

05. bcrypt 작성

auth.service.ts

```
import { Injectable } from '@nestjs/common';
import { JwtService } from '@nestjs/jwt';
import * as bcrypt from 'bcrypt';

@Injectable()
export class AuthService {
  constructor(private jwtService: JwtService) {}

  // 사용자 검증 로직
  async validateUser(username: string, pass: string): Promise<any> {
    // 여기에 사용자 검증 로직을 구현합니다.
    // 예: 데이터베이스에서 사용자 정보를 조회하고 비밀번호를 비교합니다.
  }

  // 로그인 로직
  async login(user: any) {
    const payload = { username: user.username, sub: user.userId };
    return {
      access_token: this.jwtService.sign(payload),
    };
  }

  // 비밀번호 해시 생성
  async hashPassword(password: string): Promise<string> {
    const salt = await bcrypt.genSalt();
    return bcrypt.hash(password, salt);
  }

  // 저장된 해시와 비밀번호 비교
  async comparePasswords(password: string, storedPasswordHash: string):
    Promise<boolean> {
    return bcrypt.compare(password, storedPasswordHash);
  }
}
```

```
}
```

1. 트러블 슈팅

```
npm install --save-dev @types/bcrypt  
npm install --save-dev @types/passport-jwt
```