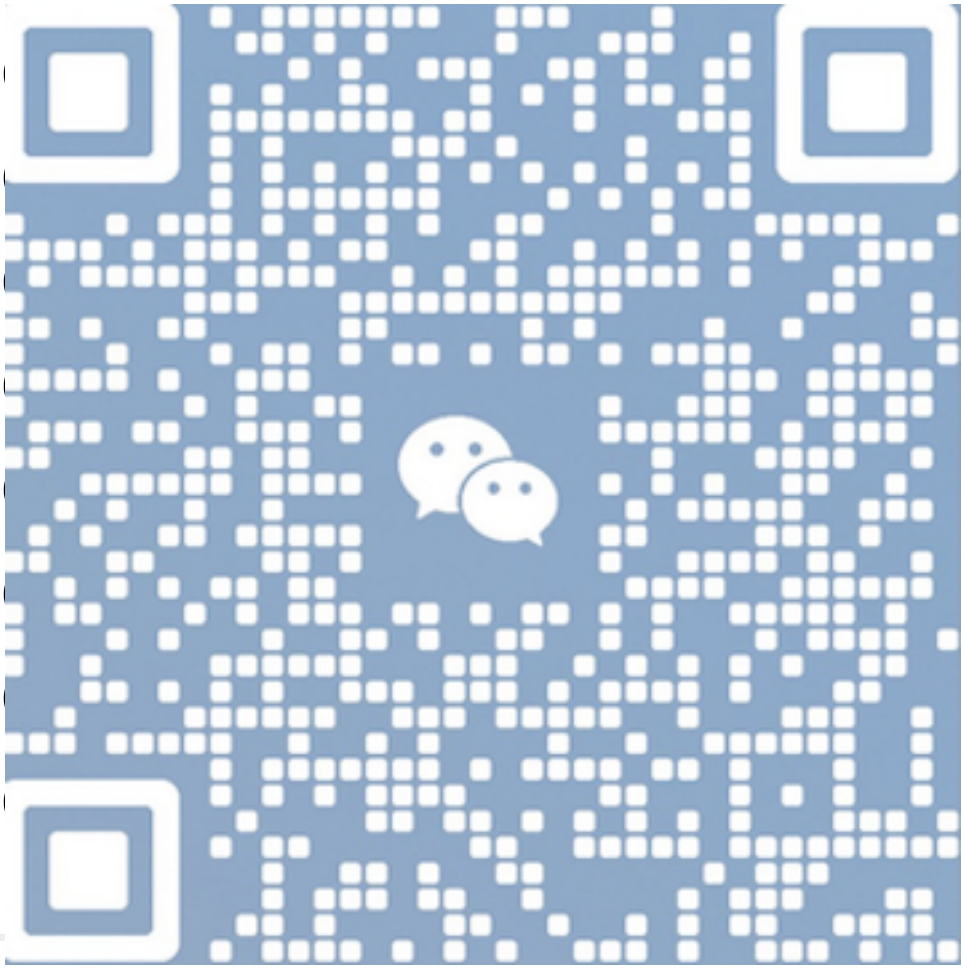
PROJECT 5: MARKETSIM

Table of Contents



REVISIONS

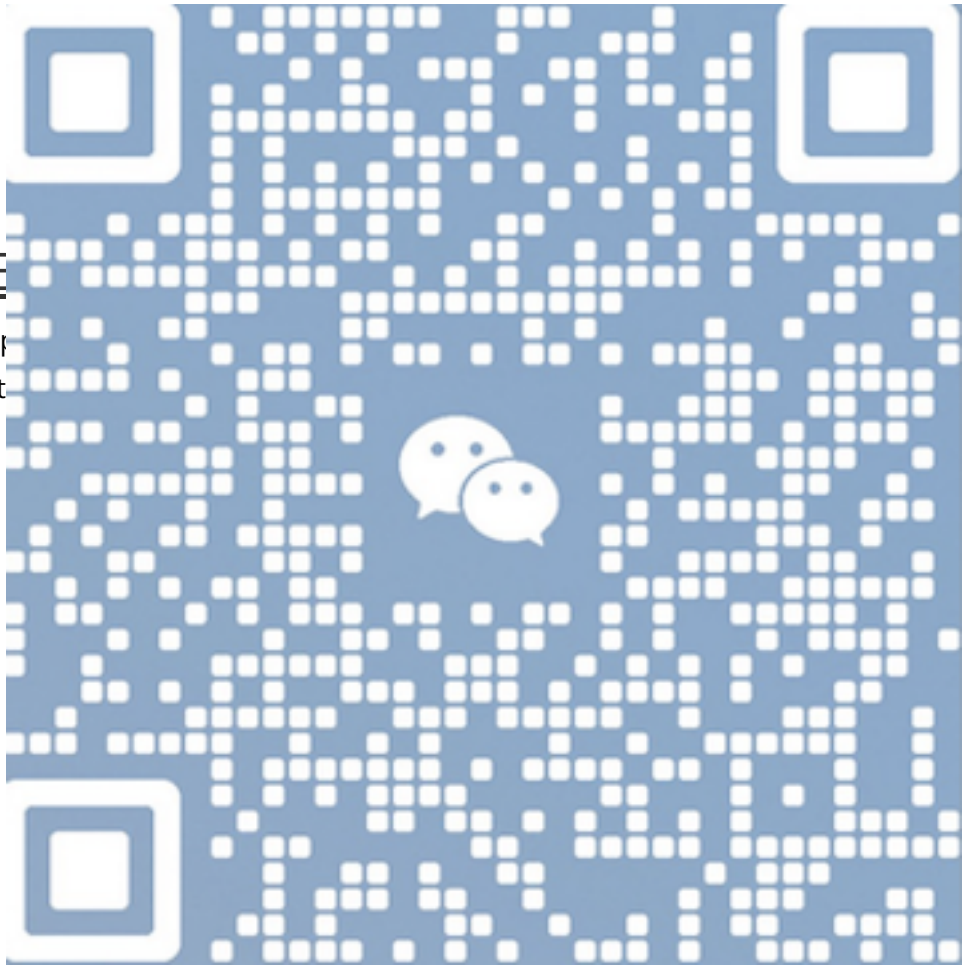
This assignment is subject to change up until 3 weeks prior to the due date. We do not anticipate changes; any changes will be logged in this section.

