Sprint 2 Plan

Product Name: Backtesting Engine, Team Name: Simple Strategies
Sprint Completion Date: 11/3/25, Revision Number: 1, Revision date: 10/21/25

Goal:
Develop a stock strategy platform that allows users to understand stocks, create trading strategies, and evaluate their performance through backtesting. The system will integrate APIs for stock data retrieval, provide intuitive frontend interactions (such as popups and forms), and display analytical results including profit/loss metrics and performance charts.

User stories:

- As a user, I want to see a login/sign-up page so that I can start learning and creating strategies
    - Create a login/sign-up page. (2 hours)
    - Add user authentication so that an email cannot be used multiple times, creating less bloat in users to store. (2 hours)
    - Write a test to confirm the API returns correct data and handles errors (1 hour)
    - Total for user story (5 hours)
- As a user, I want to have a highly engaging way to learn and create stock strategies.
    - Create a Figma for portfolio containing a learn, create, and basic profile information, with a dashboard on the side, all to be implemented (1.5 hr)
    - Design the learn levels in Figma (3 hours)
    - Implement the learn level to the website (6 hours)
    - Write a test to confirm the API returns correct data and handles errors (1 hour)
    - Total for user story (4.5 hours)
- As a user, I want to view the results (profit/loss and performance metrics) after running a saved strategy so that I can compare outcomes.
    - Run a backtest using the selected strategy (fetch stock data and calculate gain/loss). (3 hrs)
    - Create an api that returns a summary (3 hrs)
        - Total Gain/Loss (%)
        - Key metrics (e.g., number of trades, max drawdown)
        - A graph showing performance over time
    - Display the results in the frontend as numbers and a simple line chart. (1 hr)

- Total for user story (7 hours)

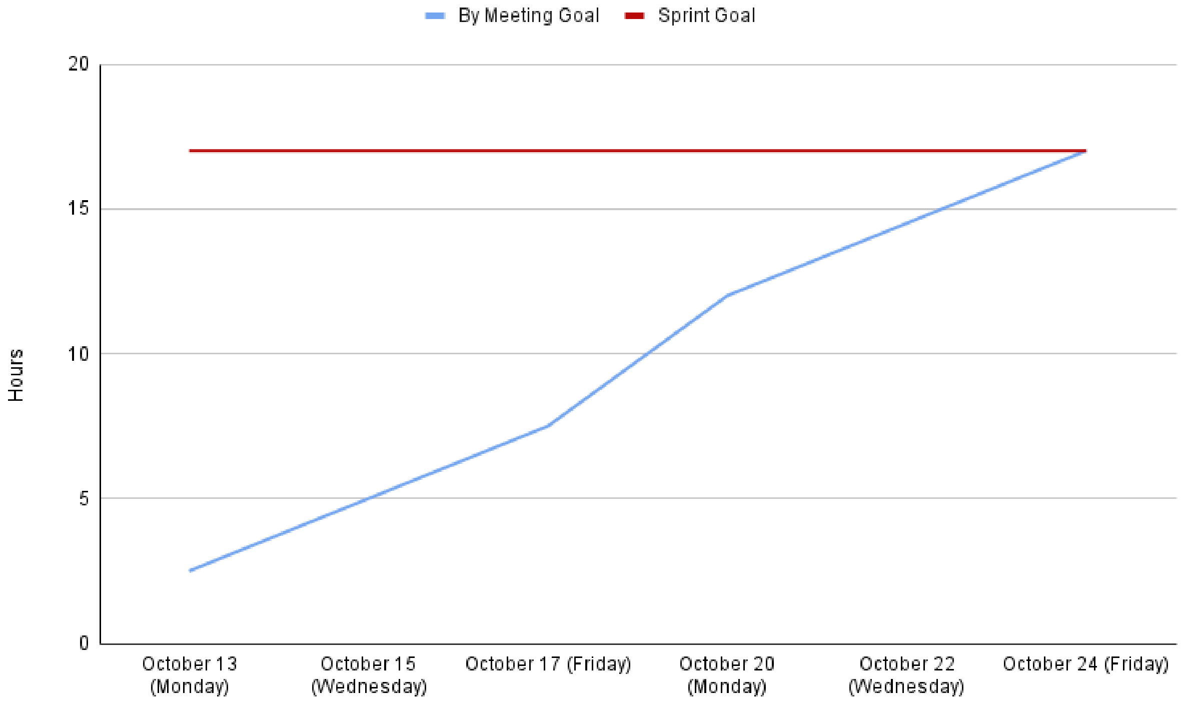
Team Roles:

- David Garcia : PO
- Nathan Nham: Scrum Master, Database Manager
- Nishan Lama: UI/UX designer, Backend developer
- Sanjana Manikandan: UI/UX designer, Backend developer
- Rohan Shukla: Frontend developer

