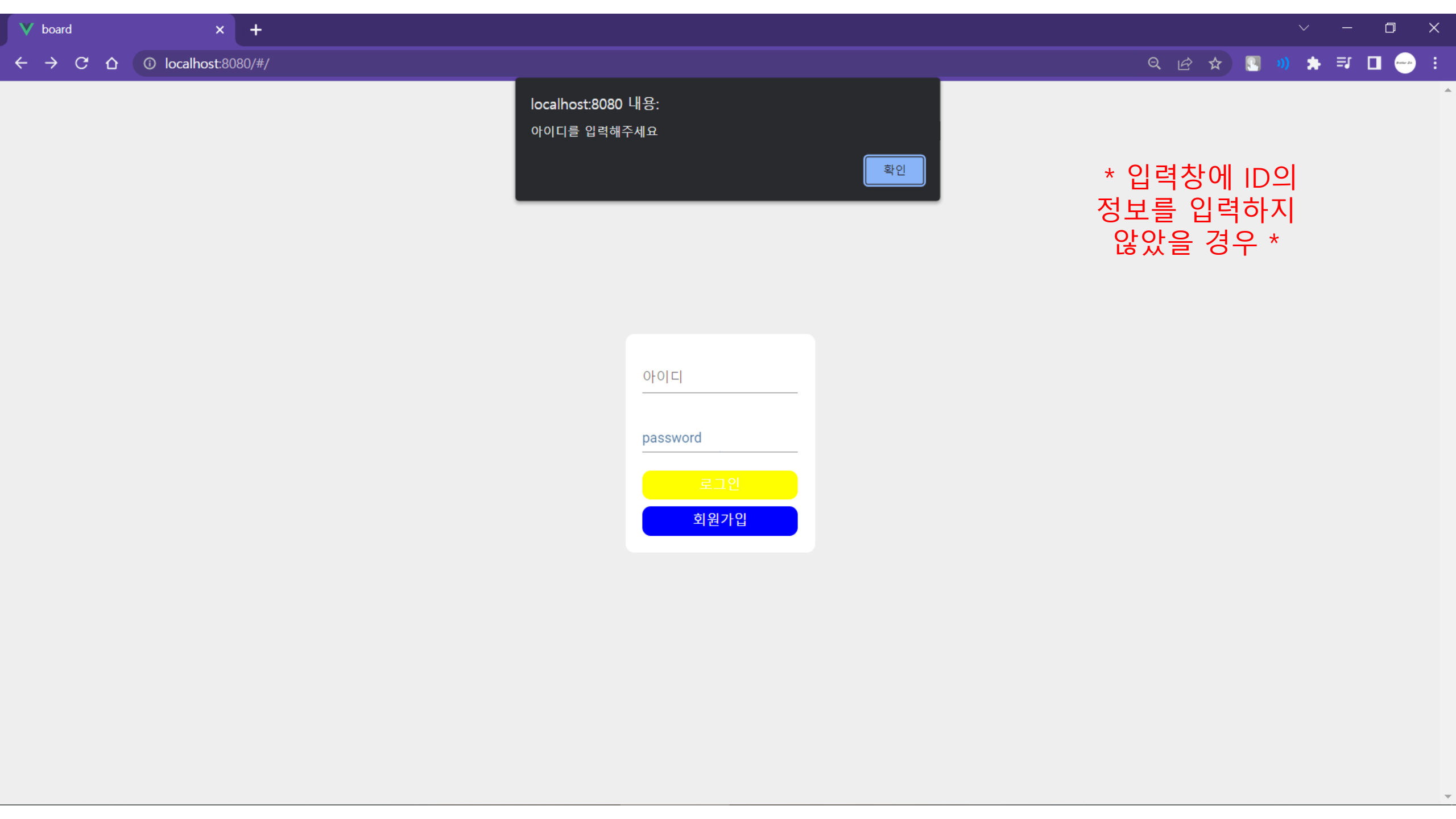
# [ 목차 ]

1. 로그인 화면
2. 회원가입 화면
3. 게시판 리스트 화면
4. 게시글 작성 화면
5. 게시글 상세 페이지 화면
6. 게시글 수정 페이지 화면

# [ 로그인 화면 ]

구분	기능	화면 / 방식	URL	설명
유저 정보	로그인 / 회원가입	POST	“/”	홈페이지를 접속 했을 때 처음으로 뜨는 화면이자 로그인 / 회원가입 버튼으로 회원 정보를 등록 가능하며, 로그인버튼 클릭시 axios.post를 통해 유저 정보를 확인 후 게시판 리스트 화면으로 넘어갈 수 있도록 한다.





localhost:8080 내용:  
아이디를 입력해주세요

확인

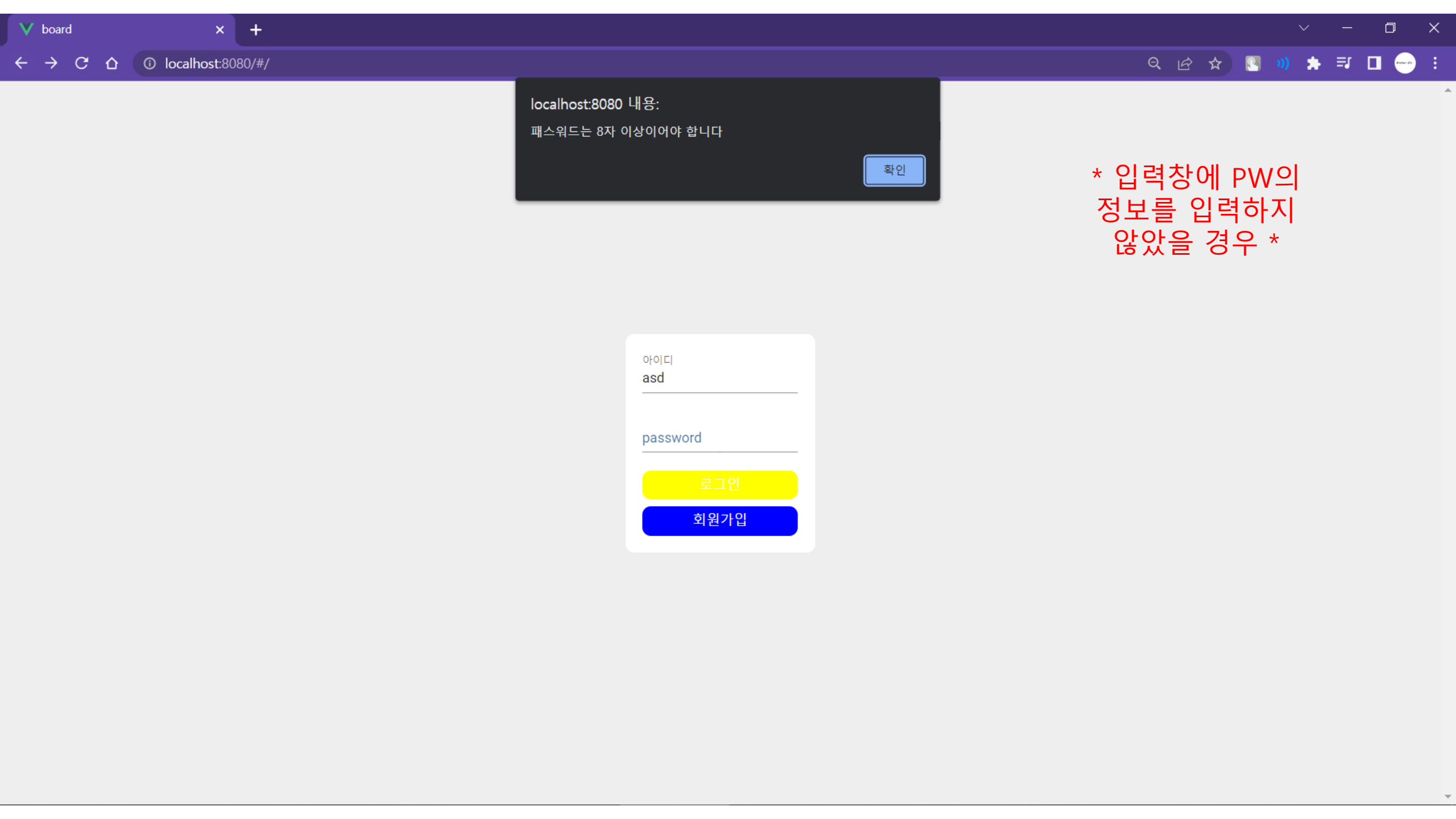
\* 입력창에 ID의  
정보를 입력하지  
않았을 경우 \*

아이디

password

로그인

회원가입



localhost:8080 내용:

