



Mini-Project Report
On
Stock Trading Simulation

Submitted by:
Nikita Pursani
Nilesh Rathod
Yogesh Chavhan

Computer Science and Engineering

2020-2021

Under the Guidance of
Prof. Y. S. Alone



Department of Computer Science & Engineering,
Prof. Ram Meghe Institute of Technology &
Research, Badnera-Amravati.

2020-2021

Vidarbha Youth Welfare Society's
Prof. Ram Meghe Institute of Technology & Research
Badnera, Amravati (M.S.) 444701



CERTIFICATE

Certified that the mini-project work entitled **“Stock Trading Simulation”**
is a bona fide work carried out by

Ms. Nikita Pursani
Mr. Nilesh Rathod
Mr. Yogesh Chavhan

of B.E. Third Year Semester-VI, Department of Computer Science
& Engineering, during the academic year 2020-21.

The report has been approved as it satisfies the academic requirements in
respect of mini-project work relevant for the course.

Prof. Y. S. Alone
Guide

Dr. G. R. Bamnote
H.O.D.

CONTENTS

Chapter No.	Title	Page No.
1	INTRODUCTION	
	1.1 Objectives	4
2	SYSTEM DESIGN	
	2.1 E-R Diagram	5
3	SYSTEM IMPLEMENTATION	
	3.1 Implementation	6
	3.2 Table Snapshots	7
	3.3 System Screen shots	8
4	CONCLUSION	
	4.1 Conclusion	12
	REFERENCES	

CHAPTER 1 - INTRODUCTION

1.1 OBJECTIVE

A stock Trading simulation is a program or application that attempts to reproduce or duplicate some or all features of a live stock market on a computer so that a player may practice trading stocks without financial risk. Paper trading (sometimes also called "Virtual Stock Trading") is a simulated trading process in which would-be investors can 'practice' investing without committing real money. This is done by the manipulation of imaginary money and investment positions that behave in a manner similar to the real markets. Before the widespread use of online trading for the general public, paper trading was considered too difficult by many new investors. Our aim is to create software that helps customers to perform Buy/Sell Stocks or trading through stock data of certain companies, with the help of certain parameters that affect stock value.

The stock market is often indicative of how resources are distributed and how prices are agreed as, for all 'stock', there is massive scope for buying and bargaining. This system is currently controlled by a number of traders who buy and sell units according to cost and demand. This is a very well-paid profession as it is assumed that to optimize allocative efficiency requires highly-skilled and knowledgeable persons, yet experiments have shown that when simulation is applied, using only simple rules and with no market knowledge, both artificial and human 'Zero Intelligence' agents will converge to an equilibrium where, not only is allocation efficiency high, but the percentage of transactions occurring is also great.

