

```

        detail="User with email provided exists already"
    )
    await user_database.save(user)
    return {
        "message": "User created successfully."
    }

```

해당 이메일의 사용자가 존재하는지 확인하고 없으면 데이터베이스에 등록한다.

이번에는 로그인 라우트를 다음과 같이 변경하자.

```

@user_router.post("/signin")
async def sign_user_in(user: UserSignIn) -> dict:
    user_exist = await User.find_one(User.email == user.email)
    if not user_exist:
        raise HTTPException(
            status_code=status.HTTP_404_NOT_FOUND,
            detail="User with email does not exist"
        )
    if user_exist.password == user.password:
        return {
            "message": "User signed in successfully."
        }
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED,
        detail="Invalid details passed"
    )

```

이 라우트는 먼저 해당 사용자가 존재하는지 확인한다. 여기서는 아주 간단한 사용자 인증 방법을 사용하는데 실제 프로덕션에는 권장하지 않는 방법이다. 인증은 <CHAPTER 7 보안>에서 자세히 다룬다.

마지막으로 main.py를 수정해서 애플리케이션 실행 시 몽고DB를 초기화하도록 만든다. 강조된 코드를 추가하면 된다.

```

from database.connection import Settings

app = FastAPI()
settings = Settings()

```

CHAPTER 6 데이터베이스 연결 131