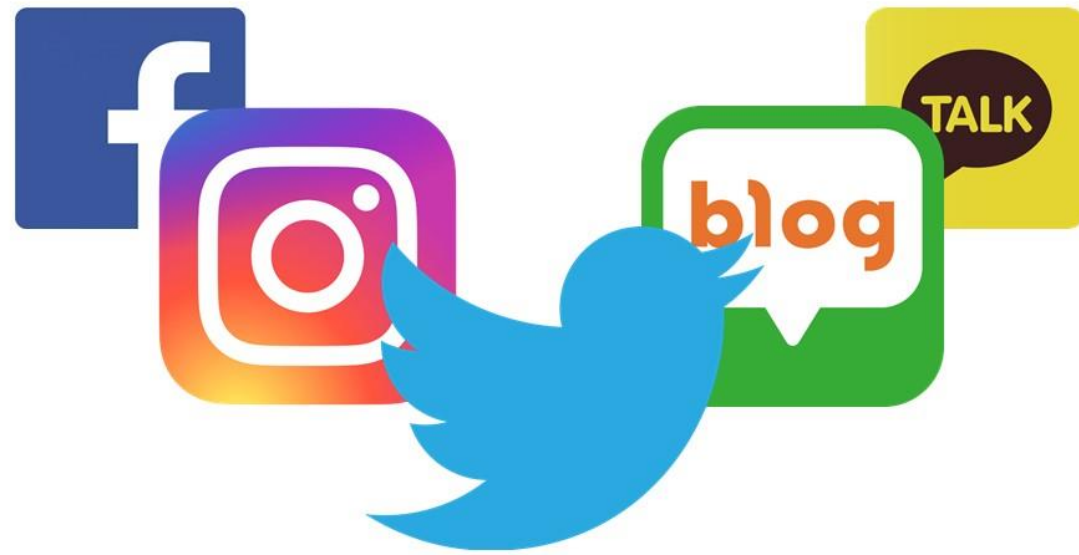


Node SNS 미니 프로젝트

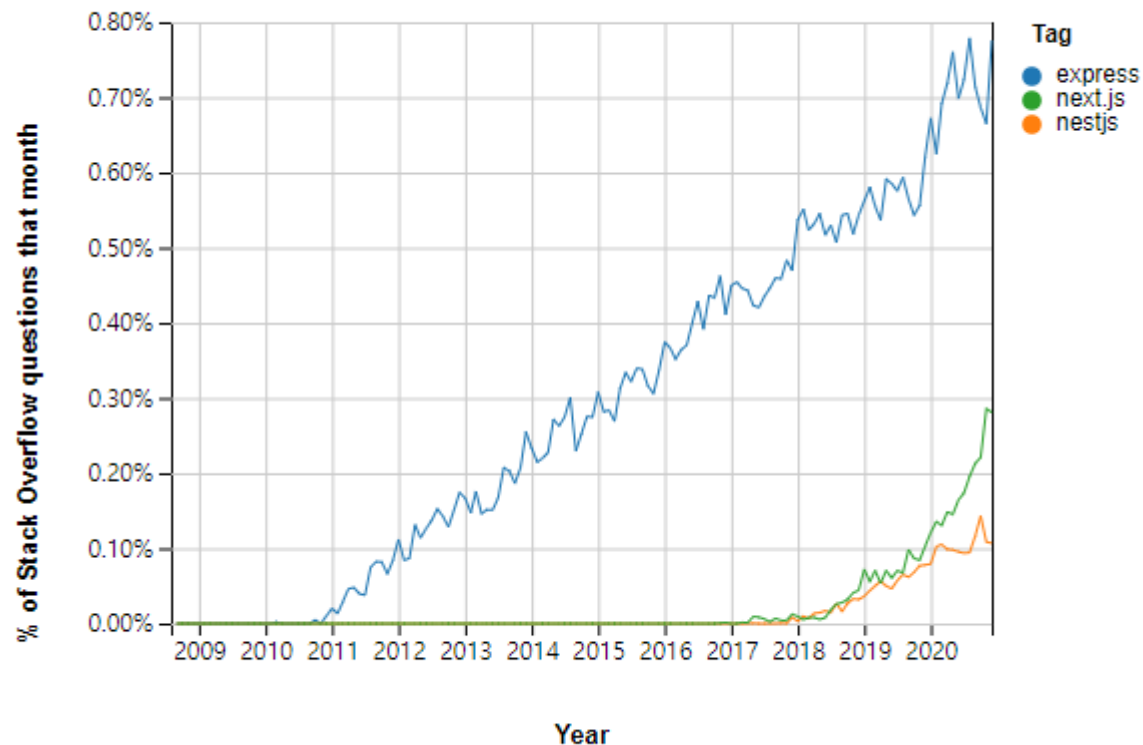


입사 희망자 이경수

목차

1. 프로젝트 선정 배경
2. 프로젝트 구성
3. 프로젝트 핵심구조
4. 프로젝트 수행 경과
5. 자체 평가 의견

프로젝트 주제 선정 배경



- ▶ Node js의 핵심 개념인 ‘논 블로킹, 이벤트 기반 모델’을 실전에 사용하고자, 실시간 입출력이 잦은 SNS를 프로젝트 주제로 선정
- ▶ 개발 생산성을 높이고, 생태계가 잘 조성된 Express 프레임워크 채택

프로젝트 테이블 - 회원

프로젝트 명		NodeSNS						
테이블 명		Users				용도	회원 관리	
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1	O		O		X	Id	int	회원 번호
2		O			O	Email	varchar(40)	계정
3					X	Nick	varchar(15)	이름
4					O	Password	varchar(100)	암호
5					X	Provider	enum(String)	계정 유형
6					O	SnsId	varchar(30)	SNS 식별 번호
7					X	CreatedAt	datetime	생성
8					X	UpdatedAt	datetime	수정
9					O	DeletedAt	datetime	삭제

프로젝트 테이블 - 게시물

프로젝트 명		NodeSNS						
테이블 명		Posts		용도		게시물 관리		
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1	O		O		X	id	int	게시물 번호
2					X	content	varchar(140)	게시물 내용
3					O	img	varchar(200)	이미지 경로
4					X	CreatedAt	datetime	생성
5					X	UpdatedAt	datetime	수정
