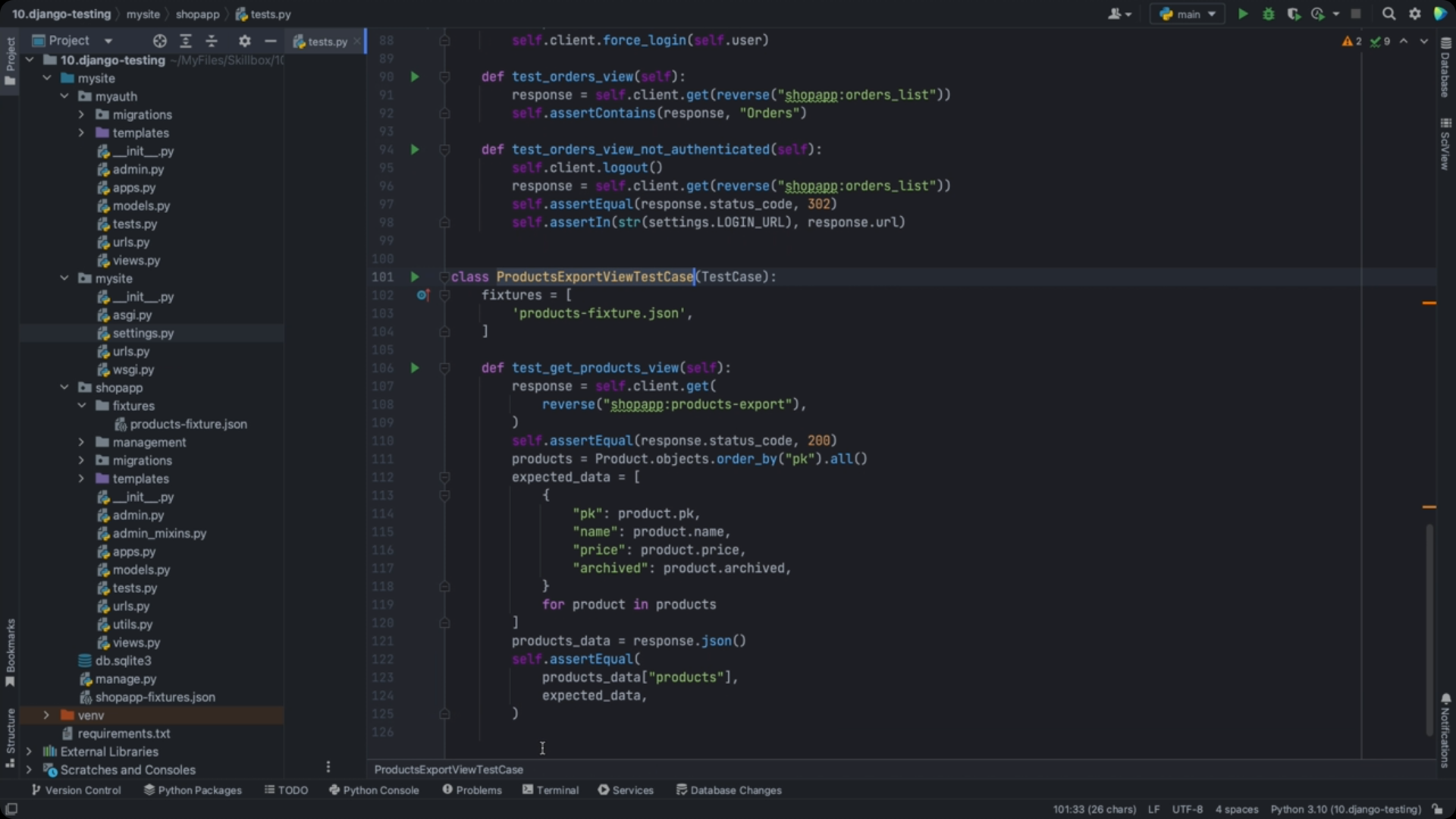
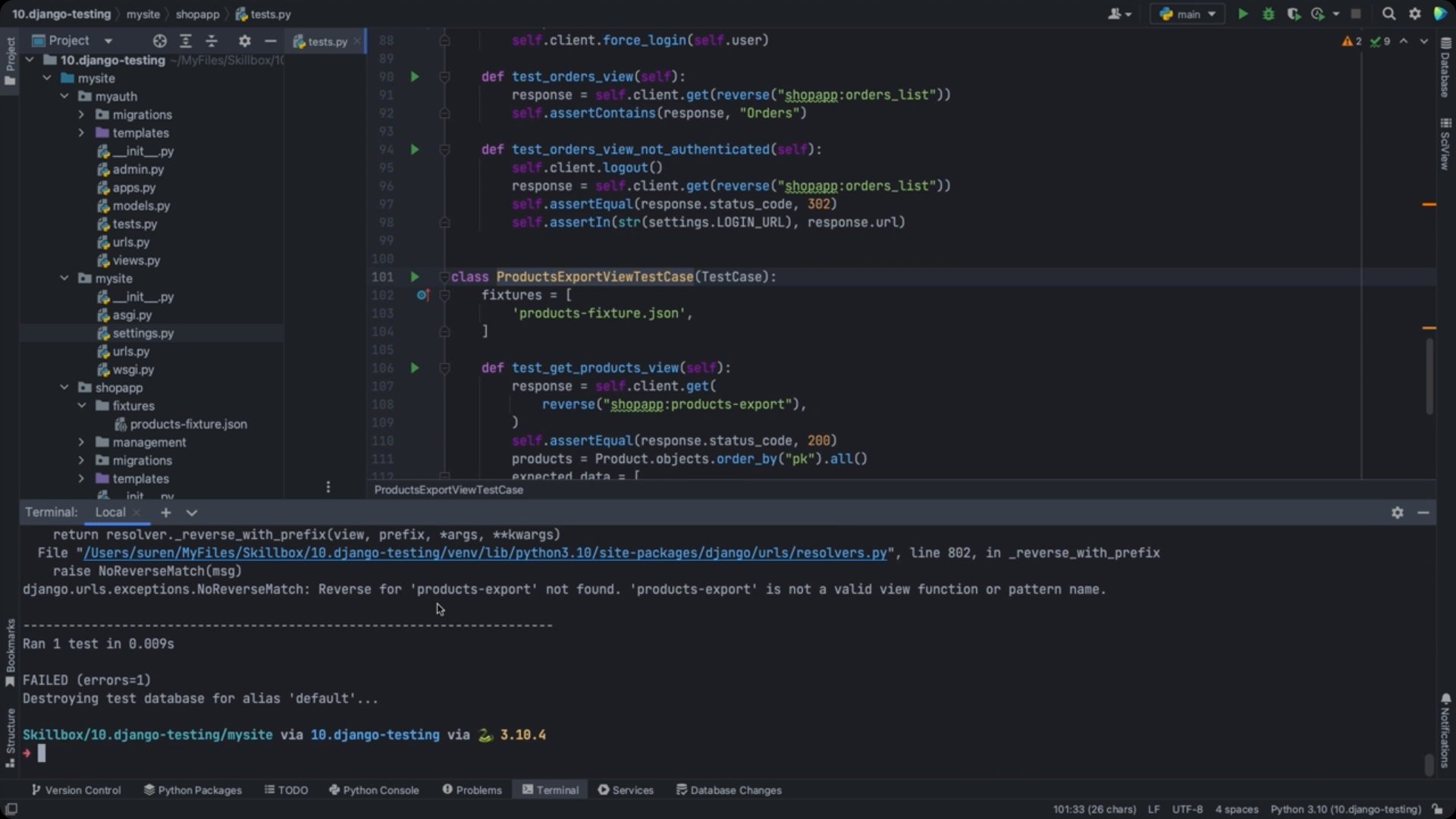
**10.5 TDD**

**Test** **Driven** **Development** (разработка через тестирование) – это методология разработки программного обеспечения, при которой разработчик сначала выполняет тестов функциональности перед тем, как написать эту новую функциональность. Тесты затем используют для поверки работоспособности кода после