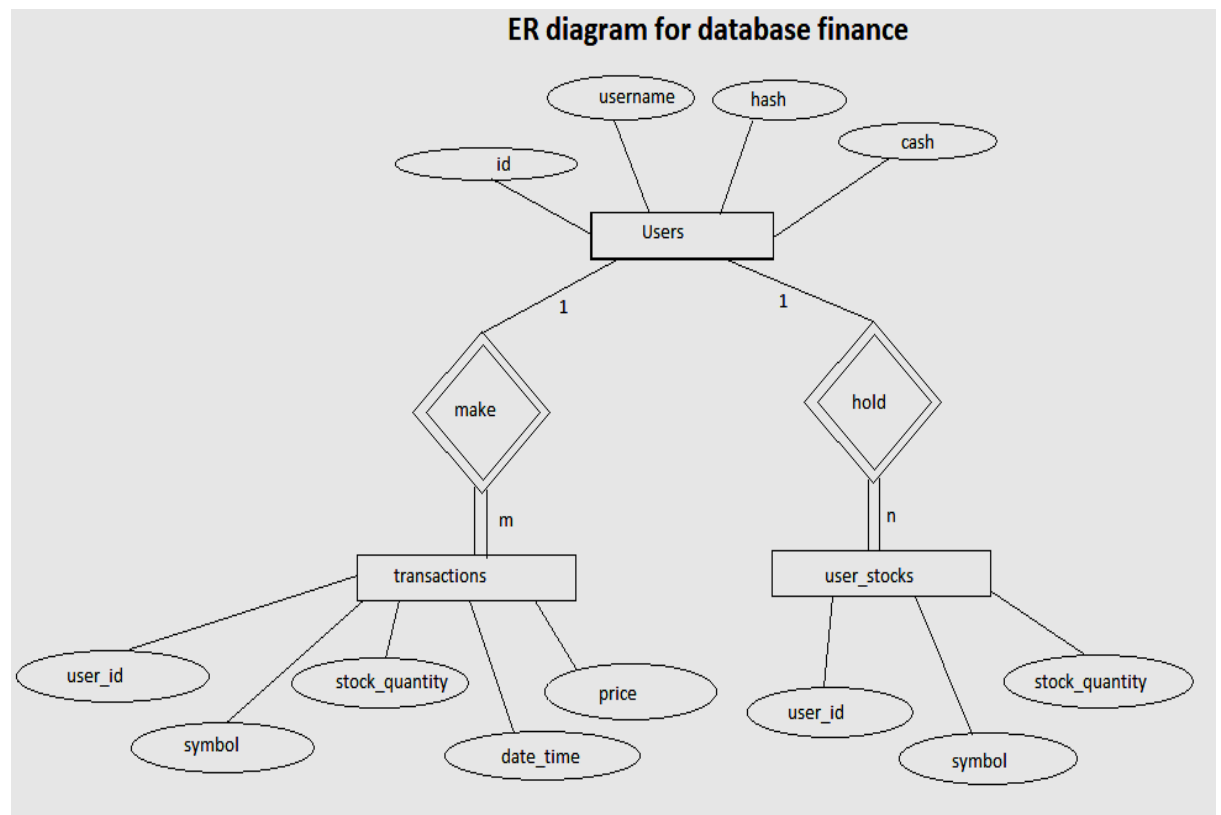
Database Management Systems (DBMS) are essential in supporting project tracking and control functions. A database provides a platform to organize, store and retrieve the planned and actual performance data of projects in a logical and efficient manner. The DBMS queries the stored project data using SQL (Structured Query Language) to generate different management reports for control purposes. It follows that the design of the database should follow a well-defined structure to support the tracking and control of individual tasks at different levels of reporting. The data structure should also facilitate the linkage of those individual tasks to their respective construction trades. A work package model is commonly used to describe the data structure of a project. The work package is a general expression that represents a well-defined scope of work that usually terminates in a deliverable product.

A website which simulates stock trading where users can log in and trade shares with real prices in real time and with not so real money. They can over time see their gains and losses.

- Built with Python - Flask framework (with an Jinja Template)
- Utilizes SQLite Database (local)
- Supports login sessions Has server-side validation on various levels
- Talks to IEX API to get real-time stock prices
- Includes a dashboard that allows users to add unlimited cash to their account \$. \$ and also, users can see their transaction history.

CHAPTER 2 – SYSTEM DESIGN

2.1 ER Diagram



CHAPTER 3 – SYSTEM IMPLEMENTATION

3.1 IMPLEMENTATION

3.1.1 SYSTEM DESIGN:

- **HTML** - It is used to create UI (User Interface). It helps the user to interact with the system.
- **Python** - Flask Framework (with Jinja template) is used for background processing.
- **SQLite** - It is used for suitable and reliable working to make smooth for Data Processing (for Fetching, Storing, deleting & deletion of data).

3.1.2 LIBRARY:

- **CS50** - It is open source and provides an IDE environment to develop a Flask and Django Framework compressed with HTML, CSS, JavaScript support of web utilities. Basically, Flask is used to interact with Web Technology using the Python language.
- **Flask** - Flask is a third party micro-framework allowed to work with web technology. It supports extensions that can add application features as if they were implemented in Flask itself.
- **Flask Session** - Flask-Session is an extension for Flask that support Server-side **Session** to your application. The Session is the time between the client logs in to the server and logs out of the server. The data that is required to be saved in the Session is stored in a temporary directory on the server.
- **IEX Cloud API** – It provides us data of a particular stock based on its ticket in JSON format. It returns nothing if the stock ticker is not valid i.e. a company with that stock ticker does not exist.

