

```

(pyenv) + planner git:(main) x coverage run -m pytest
test session starts
platform darwin -- Python 3.8.2, pytest-7.1.2, pluggy-1.0.0
rootdir: /Users/youngestdev/Work/Building-Web-APIs-with-FastAPI-and-Python/ch08/planner, configfile: pytest.ini
plugins: asyncio-0.18.3, anyio-3.5.0
asyncio: mode=auto
collected 14 items

tests/test_arithmetic_operations.py .... [ 28%]
tests/test_fixture.py . [ 35%]
tests/test_login.py .. [ 58%]
tests/test_routes.py ..... [100%]

===== 14 passed in 0.57s =====
(pyenv) + planner git:(main) x

```

그림 8-15 테스트 커버리지 보고서

생성된 보고서를 확인해보자. 보고서는 터미널과 HTML로 생성된 웹 페이지 이렇게 두 가지 형식으로 볼 수 있다. 두 가지를 모두 살펴보자.

다음 명령을 실행하면 [그림 8-16]과 같이 터미널상에 표시된 보고서를 볼 수 있다.

```
(venv)$ coverage report
```

```

(pyenv) + planner git:(main) x coverage report
Name                               Stats    Miss  Cover
-----
auth/__init__.py                    0      0 100%
auth/authenticate.py                9      1   89%
auth/hash_password.py              7      0 100%
auth/jwt_handler.py               25      5   80%
database/__init__.py                0      0 100%
database/database.py               44      2   95%
main.py                             18      3   83%
models/__init__.py                  0      0 100%
models/events.py                   22      0 100%
models/users.py                    12      0 100%
routes/__init__.py                  0      0 100%
routes/events.py                   41      4   90%
routes/users.py                    27      3   89%
tests/conftest.py                   23      0 100%
tests/test_arithmetic_operations.py 16      0 100%
tests/test_fixture.py               7      0 100%
tests/test_login.py                 17      0 100%
tests/test_routes.py               59      0 100%
TOTAL                             327     18   94%
(pyenv) + planner git:(main) x

```

그림 8-16 터미널상에 표시된 테스트 커버리지 보고서