패스워드는 8자 이상이어야 합니다

확인

\* 입력창에 PW의  
정보를 입력하지  
않았을 경우 \*

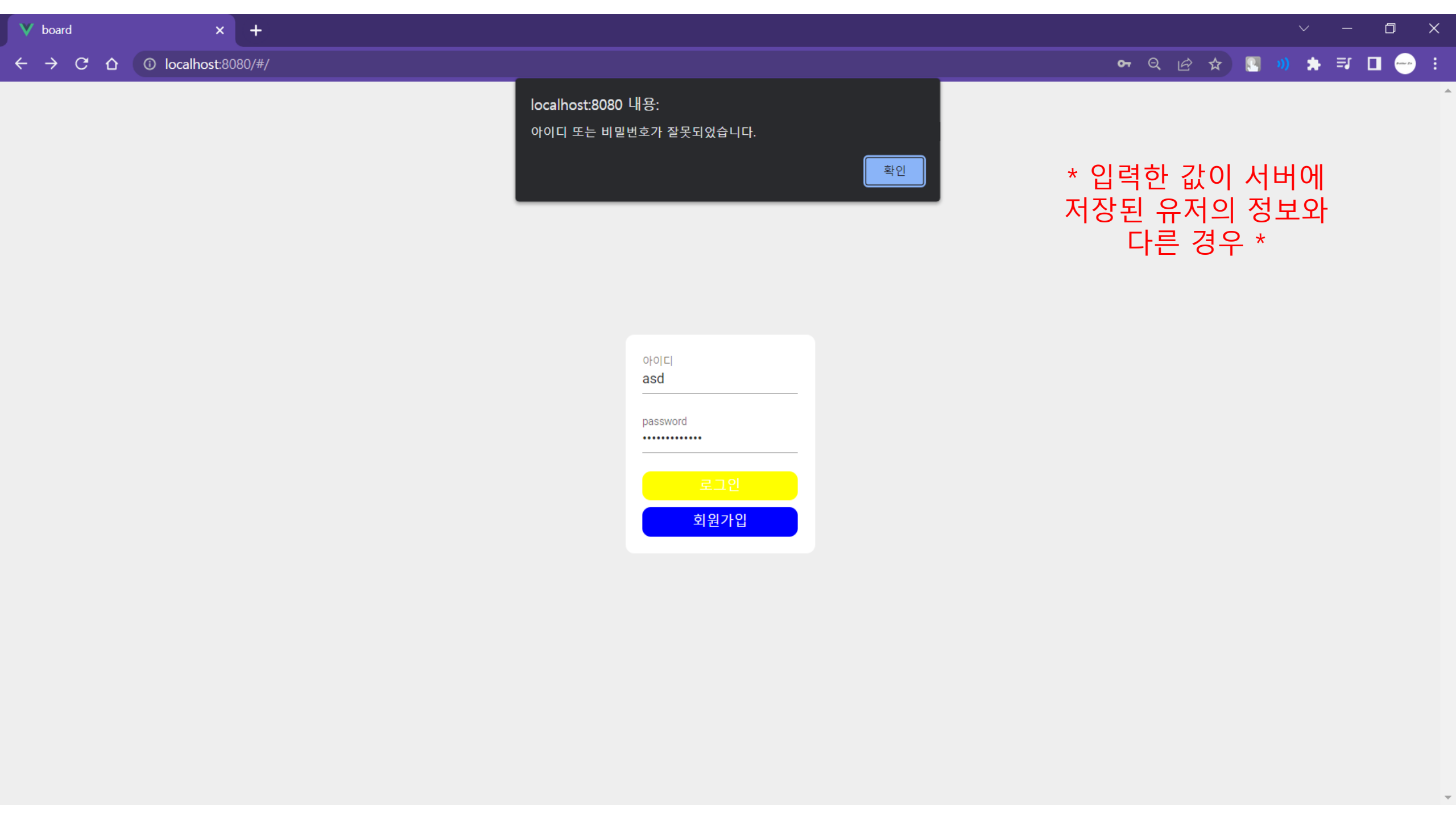
아이디

asd

password

로그인

회원가입



localhost:8080 내용:  
아이디 또는 비밀번호가 잘못되었습니다.

확인

\* 입력한 값이 서버에  
저장된 유저의 정보와  
다른 경우 \*

아이디  
asd

password  
.....

로그인

회원가입

# [ 회원가입 화면 ]

구분	기능	화면 / 방식	URL	설명
사용자 관리	회원가입	POST	“/join”	회원 가입을 위한 페이지로 사용자가 입력한 정보(이름,ID,PW)를 axios.post를 통해 DB에 저장됩니다.



board

localhost8080/#/join

```
join() {
  if (this.form.id == "") {
    window.alert("아이디를 입력해주세요");
    return;
  }
  if (this.form.name == "") {
    window.alert("이름을 입력해주세요");
    return;
  }
  if (this.form.password.length < 8) {
    window.alert("패스워드는 8자 이상이어야 합니다");
    return;
  }
  if (this.form.password != this.form.passwordConfirm) {
    window.alert("패스워드가 일치하지 않습니다");
    return;
  }

  // TODO : 서버에 전송해서 회원가입 시키기
  this.axios.post("/api/users/join", this.form).then((result) => {
    console.log(result);
    if (result.data.result == "ok") {
      //회원가입이 성공한경우
      window.alert("회원가입이 완료되었습니다");
      this.$router.push("/");
    }
    if (result.data.result == "fail") {
      //회원가입이 실패한경우
      window.alert(result.data.message);
    }
  });
},
```

아이디

이름

패스워드

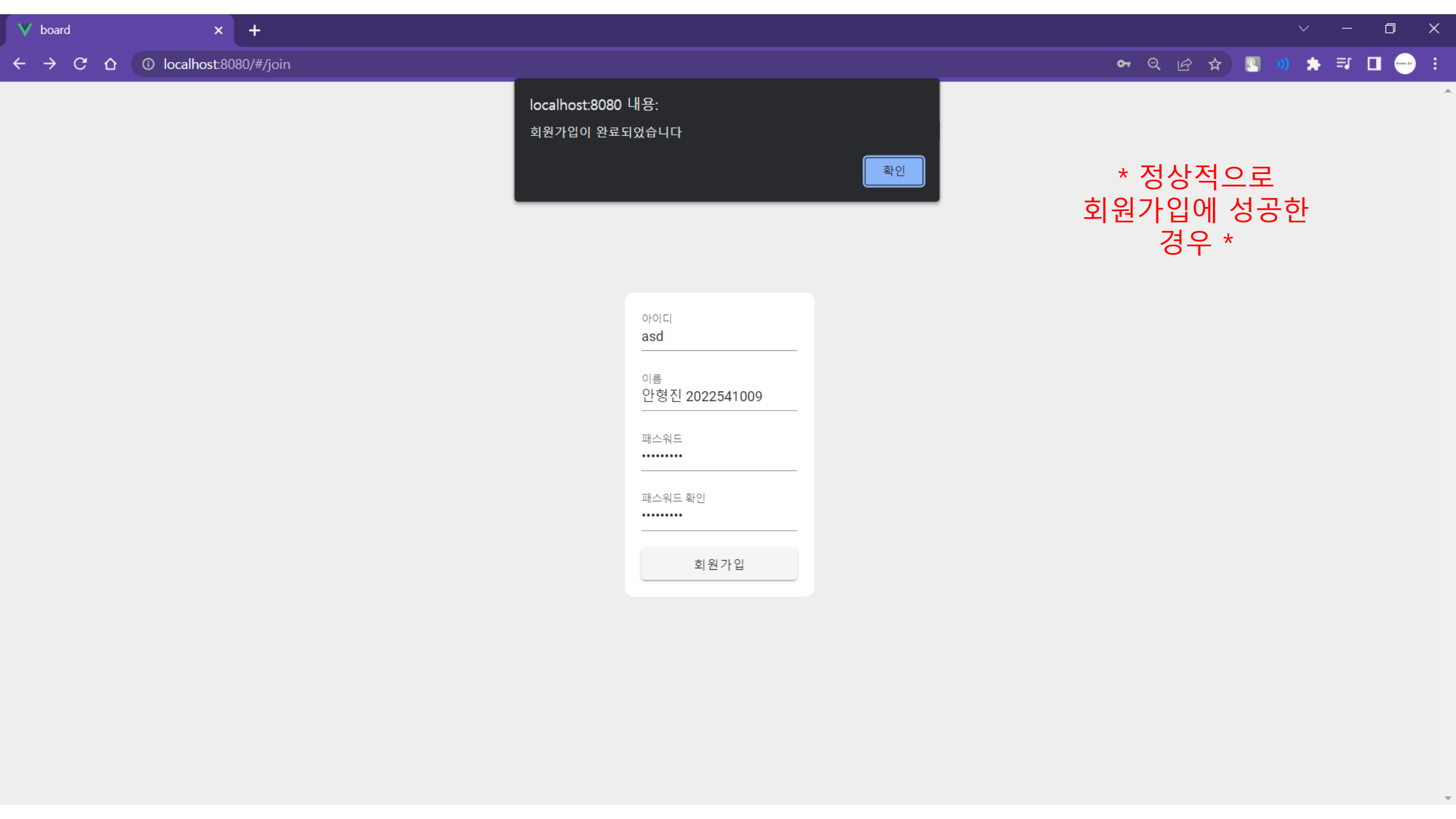
패스워드 확인

회원가입

```
router.post("/join", async function (req, res) {
  var body = req.body
  console.log(body)
  var alreadyUser = await User.findOne({
    where: {
      id: body.id
    }
  })
  if (alreadyUser != null) {
    res.json({
      result: "fail",
      message: "이미 존재하는 아이디입니다."
    })
    return
  }
  console.log(alreadyUser)

  var result = await User.create(body)
  res.json({
    result: "ok"
  })
})
```