его написания. Разработка по методологии TDD состоит из трёх этапов:

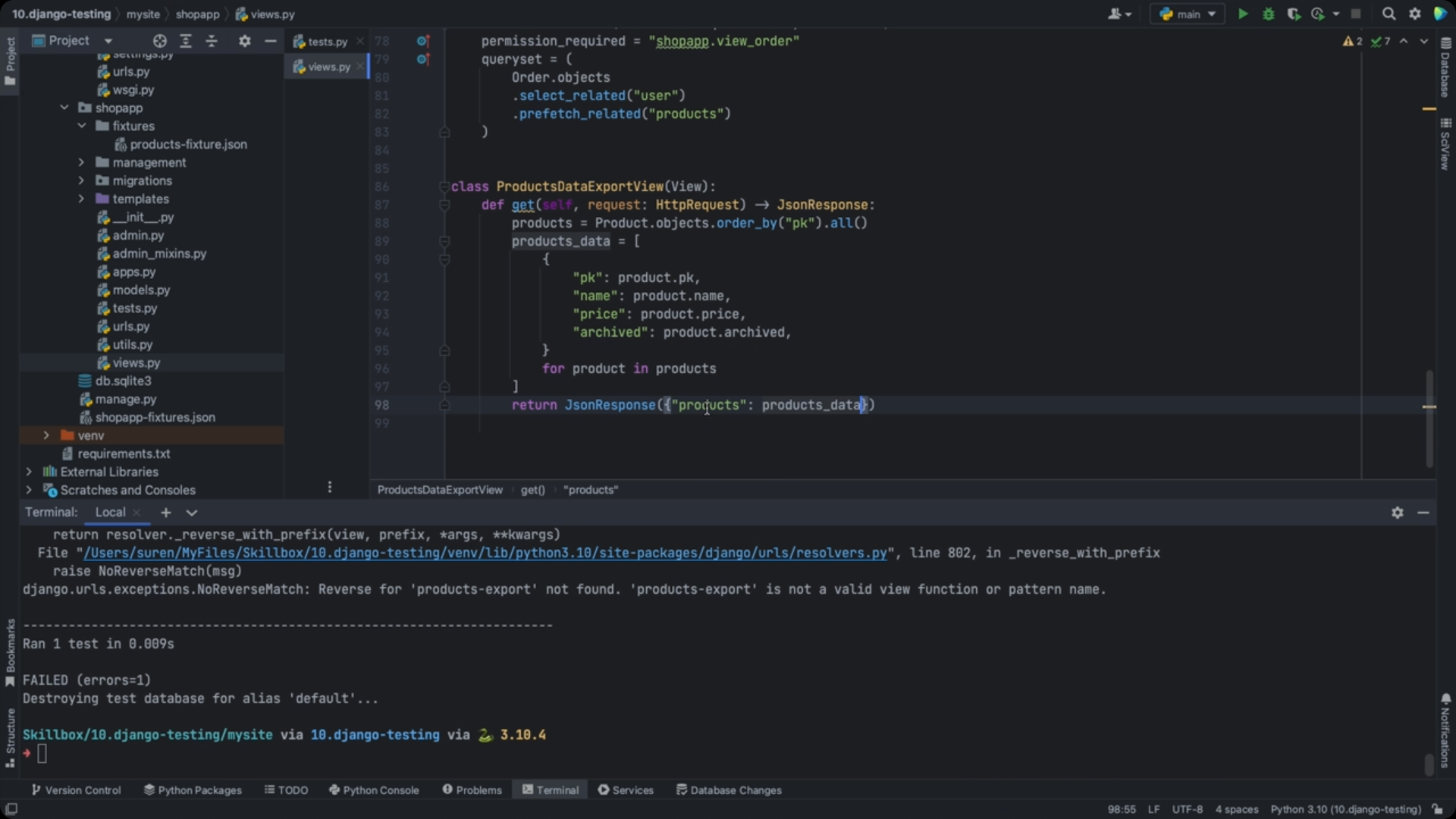
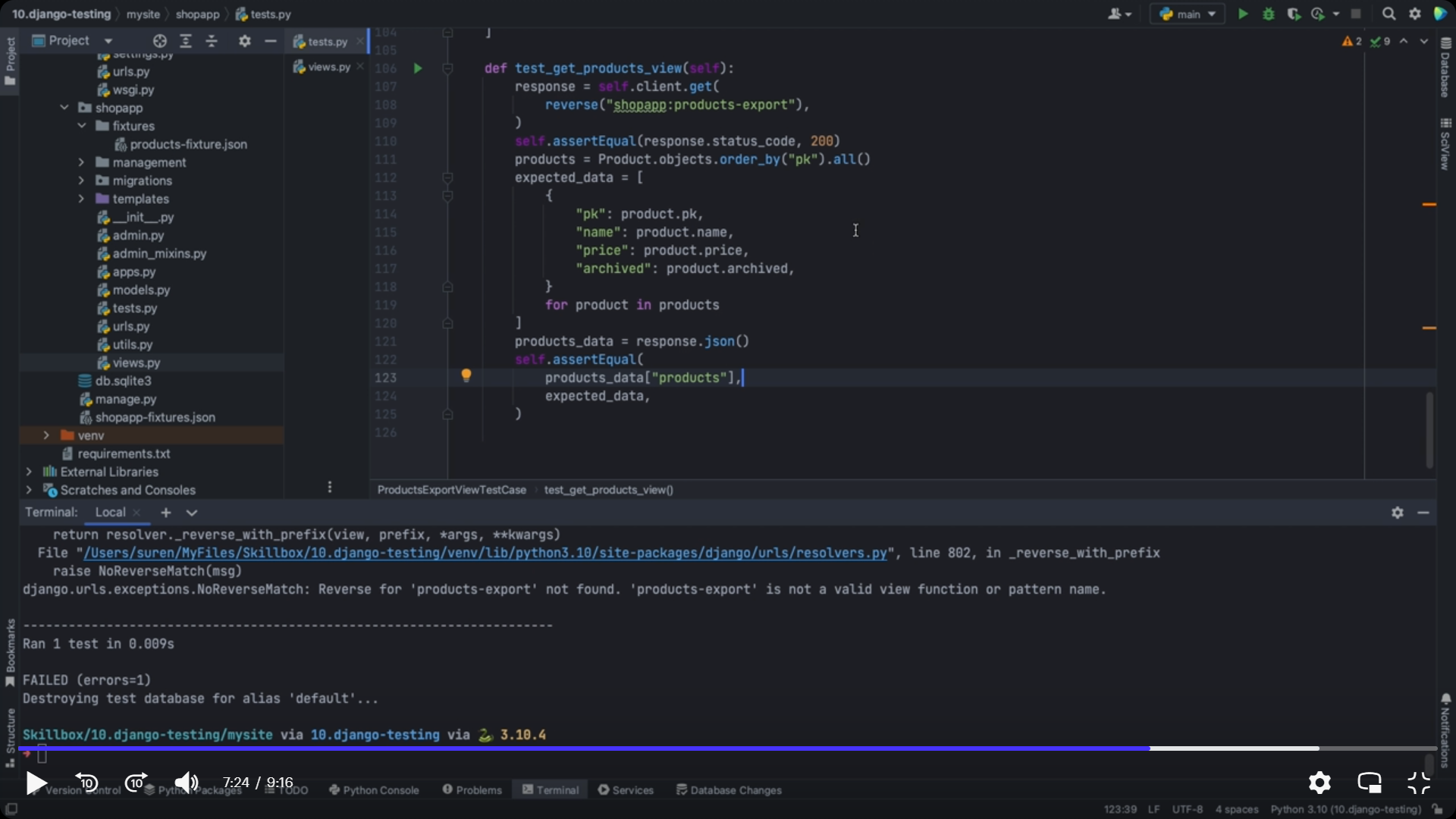
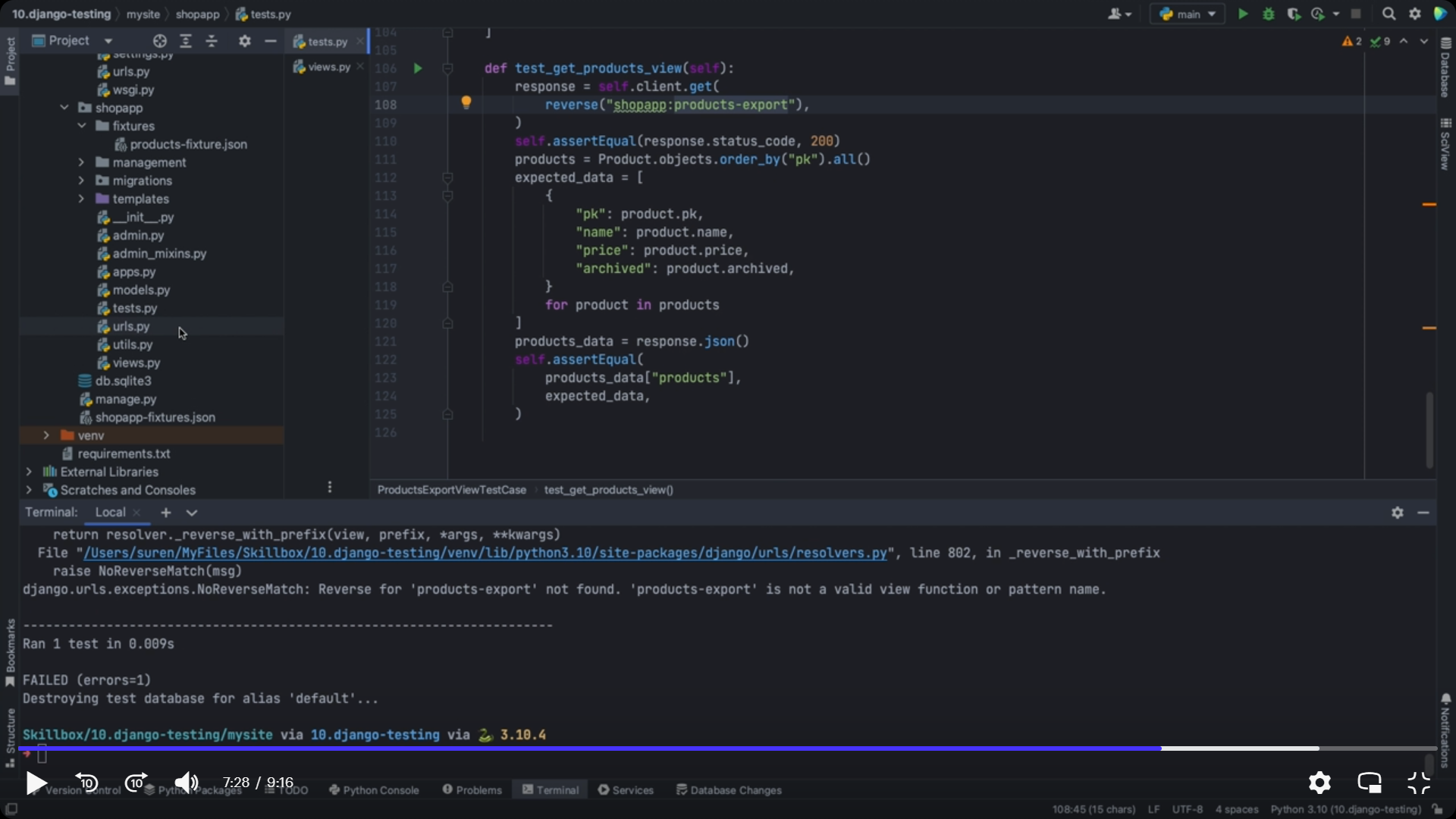
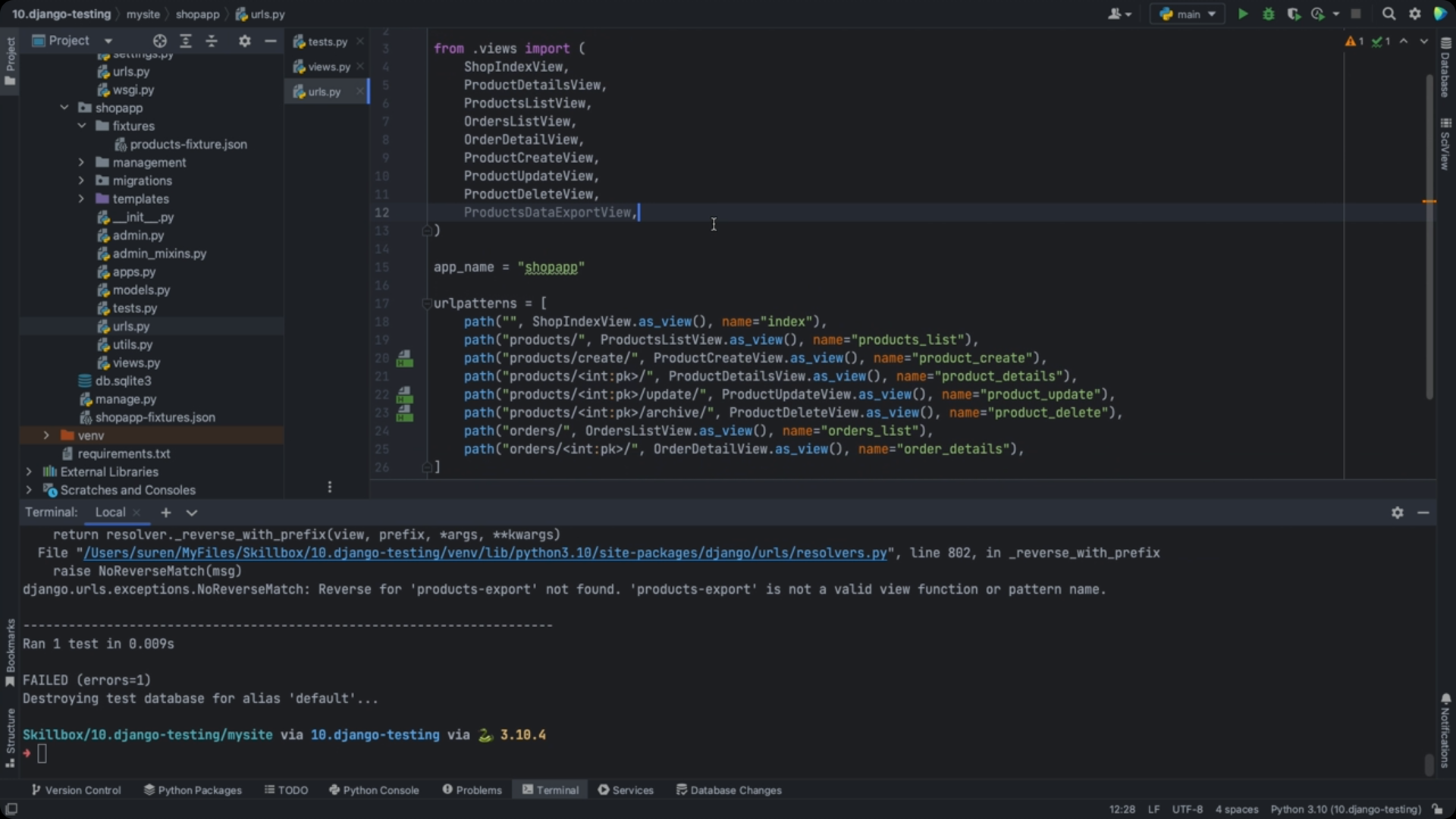
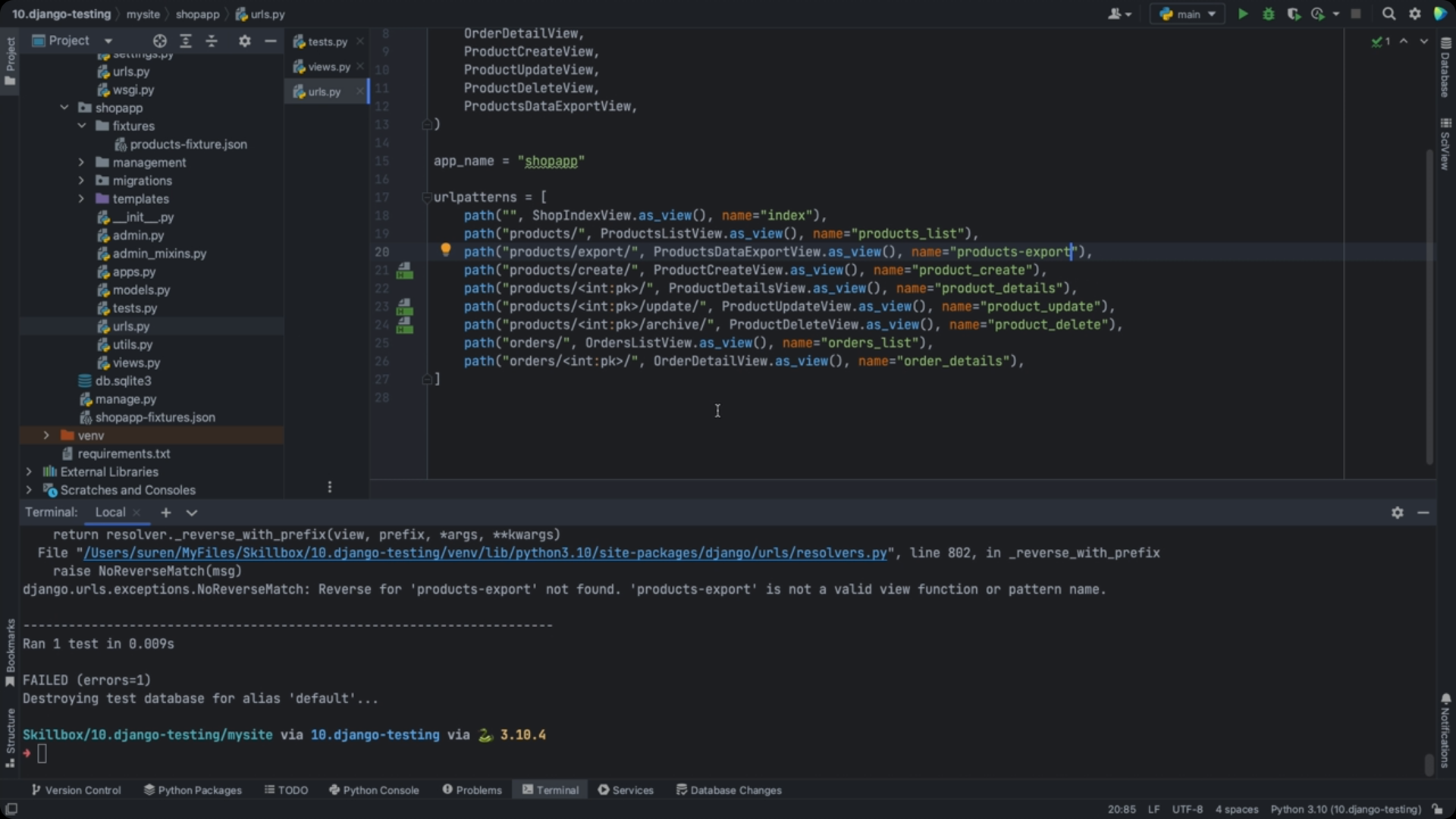
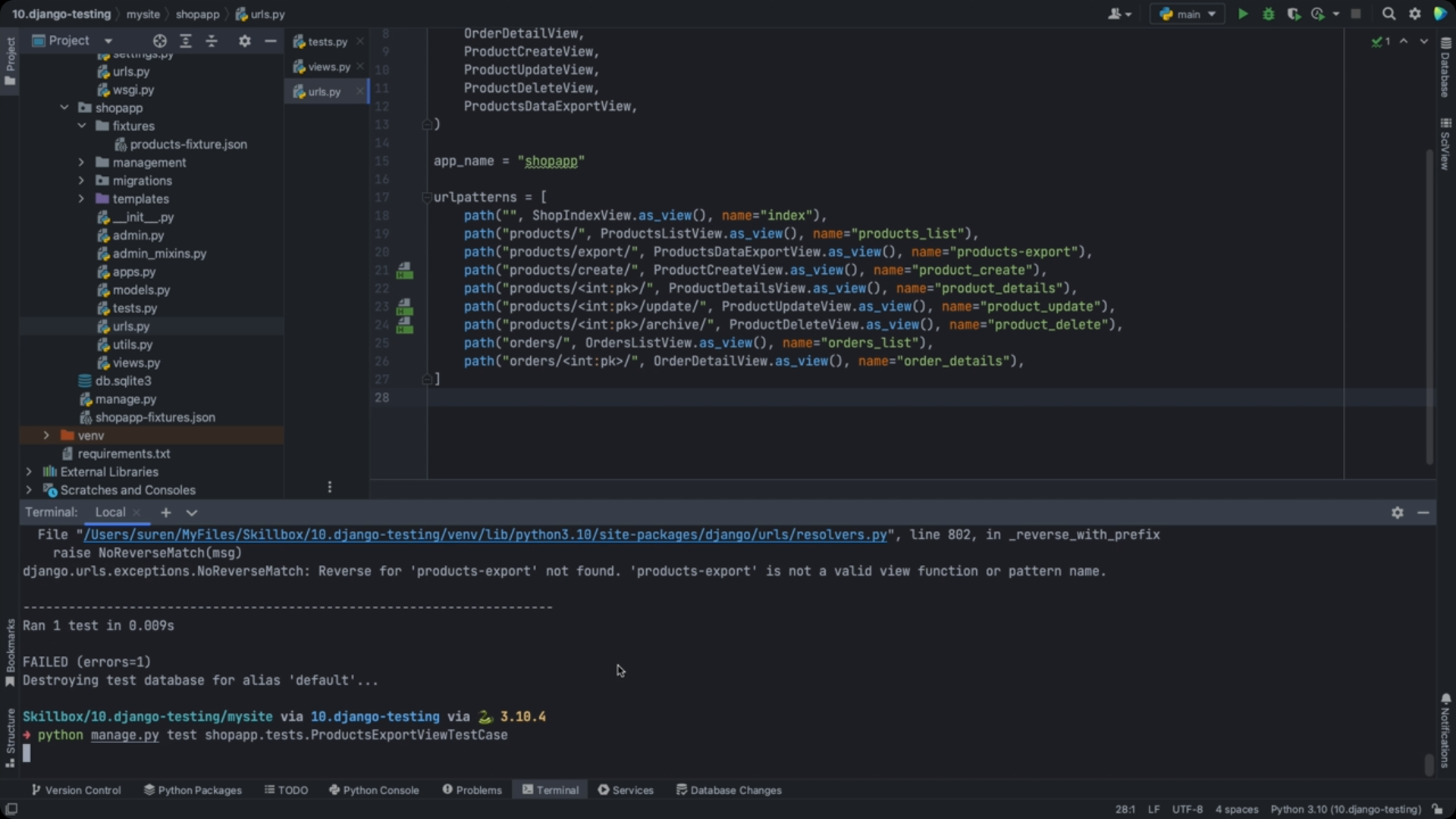
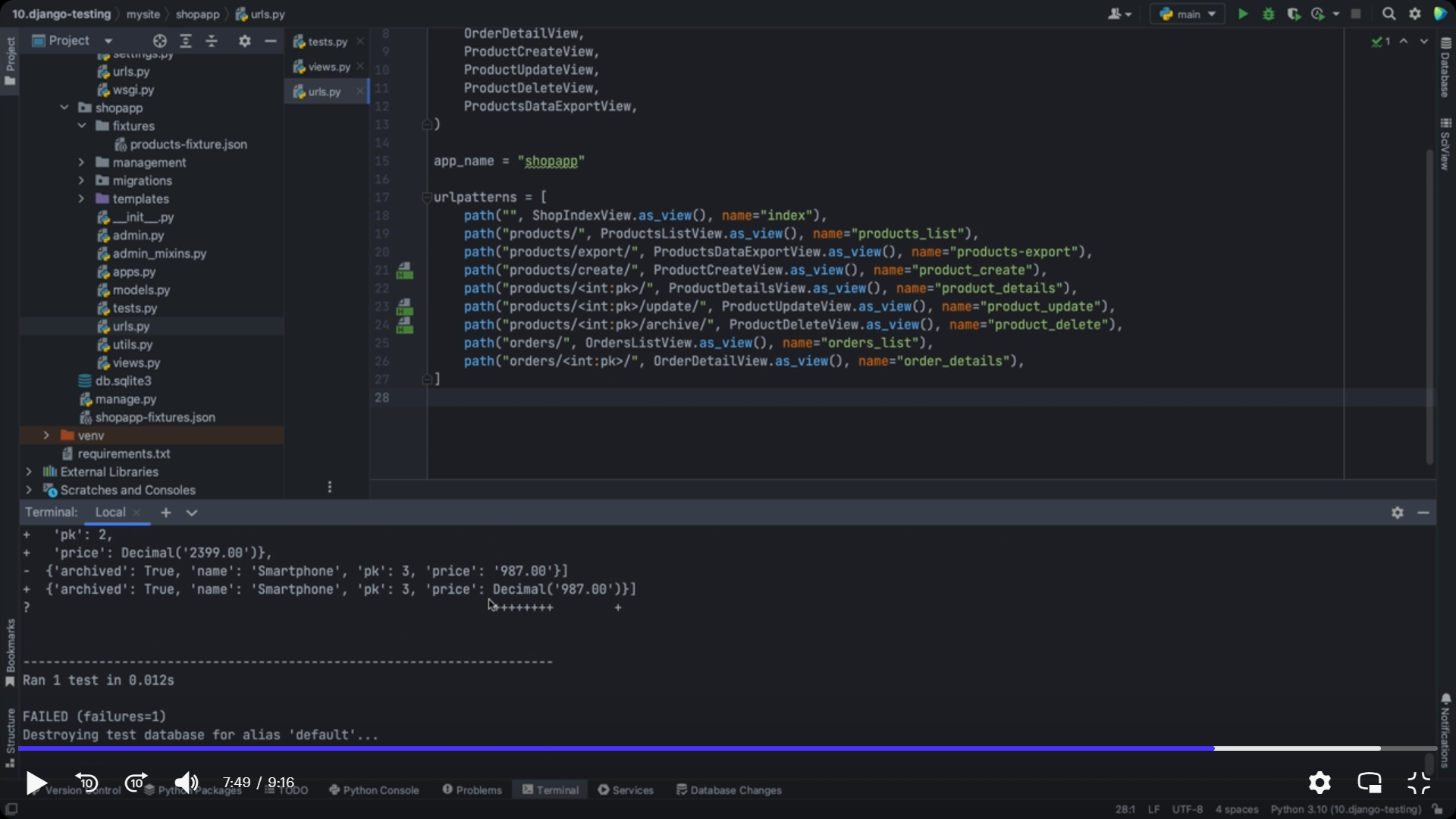
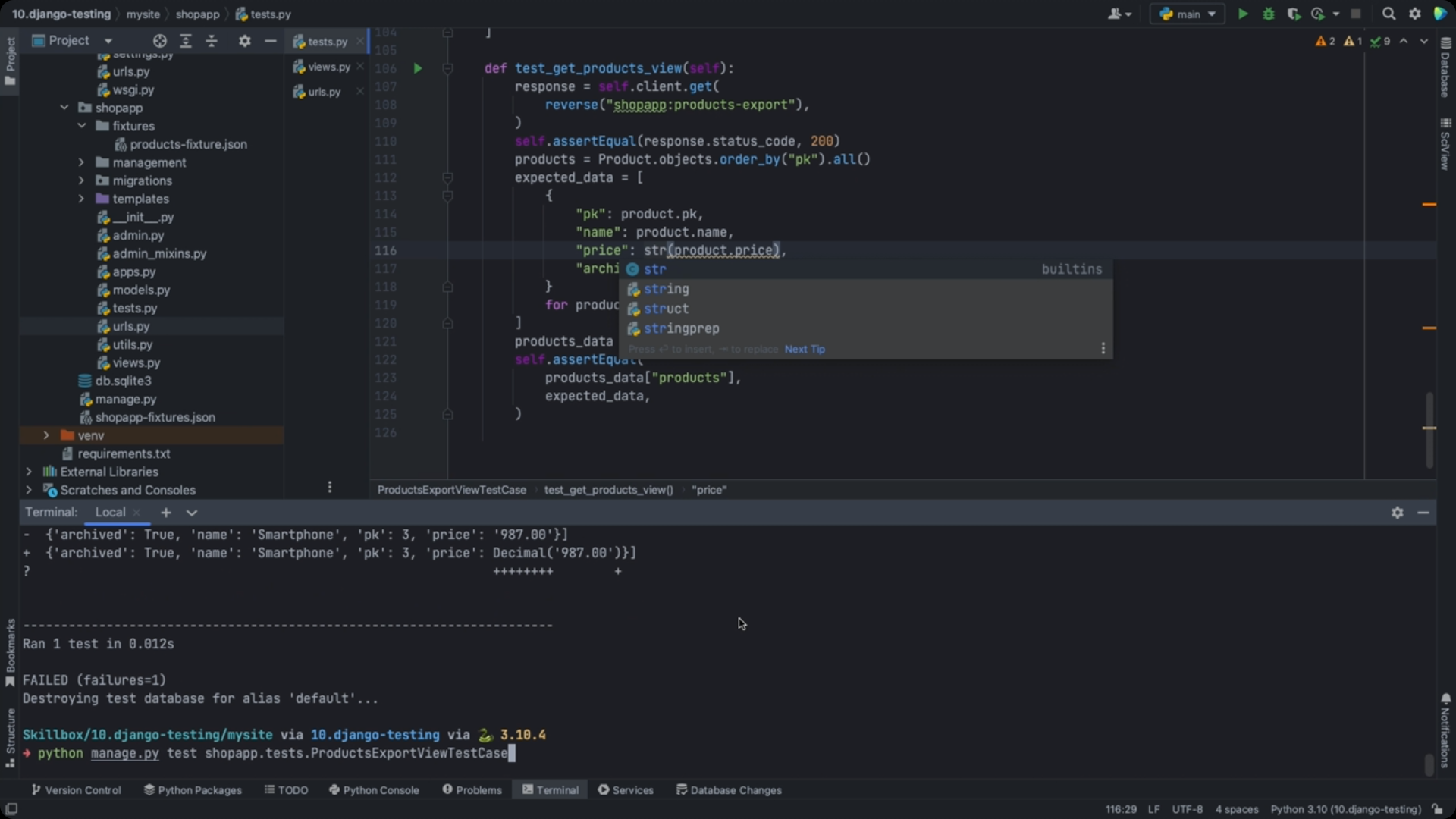
