

심민규 goorm 사이트

<https://nodebird-kruta.run.goorm.site>

<팔로잉 끊기 기능>

1. 팔로잉한 사용자에게 한해 팔로잉 끊기 버튼 만들기

해당 코드는 main.html 파일에서 팔로잉한 사용자에게는 "팔로우하기" 버튼을, 팔로잉하지 않은 사용자에게는 "팔로잉끊기" 버튼을 보여줍니다.

```
{% if not followingIdList.includes(twit.User.id) and twit.User.id !== user.id %}
    <button class="twit-follow">팔로우하기</button>
{% else %}
    <button class="twit-notfollow">팔로잉끊기</button>
{% endif %}
<div class="twit-content">{{ twit.content }}</div>
{% if twit.img %}
```

2. 버튼 이벤트 코드 작성하기

해당 코드는 main.html 파일에서 "팔로잉끊기" 버튼을 클릭했을 때, 팔로잉을 끊을 것인지를 묻는 확인 대화상자가 나타나고, 확인을 선택하면 백엔드로 팔로잉 끊기 요청을 보내고 페이지를 새로고침합니다.

```
document.querySelectorAll('.twit-notfollow').forEach(function(tag) {
    tag.addEventListener('click', function() {
        const myId = document.querySelector('#my-id');
        if (myId) { //로그인된 사용자 존재하면
            const userId = tag.parentNode.getAttribute('data-user-id');
            if (myId !== userId) { //현재 사용자와 팔로잉끊기한 사용자가
                if (confirm('팔로잉을 끊으시겠습니까?')) { //팔로잉끊기 여부
                    axios.post(`/user/${userId}/notfollow`)
                        .then(() => {
                            location.reload();
                        })
                        .catch((error) => {
                            console.error(error);
                        });
                }
            }
        }
    });
});
```

```

    }
  }
}
});
});

```

3. routes/user.js에 notfollow 코드 작성하기

해당 코드는 routes/user.js 파일에서 "/:id/notfollow" 경로로 POST 요청이 들어왔을 때, 팔로잉을 끊는 기능을 수행합니다. 해당하는 사용자를 데이터베이스에서 찾아 팔로잉 관계를 제거하고 "success"를 응답합니다.

/ 팔로잉 끊기 기능 */*

```

router.post('/:id/notfollow', isLoggedIn, async(req, res, next) => {
  try{
    const user = await User.findOne({ where: {id: req.params.id } });
    if (user){ //데이터베이스에서 찾은 사용자가 있다면
      await User.removeFollower(parseInt(req.user.id)); //팔로잉 끊기
      res.send('success');
    }
  } catch (error){
    next(error);
  }
});

```

<프로필 정보 변경 기능>

1. 프로필 정보 변경

해당 코드는 layout.html 파일에서 프로필 수정 버튼을 추가합니다.

```
<button class="change-nick">닉네임 수정</button>
```

2. 버튼 이벤트 코드 작성하기

해당 코드는 프로필 수정 버튼을 클릭했을 때, 닉네임을 입력받고 입력값이 있을 경우 "/user/:nick" 경로 요청을 보냅니다.

```

// POST /user/nickname
router.post('/nickname', isLoggedIn, async (req, res, next) => {
  try {
    if (!req.user) {
      // If user is not authenticated, return an error response
      return res.status(403).json({ error: 'Unauthorized' });
    }
    const { nick } = req.body;

```

```
req.user.nick = nick;
await req.user.save();
res.redirect('/profile');
} catch (error) {
next(error);
}
});
```

<좋아요 기능>

1. 이벤트 리스너 버튼 생성

해당 코드는 **main.html** 파일에서 "좋아요" 버튼을 생성합니다.

```
{% endif %}
<button class="twit-like btn btn-primary">좋아요</button>
document.querySelectorAll('.twit-like').forEach(function(tag) { //
좋아요를 누르면 좋아요
tag.addEventListener('click', function() {
const twitId = tag.parentNode.querySelector('.twit-id').value;
axios.post('/post/${twitId}/like')
.then(() => {
location.reload();
})
.catch((err) => {
console.error(err);
});
});
});
```

<게시글 삭제 기능>

1. 이벤트 리스너 버튼 생성

해당 코드는 **main.html** 파일에서 "삭제" 버튼을 생성합니다. 해당 포스트의 작성자와 현재 사용자가 같을 경우에만 삭제 버튼을 보여줍니다.

```
{% if twit.User.id == user.id %}
<button class="twit-delete btn btn-outline-danger">삭제</button>
{% endif %}
(routes user)
// POST /user/profile
```

```
router.post('/profile', isLoggedIn, upload.single('avatar'), async
(req, res, next) => {
  try {
    const { nick, bio } = req.body;
    const { user } = req; // Retrieve the authenticated user from the
    request object
    // Update the user's nickname and bio
    user.nick = nick;
    user.bio = bio;
    if (req.file) {
      user.avatar = '/uploads/${req.file.filename}'; // Set the avatar path
      if a file is uploaded
    }
    await user.save(); // Save the changes to the user
    res.redirect('/profile');
  } catch (error) {
    next(error);
  }
});
```

사용자 이름 누르면 해당 사용자 게시글만 보여주는 기능은은 프론트만 실행됨

layout.html