This exercise is meant to test your skills in Python, programming logic, structuring of a multi-tier application, and database modeling. The aesthetics of the application are not as important as the programming assessment. Use Tornado, SQLAlchemy, and SQLite to complete the task detailed below.

**Required Build**

* Using Tornado, create 2 individual home pages for 2 clients: Jack and Jill.
* A basic login page should authenticate each user and redirect them to their respective home pages where they can view their bank accounts.
* Use SQLAlchemy to model a database that holds bank account information for Jack and Jill.
  + Each client should have 2 accounts; a Checking account and a Trading account.
  + Both accounts will have balances and the balances for these accounts should be displayed on the client’s home page.
  + All database interaction (querying and populating data) must be handled through SQLAlchemy.

**Required Tasks**

* Jack and Jill should be able to trade through their trading accounts.
* Write a query to check that the trade requested by Jack or Jill, can be satisfied by their account.
* For a trade to be approved, the sum of that individual’s trading and checking account must be greater than 20% of the trade value.
* The client must be informed of the approval status of the trades they request.
  + Approved trades should be reflected in the adjusted account balance.
  + For declined trades, the client should be notified of how much money should be deposited into their accounts to satisfy the requirement for trade approval.
* Write unit tests and generate your own fixes.

**Bonus Tasks**

The required tasks **MUST** be completed before the bonus task can be taken into consideration. Bonus tasks are there to flex your skills and to see how you would improve on a project.

* Dockerize the project including the tests
* Use MVVM design pattern in the UI design to make your project more responsive.
* Add any features you think would be useful.
* Implement a REST API into this project.