import { JwtService } from '@nestjs/jwt';

import \* as bcrypt from 'bcrypt';

import { UserService } from '../user/user.service';

import { User } from '../user/user.entity';

@Injectable()

export class AuthService {

constructor(

private userService: UserService,

private jwtService: JwtService,

) {}

async validateUser(email: string, pass: string): Promise<any> {

const user = await this.userService.findOneByEmail(email);

if (user && await bcrypt.compare(pass, user.password)) {

const { password, ...result } = user;

return result;

}

return null;

}

async login(user: any) {

const payload = { username: user.email, sub: user.id };

return {

access\_token: this.jwtService.sign(payload),

};

}

async register(user: User): Promise<User> {

const hashedPassword = await bcrypt.hash(user.password, 10);

return this.userService.create({ ...user, password: hashedPassword });

}

}