6				O	O	UserId	int	유저 번호

프로젝트 테이블 - 게시물 & 태그

프로젝트 명		NodeSNS						
테이블 명		PostHashTags				용도	게시물 & 태그 관리	
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1				O	X	HashtagId	int	태그 번호
2				O	X	PostId	int	게시물 번호
3					X	CreatedAt	datetime	생성
4					X	UpdatedAt	datetime	수정

프로젝트 명		NodeSNS						
테이블 명		HashTags				용도	태그 관리	
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1	O		O		X	id	int	태그 번호
2		O			X	title	varchar(15)	태그 이름
3					X	CreatedAt	datetime	생성
4					X	UpdatedAt	datetime	수정

프로젝트 테이블 - 게시물 & 댓글

프로젝트 명		NodeSNS						
테이블 명		PostComment				용도	게시물 & 댓글 관리	
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1				O	X	CommentId	int	댓글 번호
2				O	X	PostId	int	게시물 번호
3					X	CreatedAt	datetime	생성
4					X	UpdatedAt	datetime	수정

프로젝트 명		NodeSNS						
테이블 명		Comments				용도	댓글 관리	
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1	O		O		X	id	int	댓글 번호
2					X	content	varchar(150)	댓글 내용
3					X	CreatedAt	datetime	생성
4					X	UpdatedAt	datetime	수정
5				O	O	UserId	int	회원 번호