\* 해당하는 버튼을 누르게 되면 선언해둔 메소드를 이용해 회원가입 정보를 DB에 저장합니다.\*



localhost:8080 내용:  
회원가입이 완료되었습니다

확인

\* 정상적으로  
회원가입에 성공한  
경우 \*

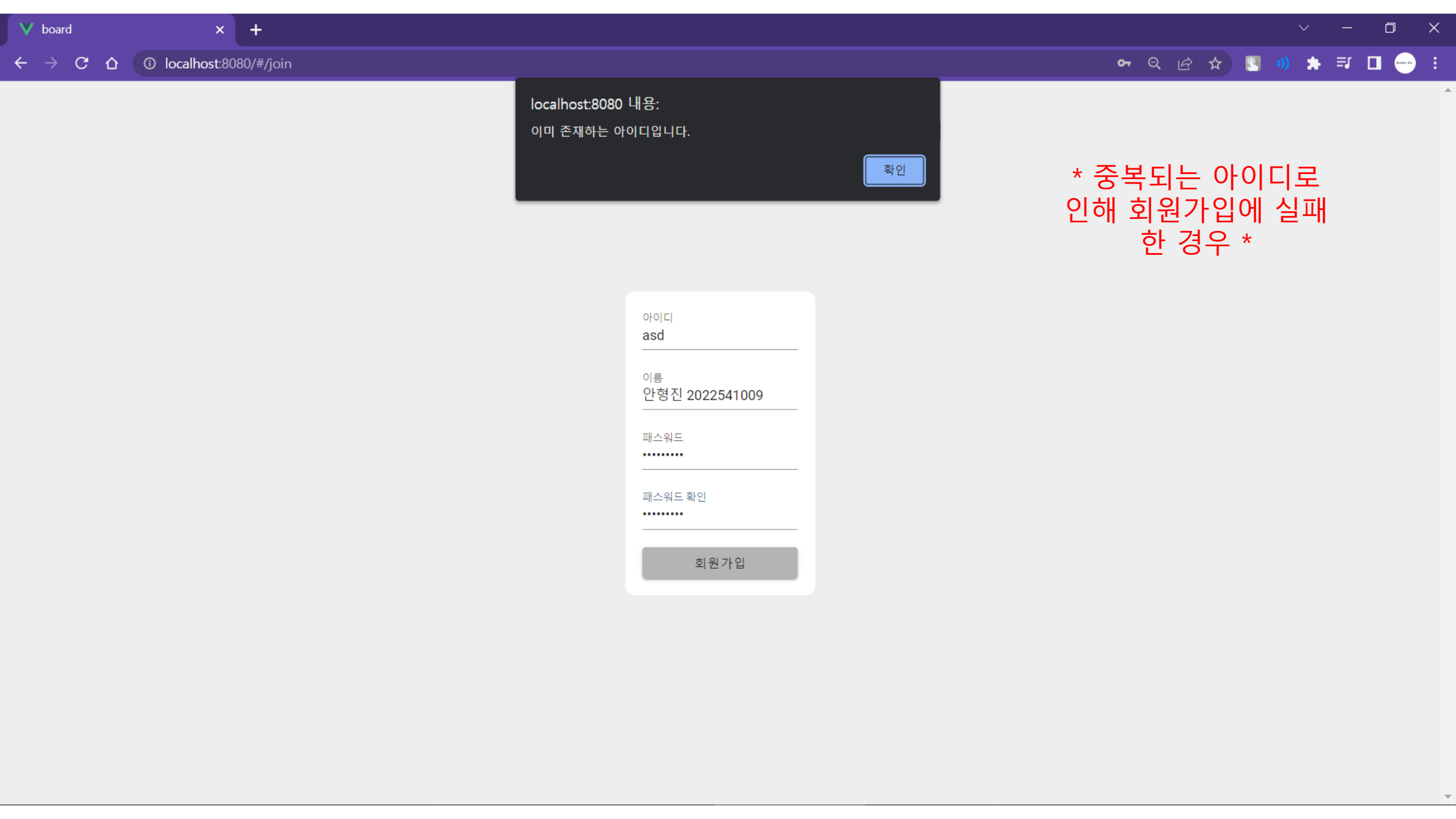
아이디  
asd

이름  
안형진 2022541009

패스워드  
.....

패스워드 확인  
.....

회원가입



localhost:8080 내용:  
이미 존재하는 아이디입니다.

확인

\* 중복되는 아이디로  
인해 회원가입에 실패  
한 경우 \*

아이디  
asd

이름  
안형진 2022541009

패스워드  
.....

패스워드 확인  
.....

회원가입

# [ 게시판 리스트 화면 ]

구분	기능	화면 / 방식	URL	설명
게시판	게시판 리스트	POST	“/board”	<p>로그인에 성공하면 출력되는 게시판 리스트 화면입니다. 게시글 작성/ 작성되어있는 게시글들을 확인 할 수 있는 페이지 입니다</p> <p>헤더부분 좌측의 Mjc Board를 누르면 게시판 리스트로 이동됩니다.</p> <p>헤더부분의 우측에는 사용자의 이름이 출력됩니다.( 로그인 되어 있는 경우에만 출력이 됩니다.)</p> <p>사용자의 이름 옆에는 로그아웃 버튼이 있습니다. (로그아웃 상태에선 게시글작성이 제한되며, 우측 상단에 사용자의 이름이 출력되지 않습니다.)</p>

board

localhost:8080/#/board

Mjc Board

로그아웃 안형진 2022541009

글번호	제목	작성자	조회수	작성일
11	test	qwe	0	2022-06-09 12:39
7	dasdasdasd		0	2022-05-28 00:45
6	1231		0	2022-05-28 00:33
5	11		0	2022-05-28 00:33
4	asdasdsad123		0	2022-05-28 00:33

< 1 2 >

changePage(page) {  
 console.log(page);  
 this.axios.post("/api/board/list", { page: page }).then((result) => {  
 this.boardList = result.data.boardList;  
 this.pageCount = result.data.pageCount;  
 });  
},