1 OVERVIEW

In this project, you will create a market simulator that accepts trading orders and keeps track of a portfolio's value over time. It also then assesses the performance of that portfolio. You will submit the code for the project to Gradescope SUBMISSION. There is no report associated with this project.

1.1 Learning Objectives

The specific learning objectives for this assignment are focused on the following areas:

- **Market Simulation**: Develop a market simulator. Variations of this market simulator will play a role in future projects.
-  they

2 AE

In this p a market

3 YOUR IMPLEMENTATION

Your submission must implement this API specification.

Before the deadline, make sure to pre-validate your submission using Gradescope TESTING. Once you are satisfied with the results in testing, submit the code to Gradescope SUBMISSION. Only code submitted to Gradescope SUBMISSION will be graded. If you submit your code to Gradescope TESTING and have not also submitted your code to Gradescope SUBMISSION, you will receive a zero (0).

3.1 Getting Started

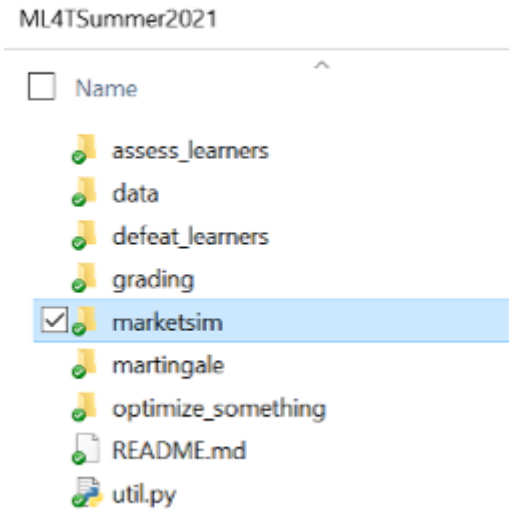
You will be given a starter framework to make it easier to get started on the project and focus o the local er ned from: M

Extract ew folder c

Within hat will

unction vided

th this



3.2 P s)

Your jo) that returns market

```
1  d                                    = 1000
2  #
3  r
marke                                   ew raw
```

The sta lers in the ord argume

-
-
- (both entry and exit)
- **impact** is the amount the price moves against the trader compared to the historical data at each transaction. Impact of 0.01 in the API corresponds to an impact of 1%.

Return the result (portvals) as a single-column pandas.DataFrame (column name does not matter), containing the value of the portfolio for each trading day in the first column from start_date to end_date, inclusive.

The files containing orders are CSV files with the following columns:

- Date (yyyy-mm-dd)
- Symbol (e.g. AAPL, GOOG)
- Order (BUY or SELL)
- Shares (no. of shares to trade)

For example:

```
Date,Symbol,Order,Shares
2008-12-3,AAPL,BUY,130
2008-12-8,AAPL,SELL,130
2
```

marke ew raw

Your si g
adjuste quities.
The res

```
2
2
2
.
```

marke ew raw

3.2.1

Your cc tfolio on
each da and
negativ position
for that stock. Negative cash means that you've borrowed money from the broker.

When a BUY order occurs, you should add the appropriate number of shares to the count for that stock and subtract the appropriate cost of the shares from the cash account. The cost should be determined using the adjusted close price for that stock on that day.

When a SELL order occurs, it works in reverse: You should subtract the number of shares from the count and add to the cash account.