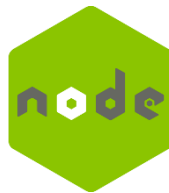
프로젝트 테이블 - 팔로우

프로젝트 명		NodeSNS						
테이블 명		Follow		용도		팔로우 관리		
NUM	PK	UQ	AI	FK	NULL	Column Name	Data Type	Comment
1	O				X	followingId	int	팔로잉 번호
2	O				X	followerId	int	팔로워 번호
3					X	CreatedAt	datetime	생성
4					X	UpdatedAt	datetime	수정

개발 환경



VSCode



Node.JS



MySQL

AXIOS



Sequelize



Passport



Bcrypt



NodeMon



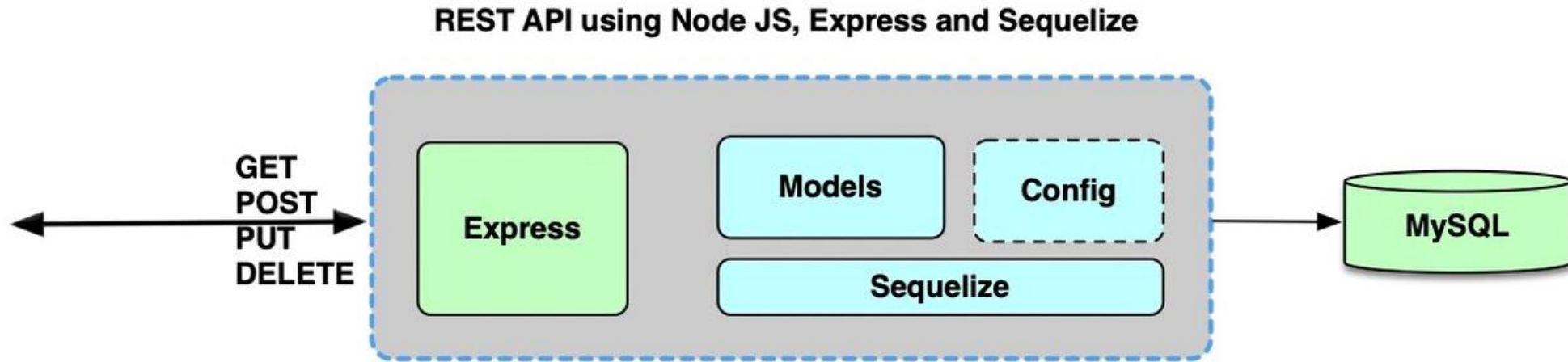
DotEnv



HTML, CSS, JavaScript

express

프로젝트 구조



사용자의 Request요청에 따른 Express내 Router가 필요에 따라 컨트롤러와 연결된 DB를 거쳐 정보를 템플릿에 가공하여 Response 합니다. 서버로부터 응답 받은 사용자의 브라우저는 해당 템플릿을 렌더링 하게 됩니다.