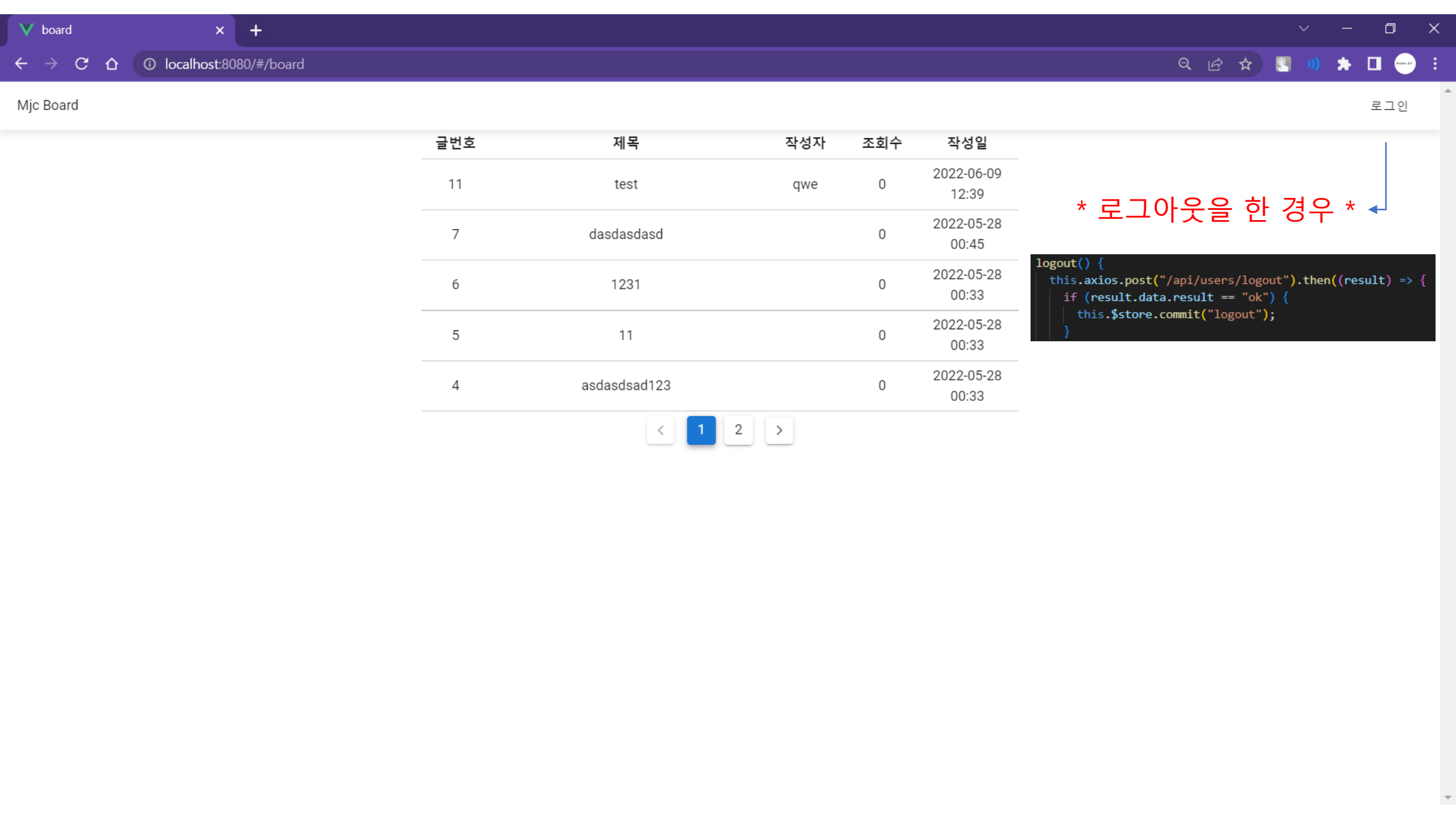
\* 한 페이지에  
게시글은 5개이며,  
5개 이상이면 다음  
페이지가 생성됩니다.\*

\* 게시판 리스트  
기본 화면  
로그인이  
되어있는 경우 \*

mounted() {  
 this.axios.post("/api/users/info").then((result) => {  
 if (result.data.result == "ok") {  
 this.\$store.commit("setUser", result.data.user);  
 }  
 });  
}

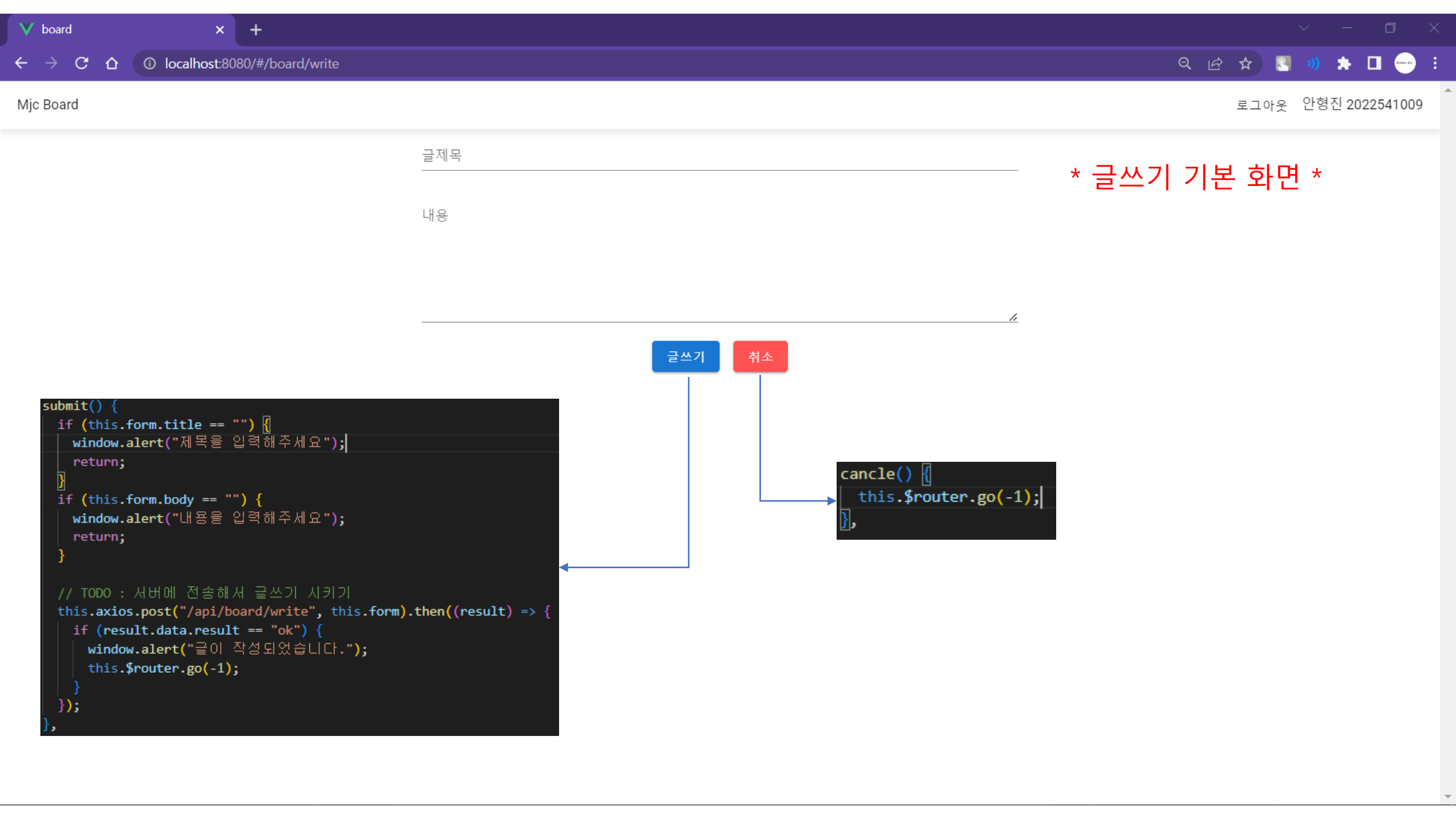
moveWrite() {  
 this.\$router.push("/board/write");  
},

