Sprint 3 Plan

Product Name: Backtesting Engine, Team Name: Simple Strategies
Sprint Completion Date: 11/18/25, Revision Number: 1, Revision date: 11/4/25

Goal:
Finish mvp: Have a website that teaches users about stock fundamentals progressively. The user should be able to log in, sign up, and be redirected to their profile with the option to view their current level of fluency in finance/stock strategy fundamentals and a clickable learn button with exercises, quizzes, and interactive puzzles that make an engaging learning experience.

User stories:

- As a user learning more about strategies, I want to be able to visualize the results of a buy-and-hold strategy on a certain stock
    - Create a graph that the user can select the ticker, buy, and sell dates (4 hours)
    - Create a model that shows the results (gain/loss) using yfinance(3 hours)
    - Learning yfinance to fetch stocks (3 hours)
    - Add three basic strategies (B&H, DCA, SMAC) and a page to house them (14 hours)
- As a user, I want to be able to progress through levels and see my progress on my profile
    - Write level APIs to store user learning progress (3 hours) - Nathan (done)
    - Implement the 4 levels of frontend (7 hours)
        - Create a visual map that matches the level the user is on (2 hours)
            - Visual map (1 hour)
            - Connect the backend to the frontend (1 hour)
        - Create a previous and next button that navigate the users through the levels (1 hour)
- As a user, I want to learn about stocks and finance fundamentals via an interactive and engaging learning experience
        - Level 0 (17 hours)
            - Write key terms/definitions with responsiveness (5 hours)
            - Make the UI look like the Figma (6 hours)
                - Add picture and text
                - Overall Design
            - Create a data model to fill in the quiz and answers(2 hours)
            - Create a drag-and-drop for the quiz (2 hours)
            - Create the chart for the user to read and the corresponding quiz (2 hours)

- As a user, I want to learn about stocks and finance fundamentals via an interactive and engaging learning experience
        - Level 1 - Nishan (16 hours)
            - Create the vocab descriptions (1 hour)
            - Create a drag-and-drop for the quiz (3 hours)
            - Create a mini functionality for the user to select buy-and-hold on a map (5 hours)
            - Create the chart for the user to read and the corresponding quiz (2 hours)
            - Create a data model to fill in the quiz and answers (2 hours)
            - Make the UI look like the Figma (3 hours)
        - Connect the frontend to the backend APIs (2 hours)

Team Roles:

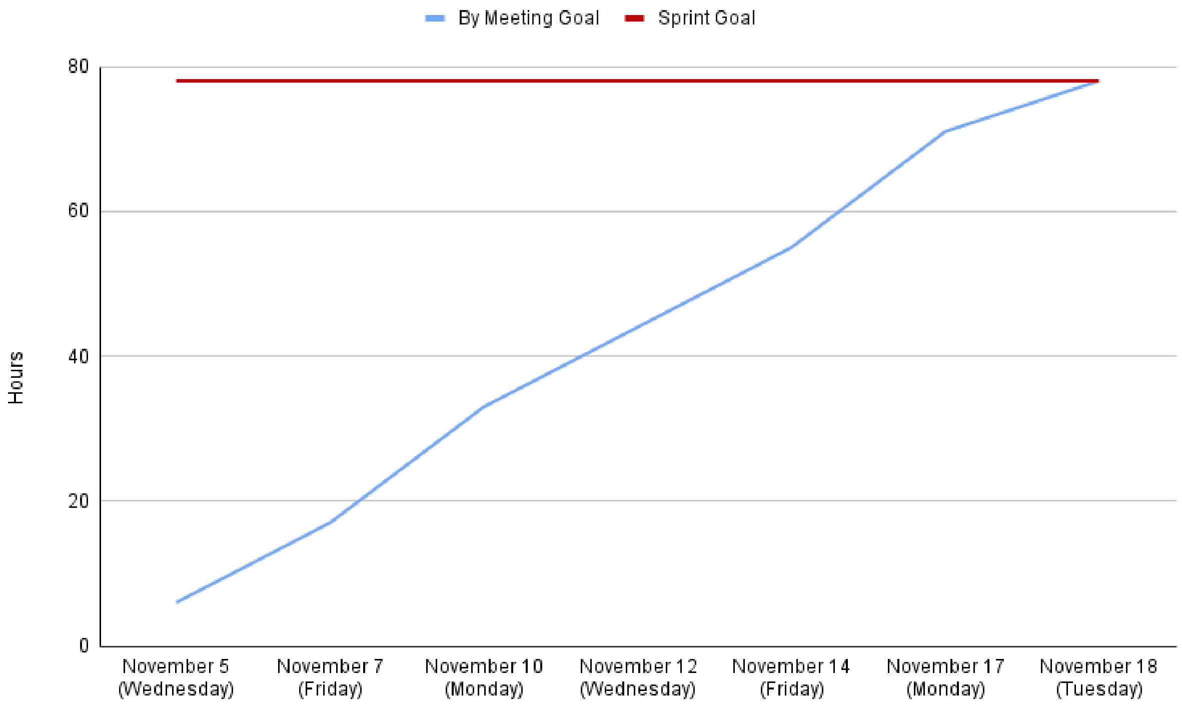
- David Garcia : PO
- Nathan Nham: Scrum Master, Database Manager
- Nishan Lama: UI/UX designer, Frontend developer
- Sanjana Manikandan: UI/UX designer, Frontend developer
- Rohan Shukla: Frontend developer