기능 구조도

이메일

비밀번호

회원 가입

카카오로그인

로그인

사진 업로드

게시글 등록

태그 검색

검색

admin

팔로우

node

#NodeJS

OPEN

작성자 : 3 | 댓글 1번

작성일: 2024-01-26T07:40:21.000Z

댓글 삭제

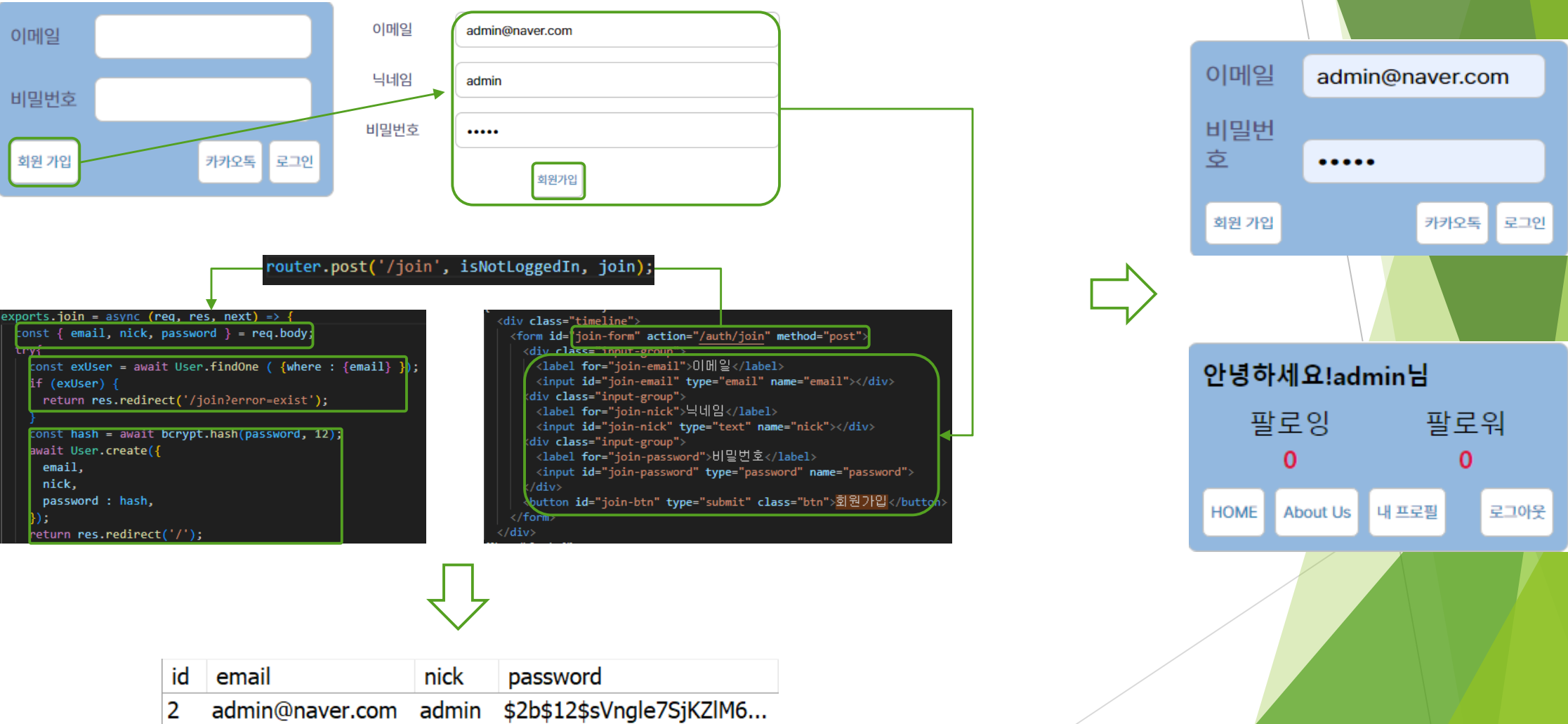
댓글을 작성하세요

댓글 작성

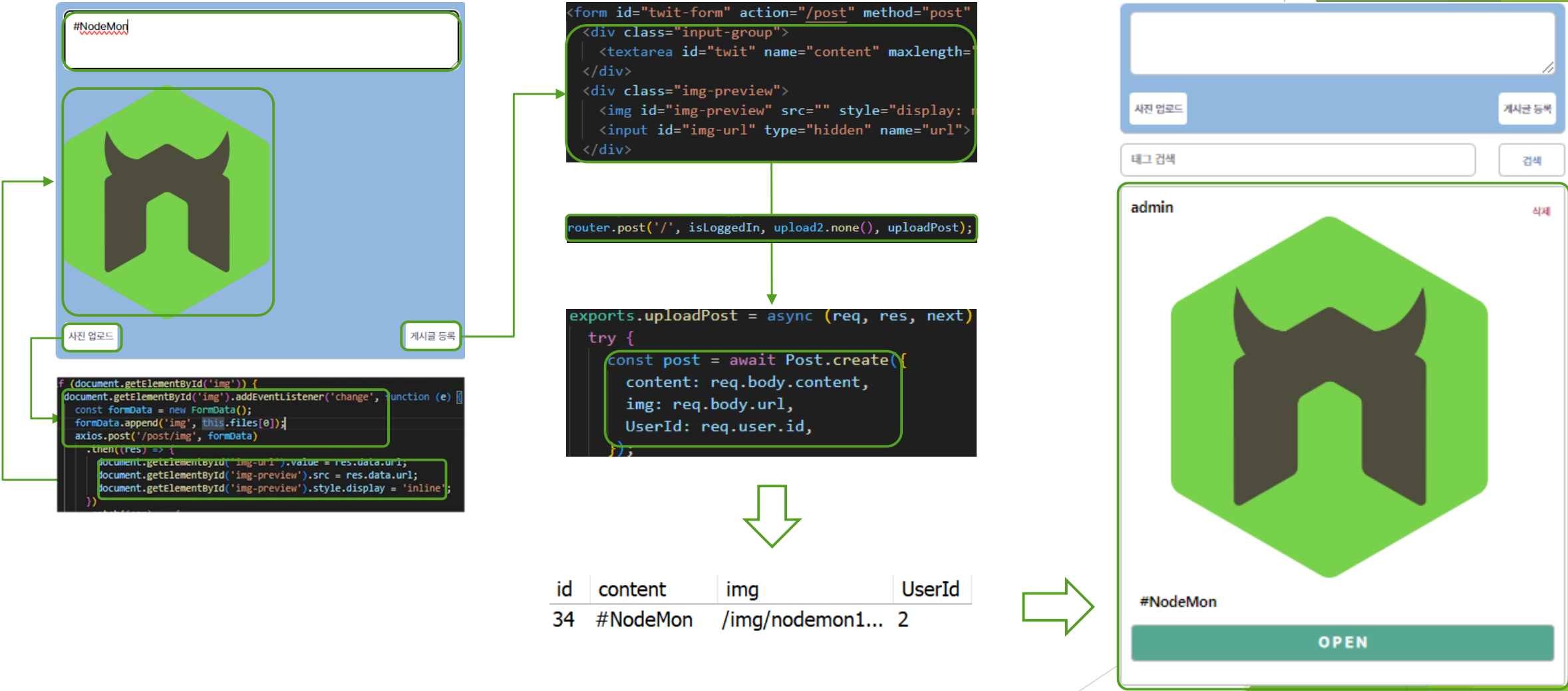
CLOSE

1. 로그인 / 카카오 로그인
2. 태그 검색 및 피드 조회
3. 게시물 & 댓글 작성 및 삭제

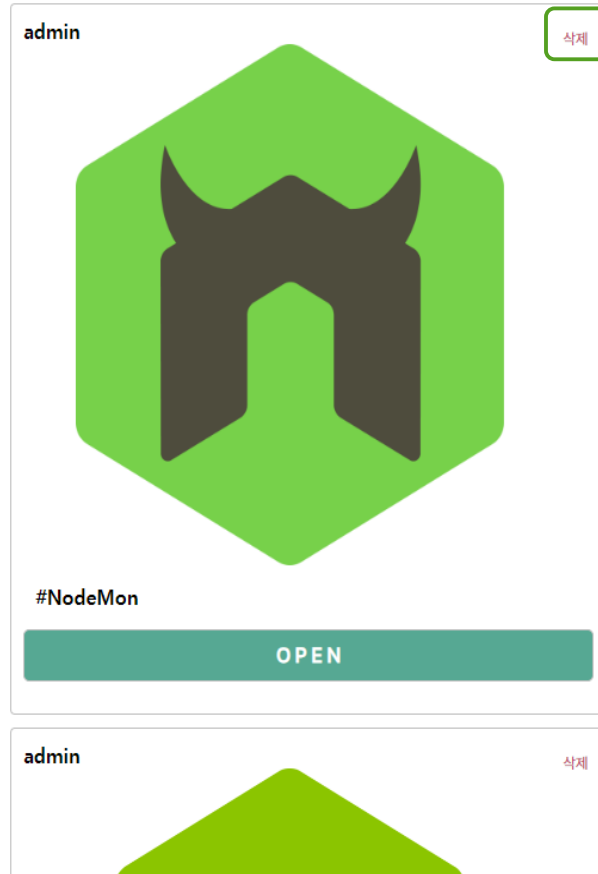
회원 가입 및 로그인



게시물 작성



게시물 삭제



```
function deleteTwit(twitId) {
  const loggedinfo = document.querySelector('#my-id').value;
  axios.delete(`post/delete/${twitId}/${loggedinfo}`)