글쓰기

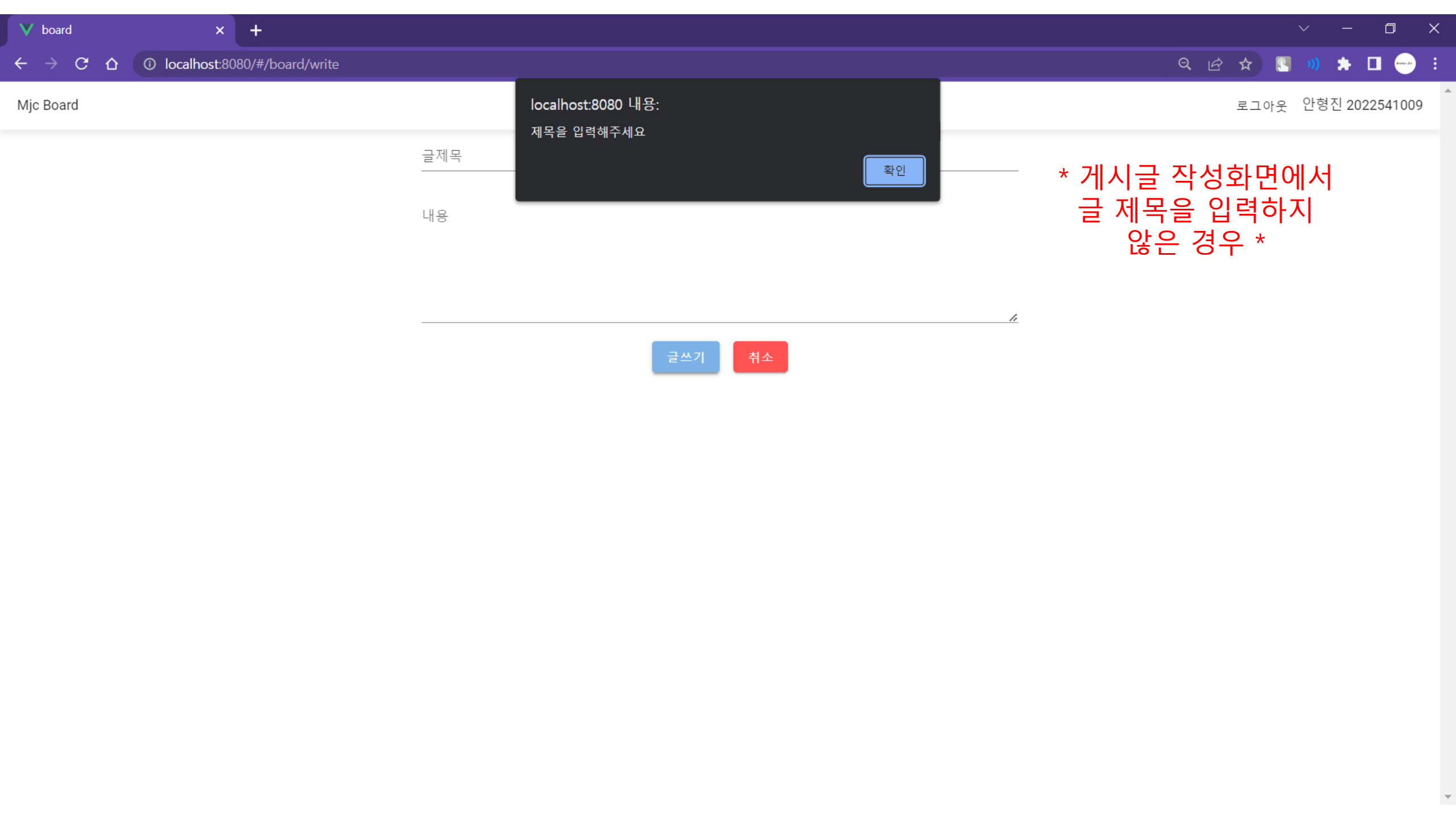


# [ 게시물 작성 화면 ]

구분	기능	화면 / 방식	URL	설명
게시판	게시글 작성	POST	“/board/write”	<p>게시판에 올리기 위한 게시글 작성 페이지 입니다.</p> <p>글 제목,내용을 입력한 후 글쓰기 버튼을 누르면 DB에 저장이 되어 게시판 리스트에서 확인이 가능합니다.</p> <p>취소 버튼을 누르면 게시판 리스트로 돌아갑니다.</p>







localhost:8080 내용:  
제목을 입력해주세요

확인

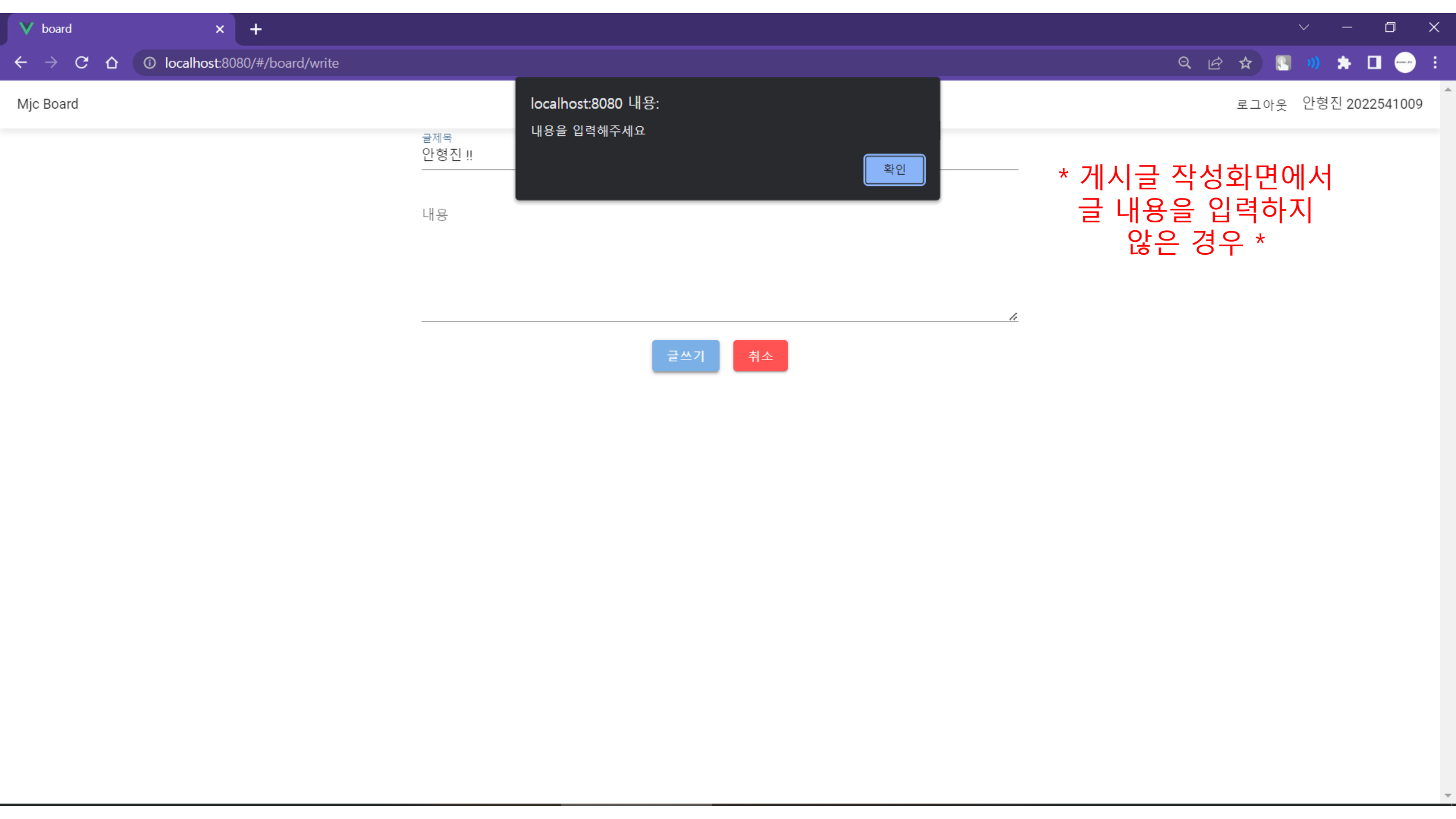
\* 게시글 작성화면에서  
글 제목을 입력하지  
않은 경우 \*

글제목

내용

글쓰기

취소



글제목  
안형진 !!

내용

localhost:8080 내용:

