

### Sprint 1 System Test Scenarios:

- A. User story 1 from sprint 1: As an investor, I want to retrieve the value of a specified ticker (without being logged in, viewing ads, opening the browser).
- B. User story 2 from sprint 1: As a developer, I want to be able to scrape with a function (GET)
- C. User story 3 from sprint 1: As a customer/investor, I want to use a presentable user interface.

### Scenario:

- 1. Fire up slug finance backend/frontend
- 2. go to individual page by typing localhost:3000/individual
- 3. Enter any US based stock in text box and hit "Enter"
- 4. User should see specified ticker(stock)'s performance for the year

### Sprint 2 System Test Scenarios:

- A. User story 1 from sprint 2: ("As an investor, I want to be able to have my own login to view my personalized portfolio so I can easily keep track of stocks that I am interested in")
  - a. Store login information inside the database (5 Hours)
  - b. Be able to authenticate users upon login (1 Hour)
  - c. Store different stocks in database per account (2 Hours)

### Scenario:

- 1. start slug finance backend/frontend
- 2. if you don't have an account you can click on "Sign Up"
- 3. if you do then you can click "Log In"
- 4. name = <[partickchen@ucsc.edu](mailto:partickchen@ucsc.edu)> password = <partick>
- 5. user should see a dialog saying "Success"
- 6. User should be able to now add stocks to their portfolio

- A. User story 2 from sprint 2: "As a data analyst, I want to be able to export data in a file with the click of a button"
  - a. Store historical price data of a stock (5 Hours)
  - b. Put the data inside a csv to export (1 Hour)

### Scenario:

- 1. start slug finance backend/frontend
- 2. go to localhost:3000 homepage
- 3. click on the menu button (3 lines) that represents TSLA's yearly performance
- 4. hit download csv

### Sprint 3 System Test Scenarios:

- A. User story 1 from sprint 3: “As an investor I would like to see FAANG displayed in chart”

#### Scenario:

1. Start slug finance backend/frontend
2. go to localhost:3000
3. User should be able to see Facebook, Apple , Amazon, Netflix, Google,

- B. As an investor, I want to see various types of charts/graphs for each stock in order to follow market trends

#### Scenario:

1. Start slug finance backend/ frontend
2. go to individual page by typing localhost:3000/individual
3. Enter any US based stock in text box and hit “Enter”
4. User should see specified stock’s history as a graph

### Sprint 4 System Test Scenarios:

- A. User story 1 from sprint 4: “As a trader, I’d like to be able to login to my account and see all stocks I’ve added to my watchlist, to see their performance”

#### Scenario:

1. Start slug finance app
2. Click login on navigation bar
3. Click “myPortfolio”
4. Users should be able to see any stocks they have added in a previous session, or stocks they have added through an individual page.

- B. User story 2 from sprint 4: “As a new trader, I’d like a description for each stock in order to educate myself on what this stock is and/or does”

#### Scenario:

1. Start slug finance app

2. Type in stock of interest into text box
3. Users should be able to see what the stock's main purpose is, as well as its sector that it is in.

C. As a trader, I want to be able to view my charts in various type of ways such as candlestick chart, or line chart.

1. Start slug finance app
2. redirect to individual page through localhost:3000/individual
3. type stock of interest
4. choose between candlestick and line graph

D. As an investor, I want to be able to import (and edit?) Yahoo portfolio from the app.

1. Start slug finance app
2. Make an account/login
3. redirect to individual page through localhost:3000/portfolio
4. Hit import

For running automated individual.test.js and portfolio.test.js

1. Start up docker
2. Start up backend
  - a. `python3 -m flask run`
3. Add a user "[lance@ucsc.edu](mailto:lance@ucsc.edu)" with create path
  - a. Either through swagger or UI
4. Add stocks to [lance@ucsc.edu](mailto:lance@ucsc.edu) with addtoportoflio path
  - a. Either through swagger or UI
5. In a separate terminal, navigate to frontend/react/src and run:
  - a. `npm run test -- --coverage --watchAll`

For running backend/app\_test.py

1. Navigate to backend dir
2. Create virtual environment if not already
3. Start virtual environment
4. Install requirements if not done already
5. `python3 -m pytest`