  router.delete(`/delete/:twitId/:userId`, isLoggedIn, deletePost);
}

exports.deletePost = async (req, res) => {
  const twitId = req.params.twitId; // 게시물 id
  const reqUserId = req.params.userId; // 작성자 id
  const user = parseInt(req.user.id, 10); // Passport 세션 인증된 사용자 정보

  if (user == reqUserId) { // 세션 인증된 사용자와 클라에서 요청한 사용자 정보 검증
    try {
      const comments = await Comment.findAll({ // 1. 게시물에 연결된 모든 댓글 찾기
        include: {
          model: Post,
          through: "PostComment",
          where: { id: twitId }, //
        },
      });
      for (const comment of comments) {
        await comment.destroy(); // 2. 게시물에 작성된 모든 댓글 삭제
      }
      const post = await Post.findOne({ // 3. 삭제 요청한 게시물 조회
        where: {
          id: twitId,
          UserId: reqUserId,
        },
      });
      if (!post) {
        return res.status(404).json({ error: "게시물을 찾을 수 없습니다." });
      }
      await post.destroy(); // 4. 삭제
      res.json({ message: "게시물이 삭제되었습니다." });
    } catch (error) {
      console.error(error);
      res.status(500).json({ error: "내부 서버 오류" });
    }
  } else {
    return res.status(401).json({ error: "삭제할 권한이 없습니다!" });
  }
}
```