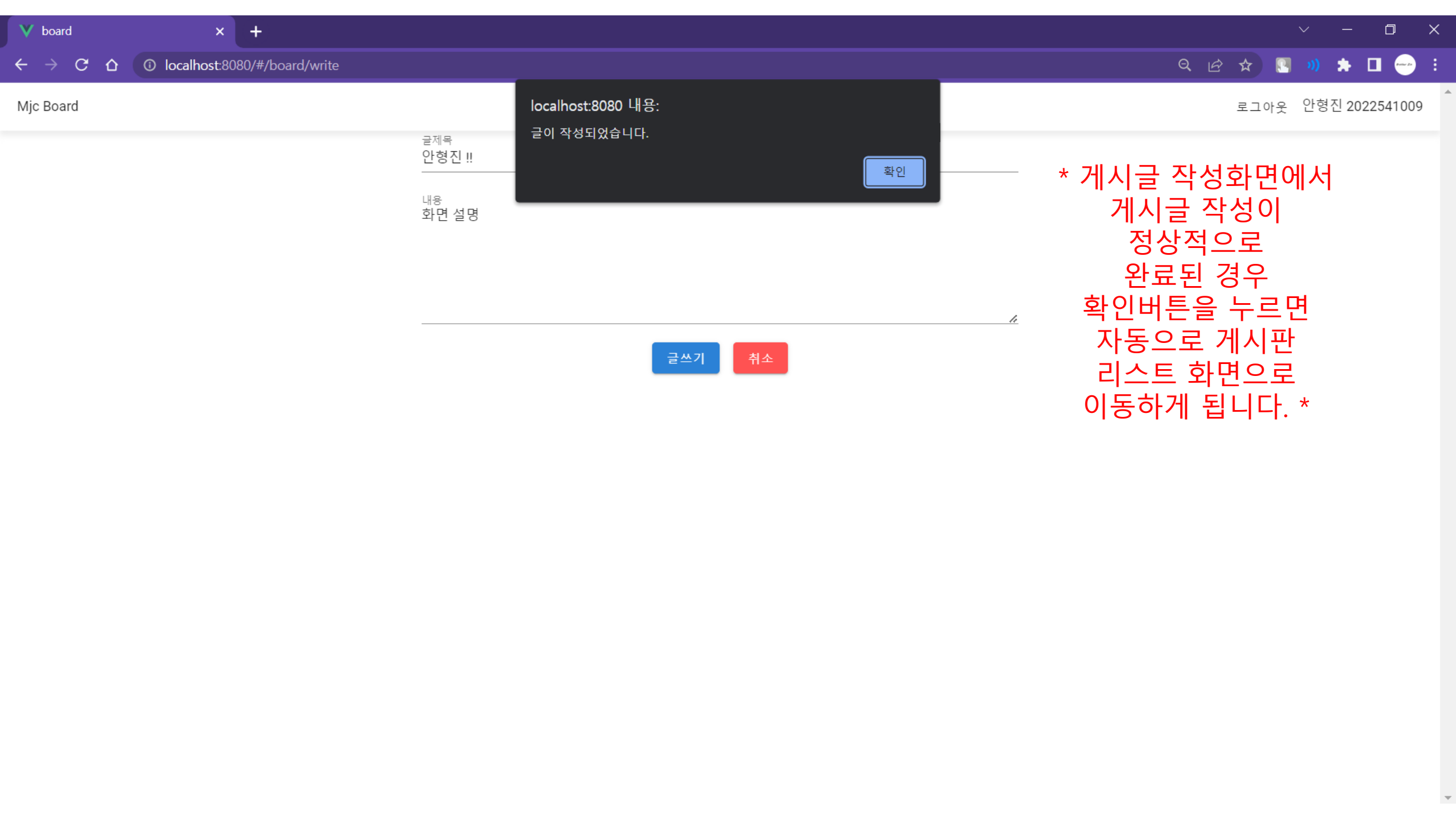
내용을 입력해주세요

확인

\* 게시글 작성화면에서  
글 내용을 입력하지  
않은 경우 \*

글쓰기

취소



localhost:8080 내용:  
글이 작성되었습니다.

확인

\* 게시글 작성화면에서  
게시글 작성이  
정상적으로  
완료된 경우  
확인버튼을 누르면  
자동으로 게시판  
리스트 화면으로  
이동하게 됩니다. \*

글쓰기

취소



글번호	제목	작성자	조회수	작성일
12	안형진 !!	안형진 2022541009	0	2022-06-11 17:22
11	test	qwe	0	2022-06-09 12:39
7	dasdasdasd		0	2022-05-28 00:45
6	1231		0	2022-05-28 00:33
5	11		0	2022-05-28 00:33

\* 작성된 게시글이  
정상적으로  
리스트에 올라온  
경우 \*

# [ 게시글 상세 페이지 화면 ]

구분	기능	화면 / 방식	URL	설명
게시판	게시글 상세 페이지 & 게시글 삭제	POST	“/board/item/: id”	<p>게시판 리스트에 올라온 게시글을 클릭하게 되면 게시글의 상세 페이지를 확인 할 수 있습니다.</p> <p>이 페이지에서 게시글의 삭제 기능을 사용할 수 있습니다.</p>

board

localhost8080/#/board/item/12

Mjc Board

로그아웃 안형진 2022541009

안형진 !!

글 제목

작성자 : 안형진 2022541009

화면 설명

글 내용

수정

삭제

```
modifyBoard() {
  this.$router.push("/board/modify/" + this.$route.params.id);
},
```

```
getBoardItem() {
  this.axios
    .post("/api/board/item", { id: this.$route.params.id })
    .then((result) => {
      this.form = result.data.board;
    });
},
```

```
removeBoard() {
  this.axios
    .post("/api/board/remove", {
      id: this.$route.params.id,
    })
    .then((result) => {
      if (result.data.result == "ok") {
        window.alert("글이 삭제되었습니다.");
        this.$router.push("/board");
      } else {
        alert(result.data.msg);
      }
    });
},
```

\* 각 게시물마다 게시물  
번호를 이용해  
상세 페이지로 이동이  
가능합니다. \*

\* 게시글의 상세 페이지  
화면 입니다. \*



글번호	제목	작성자	조회수	작성일
14	삭제 테스트중	안형진 2022541009	0	2022-06-11 18:05
11	test	qwe	0	2022-06-09 12:39
7	dasdasdasd		0	2022-05-28 00:45
6	1231		0	2022-05-28 00:33
5	11		0	2022-05-28 00:33