Initial Task Assignment:

- Nathan: As a user learning more about strategies, I want to be able to visualize the results of a buy-and-hold strategy on a certain stock
    - Create a model that accurately shows the results (gain/loss) using yfinance(3 hours)
    - Learning yfinance to fetch stocks (3 hours)
- David: As a user learning more about strategies, I want to be able to visualize the results of a buy-and-hold strategy on a certain stock
    - Create a graph that the user can select the ticker, buy, and sell dates (4 hours)
    - Create level 2 and 3 figma designs
- Rohan: As a user, I want to be able to progress through levels and see my progress on the portfolio
    - Level 0
        - Write key terms/definitions with responsiveness (5 hours)
        - Make the UI look like the Figma (6 hours)
            - Add picture and text
            - Overall Design

        - Create a data model (2 hours)
        - Create a drag-and-drop for the quiz (2 hours)
        - Create the chart for the user to read and the corresponding quiz (2 hours)
    - Create a visual map that matches the level the user is on (2 hours)
- Nishan: As a user, I want to be able to progress through levels and see my progress on the portfolio
    - Level 1 - Nishan (16 hours)
        - Create the vocab descriptions (1 hour)
        - Create a drag-and-drop for the quiz (3 hours)
        - Create a mini functionality for the user to select buy-and-hold on a map (5 hours)
        - Create the chart for the user to read and the corresponding quiz (2 hours)
        - Create a data model (2 hours)
        - Make the UI look like the Figma (3 hours)
    - Connect the frontend to the backend APIs (2 hours)
    - Create a previous and next button that navigate the users through the levels (1 hour)
- Sanjana: As a user, I want to be able to progress through levels and see my progress on the portfolio
    - Create levels 2 and 3 Figma designs
    - Create a graph that the user can select the ticker, buy, and sell dates (4 hours)

Initial Burnup Chart:



Initial Scrum board:

User Stories | 2

As a user learning more about strategies, I want to be able to visualize the results of a buy-and-hold strategy on a certain stock

As a user, I want to be able to progress through levels and see my progress on the portfolio

+

To do | 5

Build a simple form to collect: Strategy NameTicker (Symbol)Buy / Sell timeCapital (amount)Rule or
david

Run a backtest using the selected
Nathan Nham

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
Nathan Nham

Display the results in the frontend as
david

implementing frontend roadmap
Nishan Lama

+

In progress | 15

Plan contents for create strategy page
Nathan Nham

connect backend apis to frontend
- Increment user lesson
- get user lesson progress
- add learning user
Nishan Lama

Level 1
- Create the vocab description
Nishan Lama

Level 0
- Create the vocab description
Rohan Shukla

L1
- Create a drag-and-drop for the quiz
Nishan Lama

Create a drag-and-drop for the quiz
Nishan Lama

Create a drag-and-drop for the quiz
Rohan Shukla

- Create the functionality for the user to test buy-and-hold (2 hours)
Nishan Lama

- Create the chart for the user to read and the corresponding quiz (2 hours)
Nishan Lama

Design learn level 2 and 3
David Samanika

Create a graph that the user can select the ticker, buy, and sell dates (4 hours)
Nathan david

Create a model that shows the results (gain/loss) using yfinance(3 hours)
David Nathan Nham

Create a visual map that matches the level the user is on (2 hours)
Nishan Rohan Shukla

Create a previous and next button that navigate the users through the levels (1 hour)
Nishan Lama

Fix portfolio page sizing and responsiveness
Rohan Shukla

+

In Review | 0

+

Done | 1

Learning yfinance to fetch stocks (3 hours)
Nathan Nham

+

miro

Scrum Meeting Times:

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

Backlog:

- Create a graph that the user can select the ticker, buy, and sell dates (4 hours)
- Create a model that shows the results (gain/loss) using yfinance(3 hours)
- Learning yfinance to fetch stocks (3 hours)
- Write level APIs to store user learning progress (3 hours) - Nathan (done)
- Implement the 4 levels of frontend (7 hours)
- Create a visual map that matches the level the user is on (2 hours)
  - Visual map (1 hour)
  - Connect the backend to the frontend (1 hour)
- Create a previous and next button that navigate the users through the levels (1 hour)
- Write key terms/definitions with responsiveness (5 hours)
- Make the UI look like the Figma (6 hours)
  - Add picture and text
  - Overall Design
- Create a data model to fill in the quiz and answers(2 hours)
- Create a drag-and-drop for the quiz (2 hours)
- Create the chart for the user to read and the corresponding quiz (2 hours)
- Create the vocab descriptions (1 hour)
- Create a drag-and-drop for the quiz (3 hours)
- Create a mini functionality for the user to select buy-and-hold on a map (5 hours)
- Create the chart for the user to read and the corresponding quiz (2 hours)
- Create a data model to fill in the quiz and answers (2 hours)
- Make the UI look like the Figma (3 hours)
- Connect the frontend to the backend APIs (2 hours)