Plan a FIVE-minute "showcase" demo that highlights your application and demonstrates MVP (minimal viable product). Write a human-executable script explaining what you plan to show and what you plan to say, practice the demo to make sure everything works and fits in the time allotted, and include this script in your github repository. Make sure to tag the exact file versions to be used for the demo. Note this "showcase" demo will be presented on demo day. Thus you need to submit this document before the demo, not after.

Features to be presented

1. Show how to place an order and demonstrate it flowing through the pipeline
    a. Navigate to “place order” page and place an order
    b. Order input handles erroneous input (e.g. non-numeric input, non-positive input, non-integer input, overflow)
    c. Navigate back to home page, show off all progressing/completed orders.
    d. Click on a single order. Show off that we create all these child orders

2. Talk about the behavior of the trading algorithm e.g. when fail to sell then what happens, when success then what happens.
    a. The trading algorithm breaks each order up into smaller orders which are spread out evenly over a time period in order to lessen its market impact. (TWAP)
    b. When an order is entered, the trading algorithm first tries to execute a smaller order at 10% of the parent order’s volume.
    c. When an order fails multiple times, the algorithm will decrease the order size.
    d. When an order succeeds multiple times, the algorithm increases the order size.
    e. Successful orders are stored in the db.
    f. Orders can be paused/resumed/cancelled.
3. Show off UI features
    a. Order progress bar updates asynchronously without page refresh through AJAX
    b. Order detail page updates asynchronously without page refresh through AJAX
    c. Can pause/resume and cancel orders

What we plan to say (in prose)

1. Order placing
    a. “So here is the order placing page”
    b. “We display friendly UI messages for bad input e.g. integer overflow, negative numbers”
    c. “We issue a popup to confirm a user’s order, just in case he/she gives the wrong input”
    d. “Issuing an order leads us back to the index page, where a user can observe all his current/past orders”
2. Behavior of algorithm
    a. After an order is placed, the trading algorithm will break the order into smaller chunks which are sold over the course of the day. The algorithm first attempts to

execute a child order which is 10% of the parent order by volume. Observe that as the child orders fails to execute repeatedly, the algorithm will adjust to the market's supply by attempting to sell smaller and smaller child orders until it succeeds. Repeated successes also signal to the algorithm that the market's supply is sufficient, and it will start attempting to sell larger and larger child orders. (kinda done)

3. Cool UI features
    a. The user can view the status of an order through asynchronously updating progress bars and transaction tables.
    b. The user can pause and resume or even cancel an ongoing order.

Order/flow of presentation

1. Hi everyone, we are working on an app to achieve XYZ → proceed to demo the app
2. Demonstrate placing several orders → (e.g 10000, 1, 3000)
3. Navigate to home page and show that our system is tracking multiple orders
4. Show that progress bars are updating asynchronously for each order
5. Click on an order and show each child order being generated
    a. This is where Aaron will talk about how the trade algorithm behaves
    b. E.g. when a child order fails, the system will try to downsize the next child order
    c. Also point out that progress bar and transaction table updates asynchronously
6. Demo Pause/Resume and Cancel features
    a. Show that child orders stop being created when we pause
    b. And that they continue being created when we resume
    c. Show that order changes status when canceled
7. Show that each user profile can only view their own orders and that orders still continue to sell even when the user is logged out.
    a. Log into another account to show that it sees a different list of orders and that orders placed by this account does not influence orders in other accounts.