Team: Avery Arjang, Thomas Kokes, John, Xiang
Title: Team Stonkz

3 Features for testing:

Logging in:

- Users must log in, and be logged in to see graphs and access other website features.

Checking Stocks:

- Check to make sure that buy and sell features work. Also check that correct information is populated on the users profile page.

Being able to open wall street bets links:

- We want to test to make sure that all the reddit links are correctly being added to the site and that a user is able to click the link and navigate to reddit to view and interact with the thread.

Test Plan for Logging in:

- Our test plan is to show logging in works 100% of the time. We should have errors in the program when you attempt to gain access without logging in and creating an account. The user will test this by clicking all the links before logging in to check if they can gain access to the links. The user should be failing to access all the links pre logging in, but after logging in we should be able to gain access to the whole site and all of its features. We will also check a log of the specific Gmail login information of the users from firebase.

Test Plan for Checking Stocks:

- The first test will use Mocha/Chai to check the return data from Finnhub. This will check to make sure a valid response containing stock data has been returned from the API.
- Next a Mocha/Chai test will be made to post user stock buy/sell orders to the database. This will verify that the data was correctly written to the database.
- For the last test a user will navigate to the buy/sell page of the app. They will then buy or sell some stocks. They will navigate to the portfolio page and then verify that the information has updated correctly

-
```
it("Checks data incoming from Finnhub", done => {
  chai
    .request(server)
    .get("/finnhub/stock")
    .end((err, res) => {

      expect(res.body).to.be.an('array').that.is.not.empty; // check to make sure its returning array
      expect(res.body).to.have.property('c');
      expect(res.body).to.have.property('h');
      expect(res.body).to.have.property('o');
      expect(res.body).to.have.property('l'); // check for all the required parts of the stock
      done();
    });
```
-
```
it("Test for adding to firebase", done => {
  chai
    .request(server)
    .post("/firebase")
    .send({name: 'test'})
    .send({stock: 'APPL'})
    .send({Price: 1000})
    .end((err, res) => {

      expect(res.body).to.have.property('name',"test");
      expect(res.body).to.have.property('totalvalue',1000);
      expect(res.body).to.have.property('totalstocks',1);


      done();
    });
```

Test Plan for Being able to open wall street links:

- When you type in the stock ticker symbol, we are given the top reddit links associated with that ticker. The user should be able to test by typing in a ticker symbol and getting the correct associate reddit threads.
- The test case will fail when the ticker symbol does not give us the correct corresponding reddit thread.

- We will use Mocha/Chai to check the return data from reddit. This will check to make sure a valid response containing post information has been returned from the API.

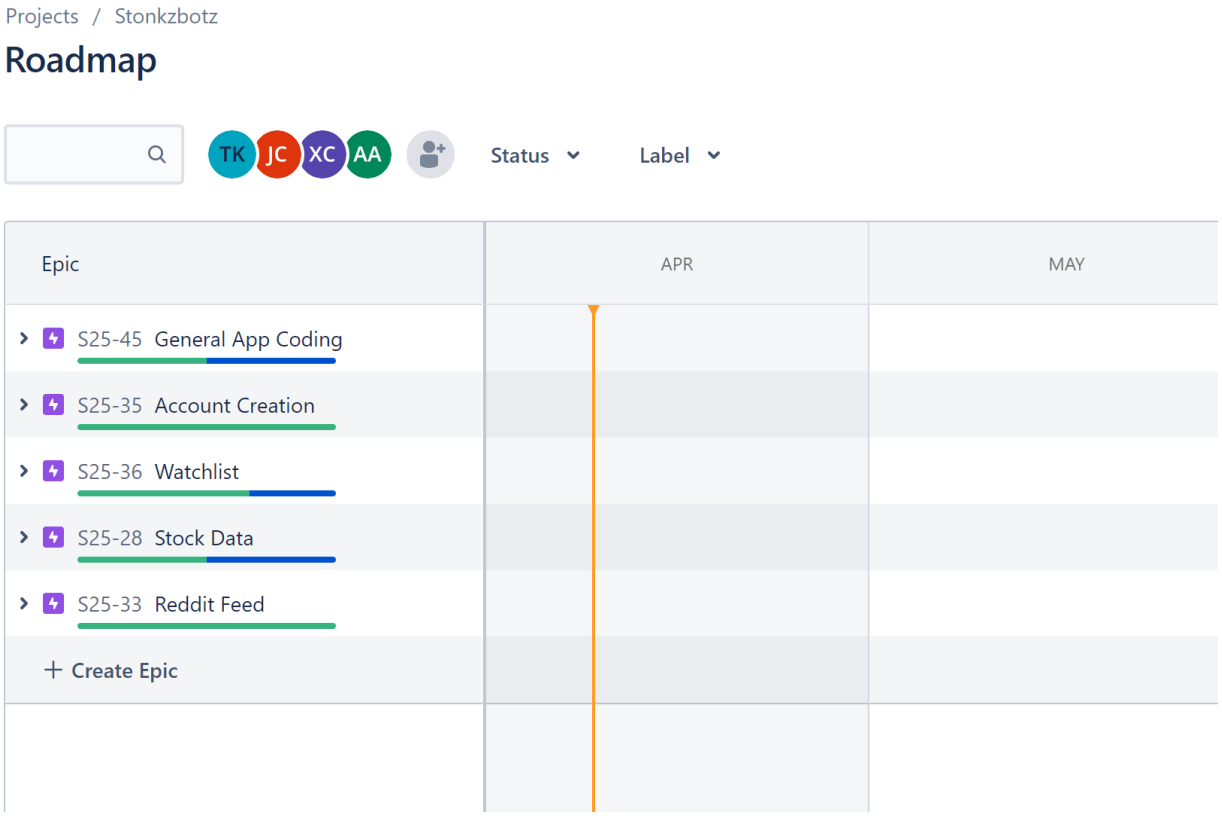
Individual contributions:

Avery and Thomas did this milestone.

Avery's latest commit
https://github.com/CSCI-3308-CU-Boulder/3308SP21_024_5/tree/main/Stonkz%20HTML/Demo

Thomas's latest commit:
https://github.com/CSCI-3308-CU-Boulder/3308SP21_024_5/commit/0ffe09caf480127da364a7862d62526d8c8f29e3

The Jira is posted below:



Projects / Stonkzbotz

Backlog

TK JC XC Epic

Finalize 9 Apr – 14 Apr (4 issues) 0 0 0 Complete sprint •••

3/30 - 4/11 Implement and fix anything outstanding

☑	S25-32 Buy/sell Stocks STOCK DATA	IN PROGRESS	JC
☑	S25-47 Finalize design GENERAL APP CODING	IN PROGRESS	XC
☑	S25-49 Bug fixes GENERAL APP CODING	IN PROGRESS	TK
☑	S25-53 HTML for watchlist WATCHLIST	IN PROGRESS	JC

+ Create issue