<

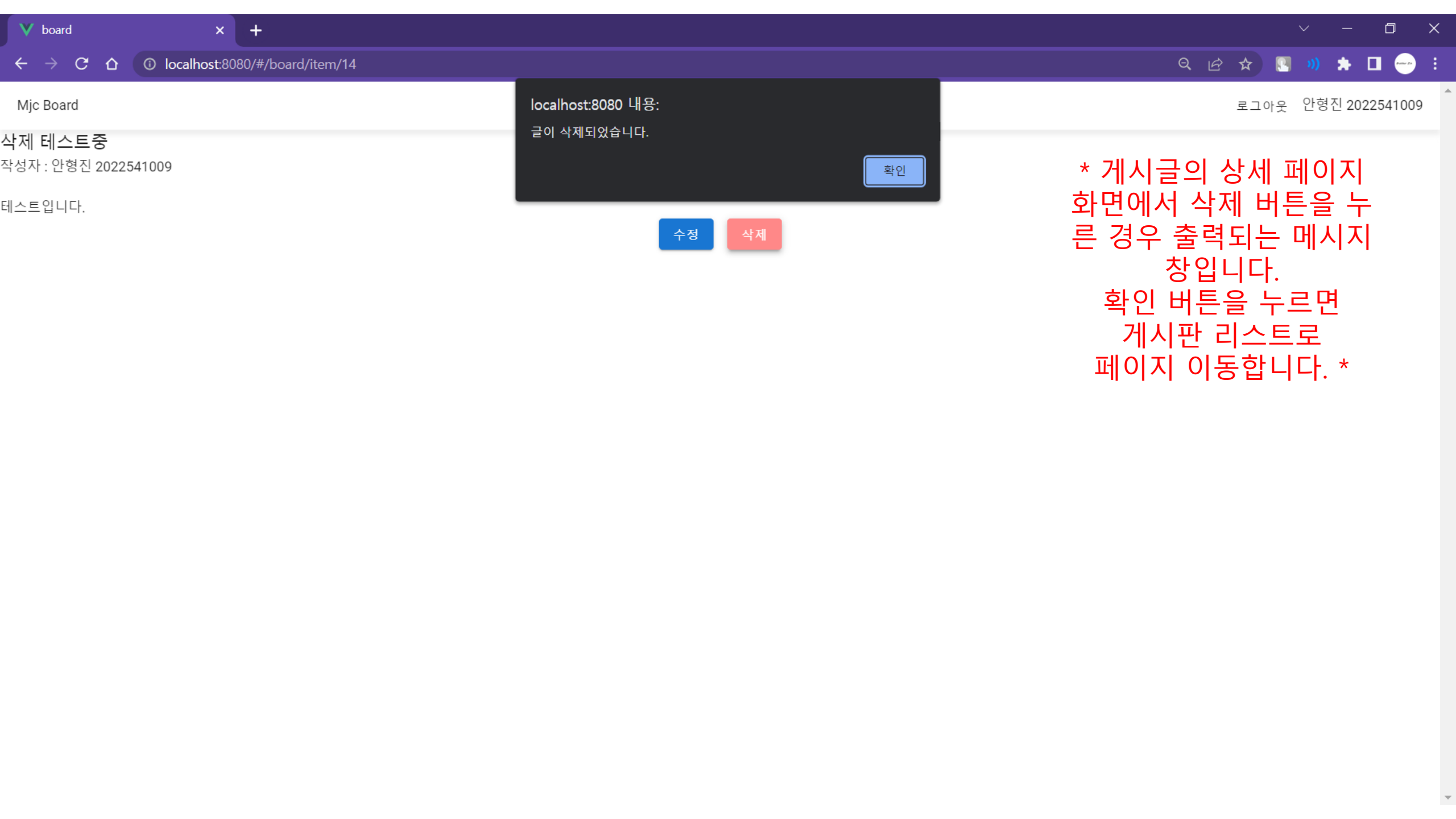
1

2

>

\* 게시물 리스트에서  
아직 삭제 하기 전  
게시글의 모습입니다.\*

글쓰기



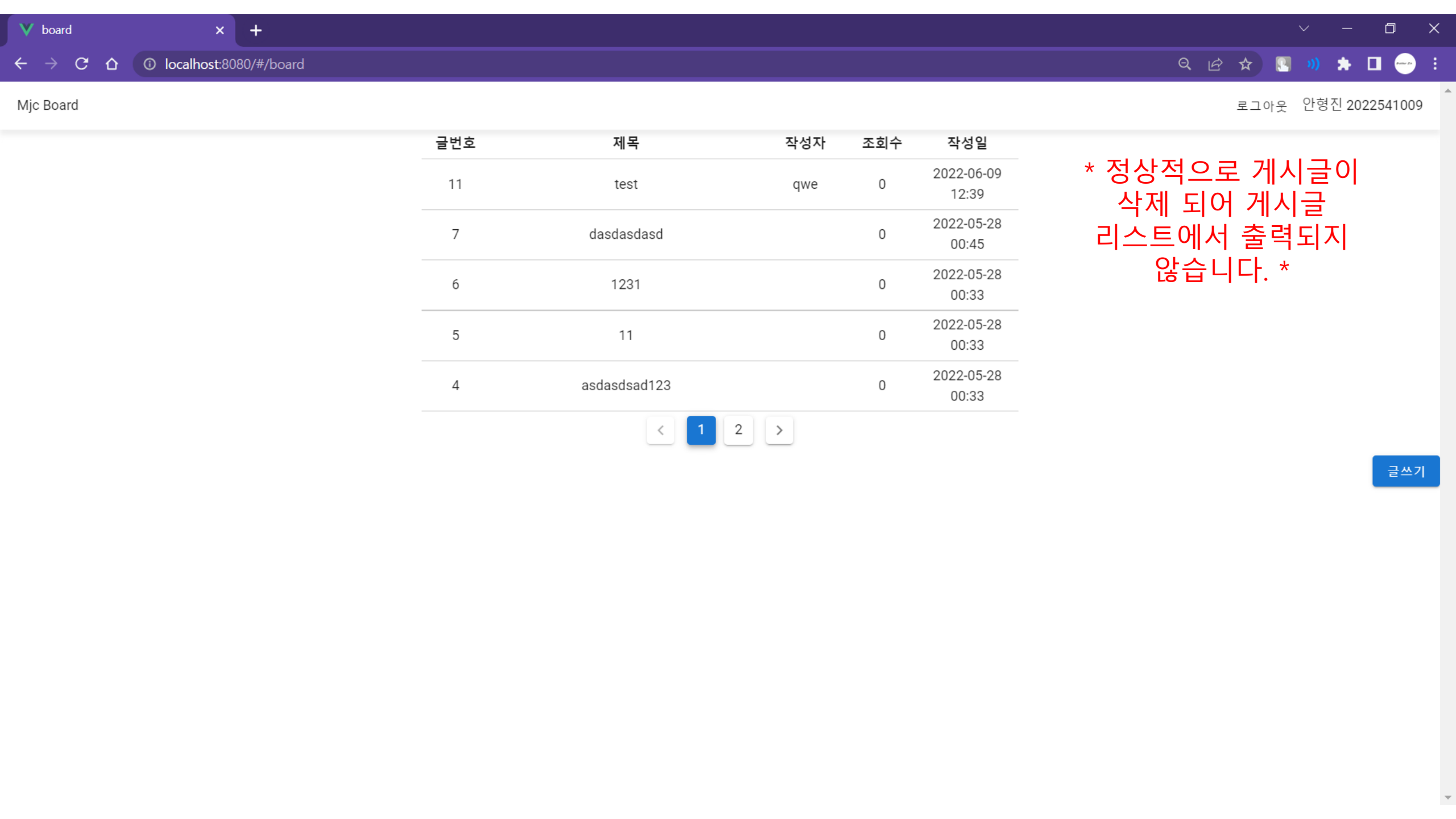
localhost:8080 내용:  
글이 삭제되었습니다.

확인

수정 삭제

\* 게시글의 상세 페이지  
화면에서 삭제 버튼을 누  
른 경우 출력되는 메시지  
창입니다.  
확인 버튼을 누르면  
게시판 리스트로  
페이지 이동합니다. \*



