

15. 선택적 도전 과제-1-1 탐색

1. 요구사항 확인

- 회원가입시 이메일 인증 기능을 추가하세요.

2. gpt 답변

이 NestJS 코드는 사용자 인증 및 관리를 위한 기본 구조를 제공합니다. 현재 로그인, 비밀번호 변경, 프로필 조회, 사용자 등록 및 조회 기능이 있습니다. 요청하신 대로 회원가입 시 이메일 인증 기능을 추가하기 위해 다음 단계를 따라야 합니다:

1. **이메일 인증 토큰 생성 및 저장:** 사용자가 회원가입을 할 때, 이메일 인증을 위한 토큰을 생성하고, 이 토큰을 사용자 레코드에 저장해야 합니다.
2. **이메일 발송 로직 추가:** 생성된 토큰을 포함한 인증 링크를 사용자의 이메일로 보내는 로직을 추가해야 합니다. 이를 위해 NestJS 메일러 모듈을 사용할 수 있습니다.
3. **이메일 인증 엔드포인트 생성:** 사용자가 인증 링크를 클릭했을 때 호출될 엔드포인트를 생성합니다. 이 엔드포인트는 토큰을 검증하고, 사용자의 이메일 인증 상태를 업데이트합니다.
4. **회원가입 로직 수정:** 회원가입 로직에서 사용자가 이메일 인증을 완료할 때까지 사용자를 '비활성화' 상태로 두어야 합니다. 이메일 인증이 완료되면 사용자를 '활성화' 상태로 변경합니다.
5. **추가적인 보안 조치:** 토큰의 유효기간 설정, 이메일 인증 상태 확인을 요구하는 기능 등을 추가할 수 있습니다.

아래는 이러한 기능을 구현하기 위한 간략한 코드 예시입니다:

```
// users.service.ts
// ... 기존 코드 ...

// 사용자 이메일 인증 토큰 생성 및 저장
async setEmailVerificationToken(username: string): Promise<void> {
  const verificationToken = /* 토큰 생성 로직 */;
  await this.userRepository.update({ username }, { emailVerificationToken:
verificationToken });
}

// ...

// auth.controller.ts
```

```
// ... 기존 코드 ...

@Public()
@Post('register')
async register(@Body() registerDto: RegisterDto) {
  // ... 기존 회원가입 로직 ...

  // 이메일 인증 토큰 생성 및 저장
  await this.userService.setEmailVerificationToken(registerDto.username);

  // 인증 이메일 발송 로직
  // ...

  return { username: registerDto.username, message: "Please verify your email to complete registration." };
}

// 이메일 인증 엔드포인트
@Public()
@Get('verify-email')
async verifyEmail(@Query('token') token: string) {
  // 토큰 검증 및 사용자 상태 업데이트 로직
  // ...
}
}
```

이 기능을 완전히 구현하려면 이메일 발송 로직, 토큰 생성 및 검증 로직, 데이터베이스 스키마 변경 등을 추가로 고려해야 합니다.

- 코드 이메일로 보내고 그거 써서 받으면 끝인줄 알았는데 뭔가 복잡하다.

3. nestjs 이메일 인증 벨로그

- <https://velog.io/@mageboy/server-Nest.js-%ED%9A%8C%EC%9B%90%EA%B0%80%EC%9E%85-%EB%B0%8F-%EC%9D%B4%EB%A9%94%EC%9D%BC-%EC%9D%B8%EC%A6%9D-%EA%B5%AC%ED%98%84-%EC%97%B0%EC%8A%B5>
- 이거 참고해서 일단 만들어보자. 여차하면 커밋한거 되돌리면서 해야할듯