

F1 Fantasy Investment Simulator - Documentation

1. User Guide

1.1 Introduction

The F1 Fantasy Investment Simulator is a web application that allows users to simulate investing in Formula 1 drivers as if they were stocks. Users can buy and sell shares of drivers based on their fluctuating market prices and track their portfolio performance.

1.2 How to use the Application

1. Market Page

- a. View available F1 drivers and their stock prices.
- b. Click “Buy” to purchase share(s) of a driver
- c. Click “Sell” or “Sell All” to sell shares of a driver
- d. Every user is given \$50,000 as the starting balance

2. Portfolio Page

- a. View your owned shares and investment details
- b. Check total invested amount, average price and unrealized P/L (Profit and Loss)
- c. Track Portfolio value and rate of return through a live chart

3. Leaderboard page

- a. Displays the best-performing portfolios
- b. Compares user investment performance against the stocks they bought into

4. Reset Game

- a. Click the “Reset Game” button on the top right corner of the application
- b. This will erase all data and restart the simulation

2. Project Overview

2.1 What the application does

The F1 Fantasy Investment Simulator allows users to:

- Buy and sell shares of F1 drivers based on dynamic pricing
- Track portfolio performances with real-time updates
- View price changes and market trends over time
- Check stock performance against each other on the leaderboard

2.1 Key Features

- **Dynamic Pricing:** Driver prices fluctuate, mimicking a real stock market scenario
- **Fractional Shares:** Users can buy partial shares of expensive drivers
- **Portfolio Tracker:** A live graph updates portfolio performance
- **Leaderboard:** Compares top investments in the simulation
- **Reset Functionality:** Users can reset and restart at any time

3. Good Endpoints (Key Functionalities)

3.1 Buying Shares

- Users can buy fractional shares based on available balance
- Example function in market.js:

```
function buyDriver(driverName, amount) {  
  let driver = drivers[driverName];  
  let price = driver.currentPrice;  
  let sharesBought = amount / price;  
  updatePortfolio(driverName, sharesBought, amount);  
}
```

3.2 Selling Shares

- Users can sell specific amounts or all shares of a driver

- Example function in market.js

```
function sellDriver(driverName, amount) {  
    let driver = portfolio[driverName];  
    let price = getDriverPrice(driverName);  
    updatePortfolioAfterSale(driverName, amount, price);  
}
```

3.3 Portfolio Tracker Chart

- Updates after each transaction to display value changes
- Example function in portfolio.js

```
function updatePortfolioChart() {  
    let history = loadPortfolioHistory();  
    portfolioChart.data.datasets[0].data = history;  
    portfolioChart.update();  
}
```

3.4 Reset Functionality

- Clears all stored user data and restarts the simulation
- Example function in global.js

```
function resetGame() {  
    localStorage.clear();  
    location.reload();  
}
```

4. Improvements

4.1 Future enhancements

- **More Drivers and Teams:** Expand the selection to all F1 Teams, maybe the inclusion of ETF (exchange traded funds) based on drivers in F1 Teams, Driver nationality and/or world champions/race winners.
- **Real F1 Data Integration:** Connect to real-world F1 performance metrics
- **Live Market Updates:** Prices fluctuate based on external data sources.
- **User Authentication:** Allow users to create accounts and save progress.
- **Social Features:** Let users compete and share performance stats.
- **Addition of different motorsport drivers:** Have a variety of drivers from different motorsport categories such F2, F3, NASCAR, WEC etc.

4.2 Challenges Faced

- **Handling Price Fluctuations:** Ensuring prices change in a balanced way
- **Preventing Portfolio Resets:** Making sure data is persistent across sessions
- **User Experience:** Optimizing UI for an engaging investment simulation

5. Conclusion

The F1 Fantasy Investment Simulator provides an engaging way for users to experience a stock market-like environment using F1 drivers. While functional, future enhancements such as data integration and user authentication could greatly improve the project.