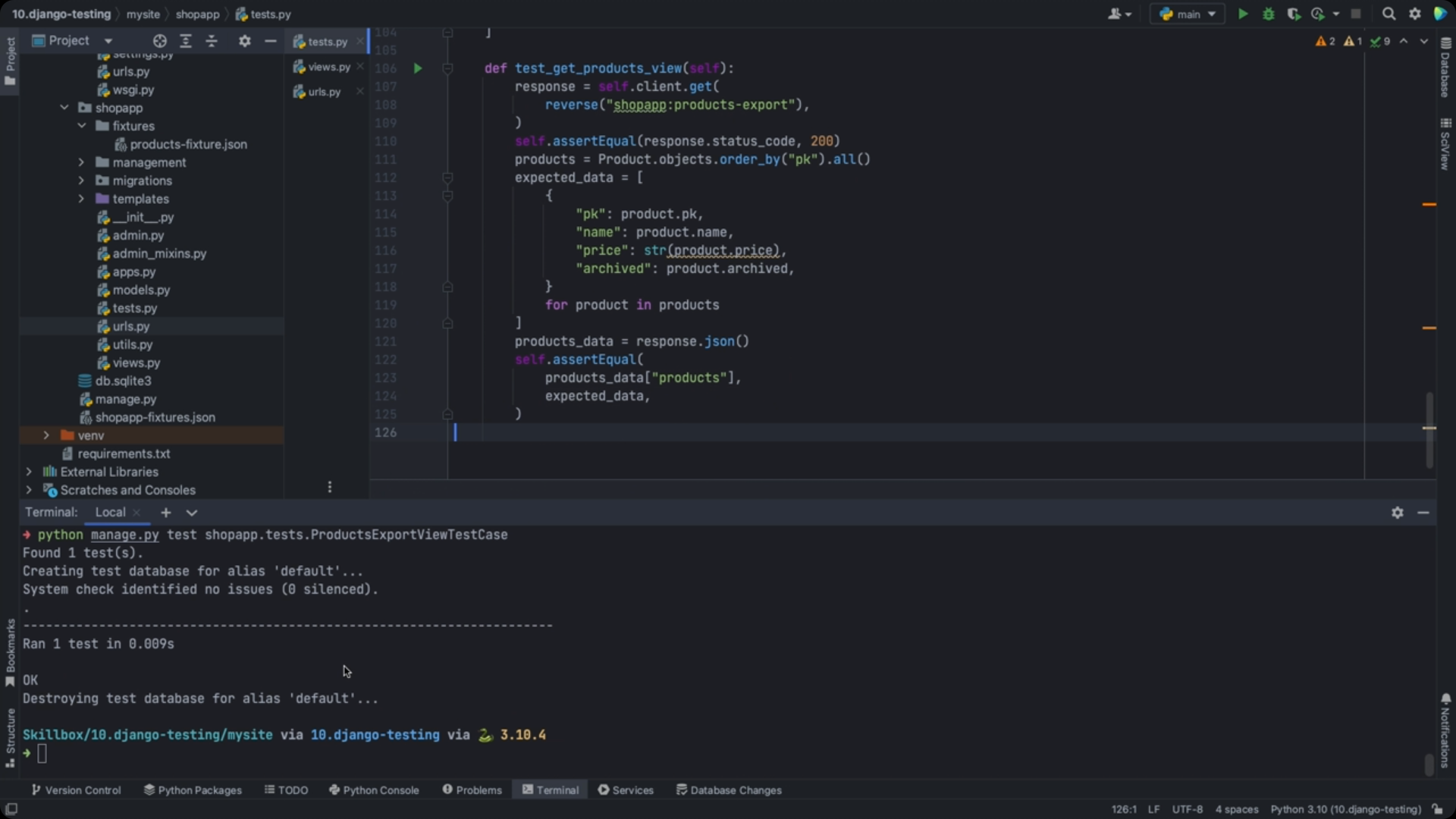
1. Написание тестов, которые «проваливаются». Если вы написали тесты, и они не проваливаются, то либо вы написали тесты неправильно, либо такая функциональность уже существует и можно уже перейти к написанию новой функциональности;
2. Написание кода так, чтобы тест начал проходить, т.е. вы пишете код, запускаете тест и если тест не проходит, то вы пишете код дальше так, чтобы тест проходил;