댓글 작성 및 삭제

댓글 작성

CLOSE

#NodeMon

작성자 : 3 | 댓글 1번

작성일: 2024-01-26T07:40:21.000Z

댓글 삭제

#NodeMon

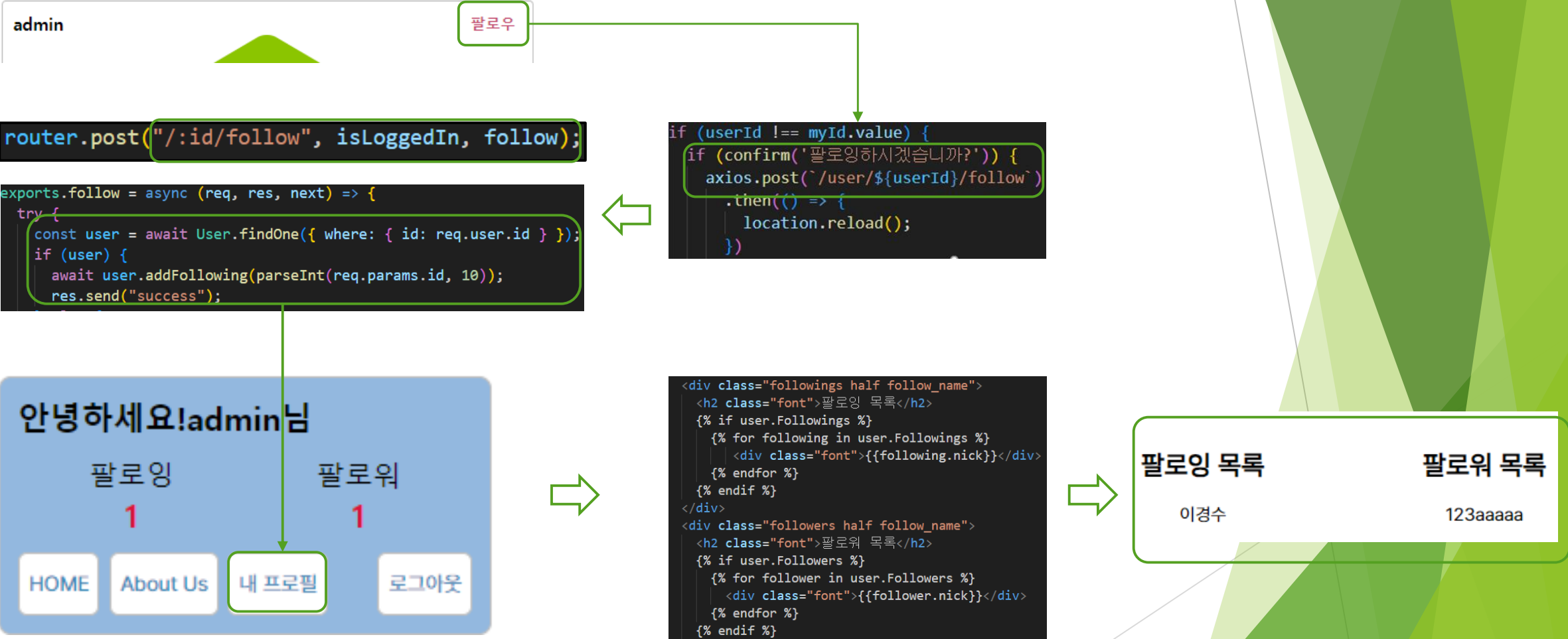
댓글 작성

CLOSE

```
exports.deleteComment = async (req, res, next) => {
  const reqId = parseInt(req.params.reqId, 10); // 삭제요청한 회원
  const userId = parseInt(req.user.id, 10); // 패스포트 로그인 회원 정보
  const reqcommentId = parseInt(req.params.commentId, 10); // 요청한 댓글

  try {
    const comment = await Comment.findOne({ // 삭제할 댓글 조회
      where: {
        id: reqcommentId,
        UserId: userId,
      },
    });
    if (!comment) {
      return res
        .status(404)
        .json({ success: false, message: "해당 댓글을 찾을 수 없습니다." });
    }
    if (reqId === userId) { // 패스포트 == 클라이언트 회원정보 확인
      await comment.destroy();
      return res
        .status(200)
        .json({ success: true, message: "댓글이 성공적으로 삭제되었습니다." });
    } else {
      return res
        .status(401)
        .json({ success: false, message: "삭제할 권한이 없습니다." });
    }
  } catch (error) {
    console.error(error);
    return res
      .status(500)
      .json({ success: false, message: "댓글 삭제 중에 오류가 발생했습니다." });
  }
};
```

팔로우 및 조회



5. 평가

프로젝트 피드백

1. 이번 프로젝트를 통해 Json 응답으로 페이지를 구현할 수 있게 되었고 이와 관련하여 sequelize 모듈을 구성하고 사용하는데 있어, Join 쿼리를 이용하는데 어려움이 있었지만 기존 ORM 스프링 부트 JPA 개발 경험이 있어 기능을 완성할 수 있게 되었습니다.
2. 정형화된 자바와 달리 스크립트 문법으로 작업함에 따라 새로운 개발 방식의 작업의 어려움을 느꼈지만 자바와 자바스크립트 간의 차이점과 특성을 알게 된 프로젝트였습니다.

자체 평가 - 6점

1. 댓글 목록 조회시 댓글 작성자의 ID만 표현하여, 사용자의 가독성이 떨어질 수 있습니다.
2. UI의 통일성 및 직관성이 부족하여 사용자에게 접근성이 용이한 서비스 이용 환경이 조성되지 않았습니다.
3. 아직은 JS문법에 익숙하지 않아, 많은 기능을 구현하지 못했지만 추후 수정작업을 통해 SNS의 기본/세부적인 기능 구현을 할 계획입니다.



감사합니다

입사 희망자 이경수