로그인 라우트 변경

routes/users.py에 다음과 같이 import문을 추가한다.

```
from fastapi import APIRouter, Depends, HTTPException, status
from fastapi.security import OAuth2PasswordRequestForm
from auth.jwt_handler import create_access_token

from models.users import User
```

이 코드는 FastAPI의 security 모듈에서 OAuth2PasswordRequestForm 클래스를 임포트한다. 이 클래스는 인증 정보(사용자명과 패스워드)를 추출하기 위해 로그인 라우트에 주입될 것이다. sign_user_in() 라우트 함수를 다음과 같이 변경하자.

```
async def sign_user_in(user: OAuth2PasswordRequestForm = Depends()) -> dict:
    user_exist = await User.find_one(User.email == user.username)
    ...
    if hash_password.verify_hash(user.password, user_exist.password):
        access_token = create_access_token(user_exist.email)
        return {
            "access_token": access_token,
            "token_type": "Bearer"
        }
    raise HTTPException(
        status_code=status.HTTP_401_UNAUTHORIZED,
        detail="Invalid details passed"
    )
```

앞서 언급했듯이 OAuth2PasswordRequestForm 클래스를 sign_user_in() 라우트 함수에 주입하여 해당 함수가 OAuth2 사양을 엄격하게 따르도록 한다. 함수 내에서는 패스워드, 반환된 접속 토큰, 토큰 유형을 검증한다. 이 라우트를 테스트하기 전에 models/users.py의 로그인용 응답 모델을 수정해서 UserSignIn 모델을 다음 토큰 모델로 교체하자(UserSignIn은 더 이상 사용되지 않는다).

```
class TokenResponse(BaseModel):
    access_token: str
    token_type: str
```