SUGGESTION

The same as mentioned within the backend/models.py file is applicable here; you can leverage the os package to load environment variables instead of hardcoding sensitive values.

SUGGESTION

Great job with your test cases, but one thing you can do for improvement is establishing a strategy to handle data independently from start to end within the test suite context.

For example, if I run the test more than once, it would fail because it relies on an ID that wouldn't exist in the subsequent runs; therefore, the test case would fail.

There is more than one way of achieving that, but the most common patterns are:

* Execute the [SQLAlchemy's create\_all method](https://www.kite.com/python/docs/sqlalchemy.schema.MetaData.create_all" \t "_blank) before starting the tests and [the drop\_all method](https://www.kite.com/python/docs/sqlalchemy.schema.MetaData.drop_all) afterward.
* Organize your test cases so that you create an item, store the ID and use it in the relevant edit/delete testing cases.

If you want more details about testing your Python application, [I recommend this guide](https://docs.python-guide.org/writing/tests).