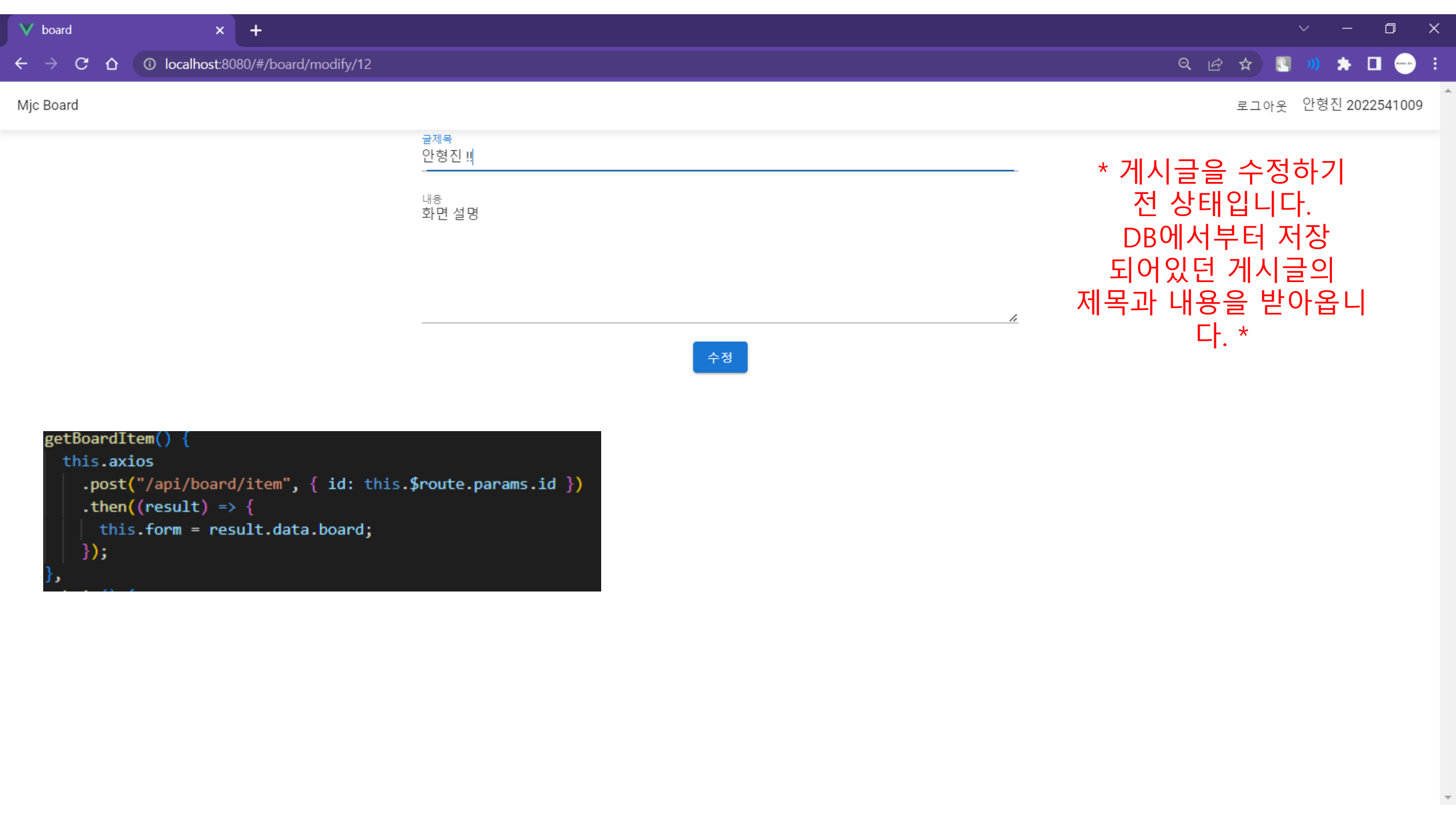


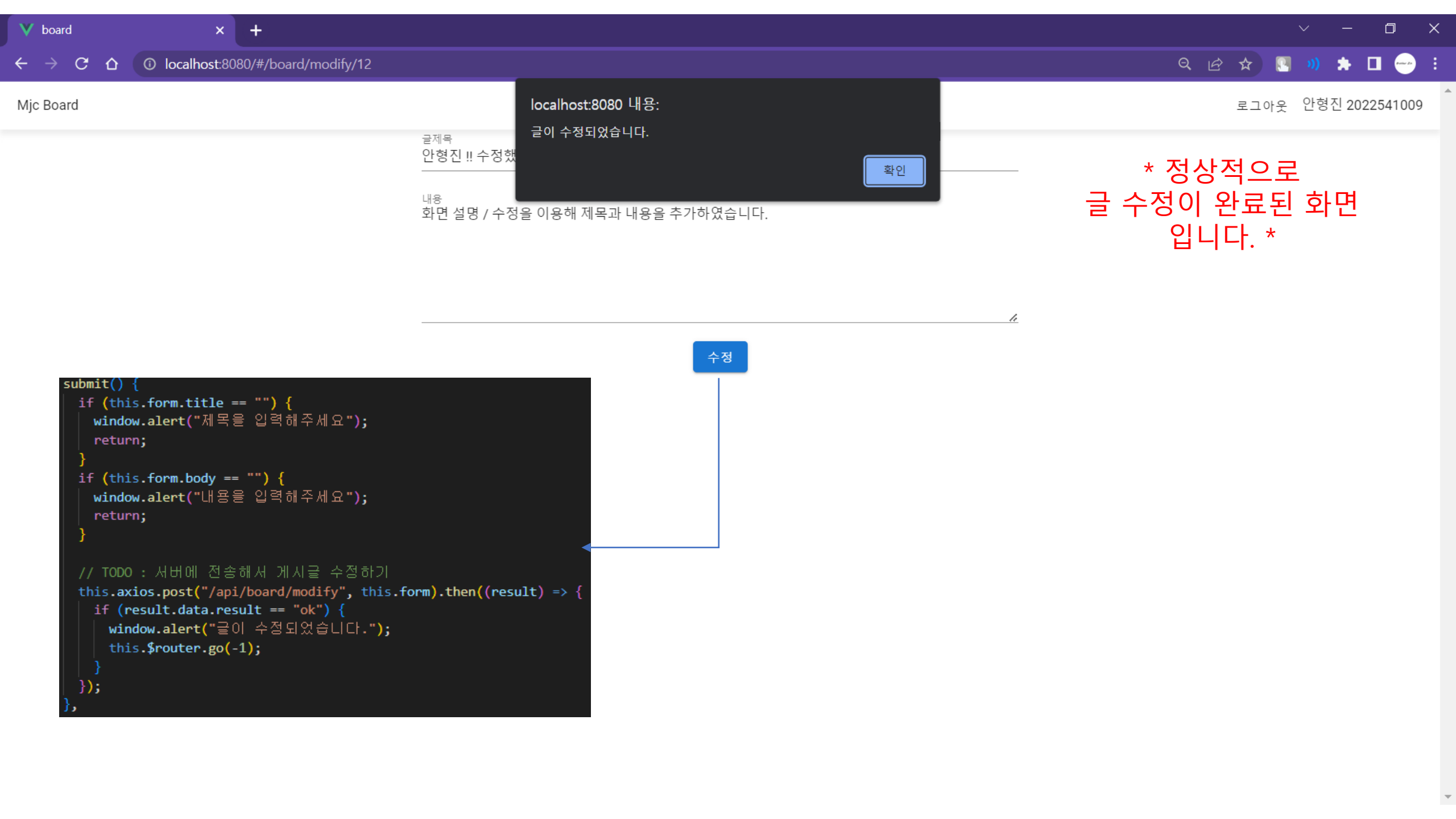
글번호	제목	작성자	조회수	작성일
11	test	qwe	0	2022-06-09 12:39
7	dasdasdasd		0	2022-05-28 00:45
6	1231		0	2022-05-28 00:33
5	11		0	2022-05-28 00:33
4	asdasdsad123		0	2022-05-28 00:33

\* 정상적으로 게시글이 삭제 되어 게시글 리스트에서 출력되지 않습니다. \*

# [ 게시글 수정 페이지 화면 ]

구분	기능	화면 / 방식	URL	설명
게시판	게시글 수정 페이지	POST	“/board/modify/: id”	게시글의 상세 페이지에서 수정하기 버튼을 누르면 수정 페이지로 이동하게 됩니다. 수정 페이지에서는 게시글의 제목, 내용을 수정할 수 있습니다.





글제목  
안형진 !! 수정했

내용  
화면 설명 / 수정을 이용해 제목과 내용을 추가하였습니다.

localhost:8080 내용:  
글이 수정되었습니다.

확인

\* 정상적으로  
글 수정이 완료된 화면  
입니다. \*

수정

```
submit() {  
  if (this.form.title == "") {  
    window.alert("제목을 입력해 주세요");  
    return;  
  }  
  if (this.form.body == "") {  
    window.alert("내용을 입력해 주세요");  
    return;  
  }  
  
  // TODO : 서버에 전송해서 게시글 수정하기  
  this.axios.post("/api/board/modify", this.form).then((result) => {  
    if (result.data.result == "ok") {  
      window.alert("글이 수정되었습니다.");  
      this.$router.go(-1);  
    }  
  });  
},
```



글번호	제목	작성자	조회수	작성일
12	안형진 !! 수정했습니다.	안형진 2022541009	0	2022-06-11 17:22
11	test	qwe	0	2022-06-09 12:39
7	dasdasdasd		0	2022-05-28 00:45
6	1231		0	2022-05-28 00:33
5	11		0	2022-05-28 00:33

<

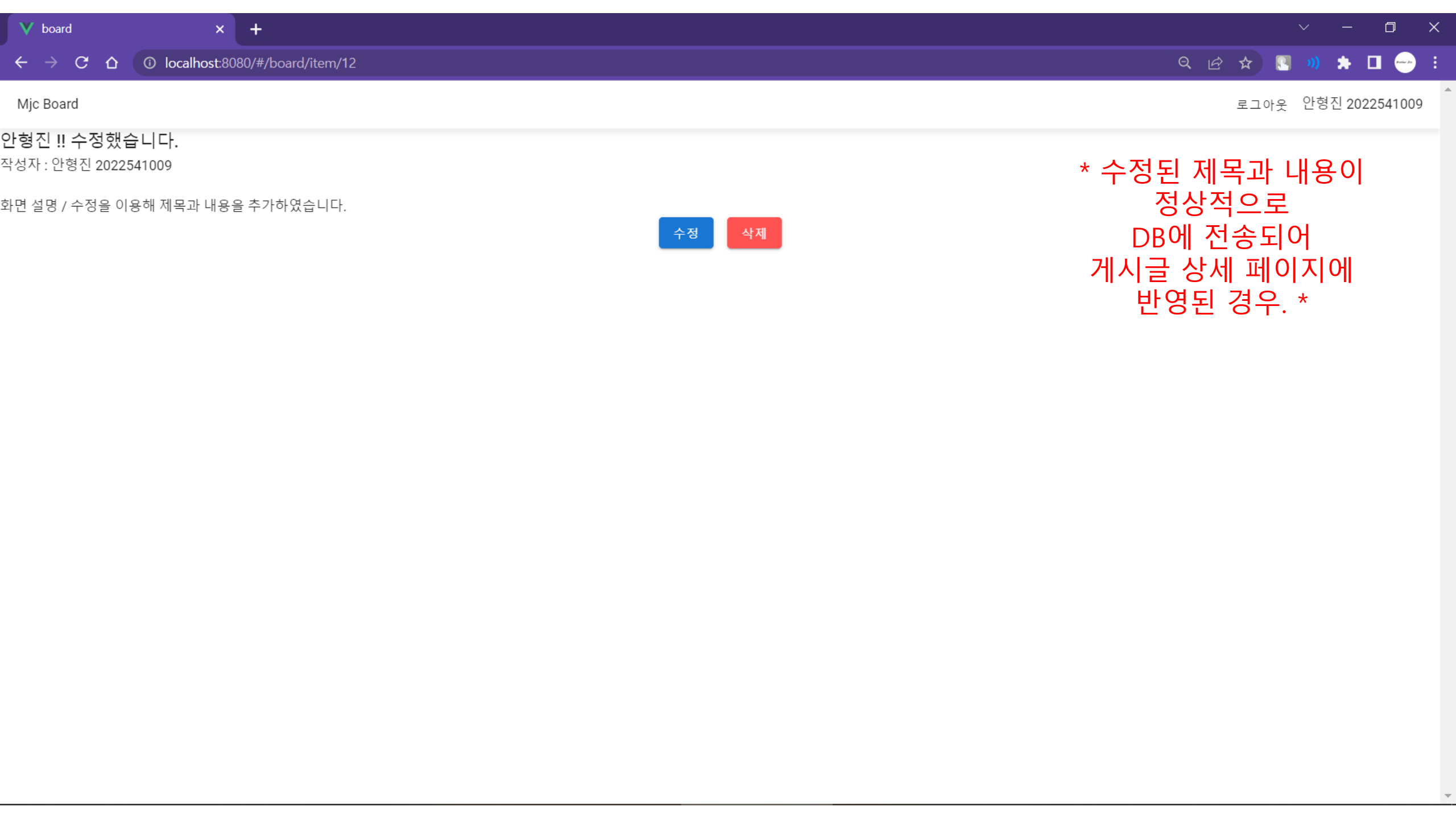
1

2

>

\* 수정된 제목과 내용이  
정상적으로  
DB에 전송되어  
게시글 리스트에  
반영된 경우. \*

글쓰기



안형진 !! 수정했습니다.

작성자 : 안형진 2022541009

화면 설명 / 수정을 이용해 제목과 내용을 추가하였습니다.

수정 삭제

\* 수정된 제목과 내용이  
정상적으로  
DB에 전송되어  
게시글 상세 페이지에  
반영된 경우. \*