3. Запуск тестов и рефакторинг кода. Т.е. вы изменяете код, например, его просто «причёсываете», запускаете тесты и убеждаетесь, что ваши изменения не сломали фичу.

Для python есть множество библиотек для написания тестов, например, Unytest, Nous, Pytest, и др. Используя одну из этих библиотек и методологию TDD вы можете написать тесты для вашего кода и быть уверенными, что новые изменения не сломают функциональность существующего кода. Также вам будет легче отлаживать и поддерживать свой код в будущем.

Чтобы применять TDD нужен большой опыт разработки, т.к. необходимо понимание того, что собираетесь делать и как это будет работать в будущем. Ещё нужно знать, как будет построено приложение, к тому же это увеличивает длительность разработки.

Рассмотрим пример использования подхода TDD для создания API для выгрузки информации по продуктам. Для начала нужно написать тест, который будет проваливаться убедимся что данный тест не проходит ошибка появляется ещё на раннем этапе, когда django не может найти данный адрес. Поэтому нужно начинать исправлять все эти ошибки по очереди.

В тесте происходит следующее: сначала выполняется поиск ссылки, затем запрос по этому адресу, проверка статуса кода ответа и сравнение того, какие данные пришли.

Перейтиде во view в shopapp и объявите новый класс обратите внимание, что этот products на верхнем уровне продикован тестом где идёт обращени к products на верхмен уровне остаётся подключить новый view класс по адресу products-export для этого перейдите в urls  сохраните изменения и выполните тест вы увидите, что тест не прошёл в данном случае потому, что в ожидаемом результате в указанном результате в виде Decimal, но Decimal в json не проходит. В json проходит просто строка. Поэтому вам нужно перейти к тестам и на product.prise указать как строку в таком случае тесты пройдут 

Если тест проходит вы можете почистить написанный код.

Описание

* [Test-driven development](https://en.wikipedia.org/wiki/Test-driven_development)