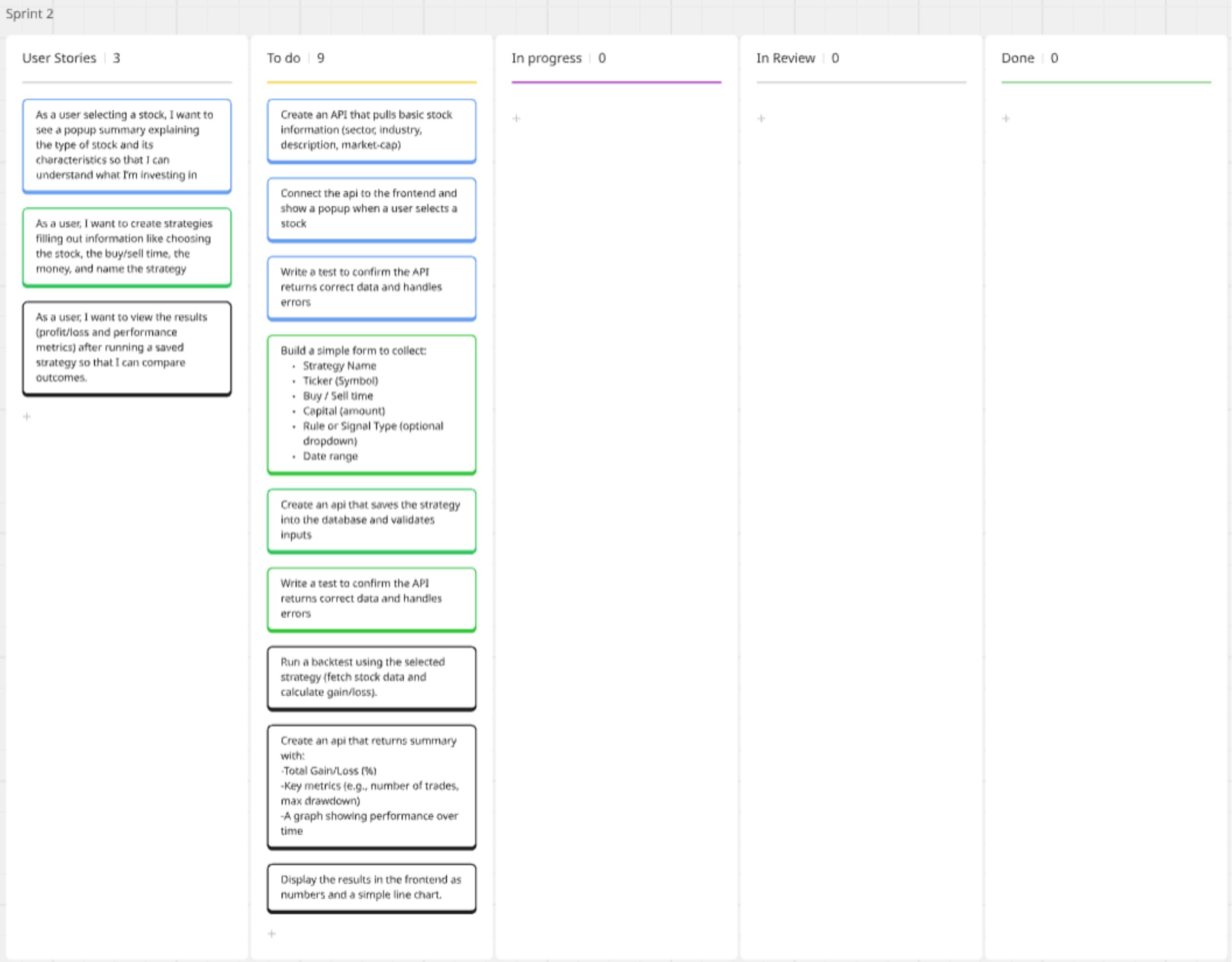
Initial Task Assignment:

- David Garcia: As a user, I want to view the results (profit/loss and performance metrics) after running a saved strategy so that I can compare outcomes, run a backtest using the selected strategy (fetch stock data and calculate gain/loss).
- Nathan Nham: As a user, I want to create strategies, filling out information like choosing the stock, the buy/sell time, the money, and name the strategy, create an api that saves the strategy into the database, and validates inputs
- Nishan Lama: As a user selecting a stock, I want to see a pop-up summary explaining the type of stock and its characteristics so that I can understand what I'm investing in, Create an API that pulls basic stock information (sector, industry, description, market-cap)
- Sanjana Manikandan: As a user, I want to create strategies filling out information like choosing the stock, the buy/sell time, the money, and name the strategy, build a simple form to collect strategy name, ticker (Symbol), Buy / Sell time, Capital (amount), Rule or Signal Type (optional dropdown), and date range
- Rohan Shukla: As a user, I want to view the results (profit/loss and performance metrics) after running a saved strategy so that I can compare outcomes, display the results in the frontend as numbers, and a simple line chart.

Initial Burnup Chart:



Initial Scrum Board:



Sprint 2

User Stories | 3

As a user selecting a stock, I want to see a popup summary explaining the type of stock and its characteristics so that I can understand what I'm investing in

As a user, I want to create strategies filling out information like choosing the stock, the buy/sell time, the money, and name the strategy

As a user, I want to view the results (profit/loss and performance metrics) after running a saved strategy so that I can compare outcomes.

To do | 9

Create an API that pulls basic stock information (sector, industry, description, market-cap)

Connect the api to the frontend and show a popup when a user selects a stock

Write a test to confirm the API returns correct data and handles errors

Build a simple form to collect:
- Strategy Name
- Ticker (Symbol)
- Buy / Sell time
- Capital (amount)
- Rule or Signal Type (optional dropdown)
- Date range

Create an api that saves the strategy into the database and validates inputs

Write a test to confirm the API returns correct data and handles errors

Run a backtest using the selected strategy (fetch stock data and calculate gain/loss).

Create an api that returns summary with:
-Total Gain/Loss (%)
-Key metrics (e.g., number of trades, max drawdown)
-A graph showing performance over time

Display the results in the frontend as numbers and a simple line chart.

In progress | 0

In Review | 0

Done | 0

Scrum Meeting Times:

- Mondays, Wednesdays, and Fridays from 8:45 to 9:00pm
- TA Meeting Time: Tuesdays from 2:45 to 3:30p

Backlog:

- Design create page
- Save user strategies (keep strategies per acc limited to 5)
- Design Levels one-four curriculum
- Design create page