3.2 TABLE SNAPSHOTS

users Table:

finance → users

Browse	Structure	SQL	Search	Insert	Export	Import	Rename	Empty	Drop
--------	-----------	-----	--------	--------	--------	--------	--------	-------	------

			Column #	Field	Type	Not NULL	Default Value	Primary Key
<input type="checkbox"/>	Edit	Delete	0	id	INTEGER	Yes	None	Yes
<input type="checkbox"/>	Edit	Delete	1	username	TEXT	Yes	None	No
<input type="checkbox"/>	Edit	Delete	2	hash	TEXT	Yes	None	No
<input type="checkbox"/>	Edit	Delete	3	cash	NUMERIC	Yes	10000.00	No

Check All / Uncheck All With Selected: Delete ▼ Go

Add field(s) at end of table Go

Query used to create this table

```
CREATE TABLE 'users' ('id' INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, 'username' TEXT NOT NULL, 'hash' TEXT NOT NULL, 'cash' NUMERIC NOT NULL DEFAULT 10000.00 )
```

transactions Table:

finance → transactions

Browse	Structure	SQL	Search	Insert	Export	Import	Rename	Empty	Drop
--------	-----------	-----	--------	--------	--------	--------	--------	-------	------

			Column #	Field	Type	Not NULL	Default Value	Primary Key
<input type="checkbox"/>	Edit	Delete	0	user_id	INTEGER	Yes	None	No
<input type="checkbox"/>	Edit	Delete	1	symbol	TEXT	Yes	None	No
<input type="checkbox"/>	Edit	Delete	2	stock_quantity	INTEGER	Yes	None	No
<input type="checkbox"/>	Edit	Delete	3	date_time	TEXT	Yes	None	No
<input type="checkbox"/>	Edit	Delete	4	price	INTEGER	Yes	0.00	No

Check All / Uncheck All With Selected: Delete ▼ Go

Add field(s) at end of table Go

Query used to create this table

```
CREATE TABLE 'transactions' ('user_id' INTEGER NOT NULL, 'symbol' TEXT NOT NULL, 'stock_quantity' INTEGER NOT NULL, 'date_time' TEXT NOT NULL, 'price' INTEGER NOT NULL DEFAULT 0.00, FOREIGN KEY (user_id) REFERENCES users (id) )
```

users_stocks Table:

finance → users_stocks

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Rename](#) [Empty](#) [Drop](#)

			Column #	Field	Type	Not NULL	Default Value	Primary Key
<input type="checkbox"/>	Edit	Delete	0	user_id	INTEGER	Yes	None	No
<input type="checkbox"/>	Edit	Delete	1	symbol	TEXT	Yes	None	No
<input type="checkbox"/>	Edit	Delete	2	stock_quantity	INTEGER	Yes	None	No

[Check All](#) / [Uncheck All](#) With Selected: [Delete](#) [Go](#)

Add field(s) at end of table [Go](#)

Query used to create this table

```
CREATE TABLE 'users_stocks' ( 'user_id' INTEGER NOT NULL, 'symbol' TEXT NOT NULL, 'stock_quantity' INTEGER NOT NULL, FOREIGN KEY (user_id) REFERENCES users (id))
```

3.3 SYSTEM SCREENSHOTS

Login Screen:

C\$50 Finance

Register Log In

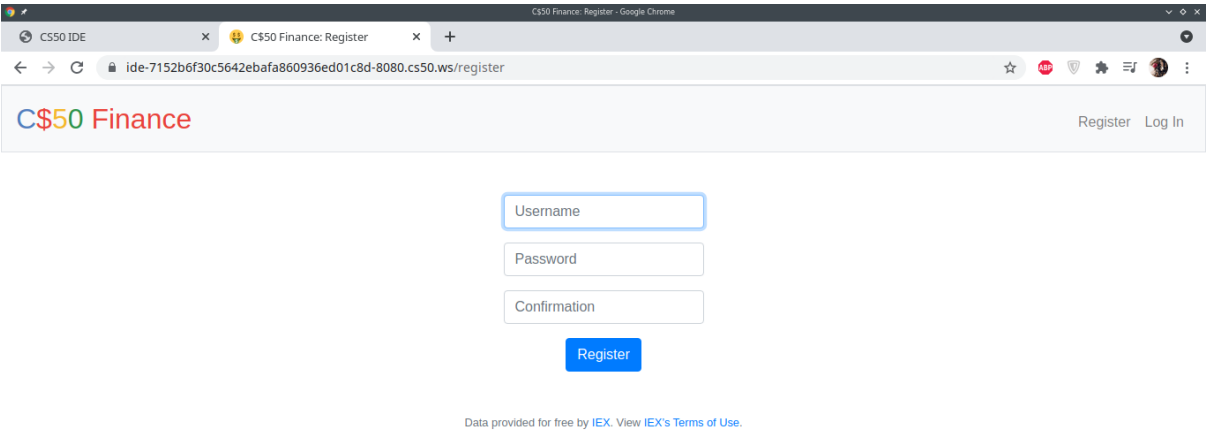
Username

Password

Log In

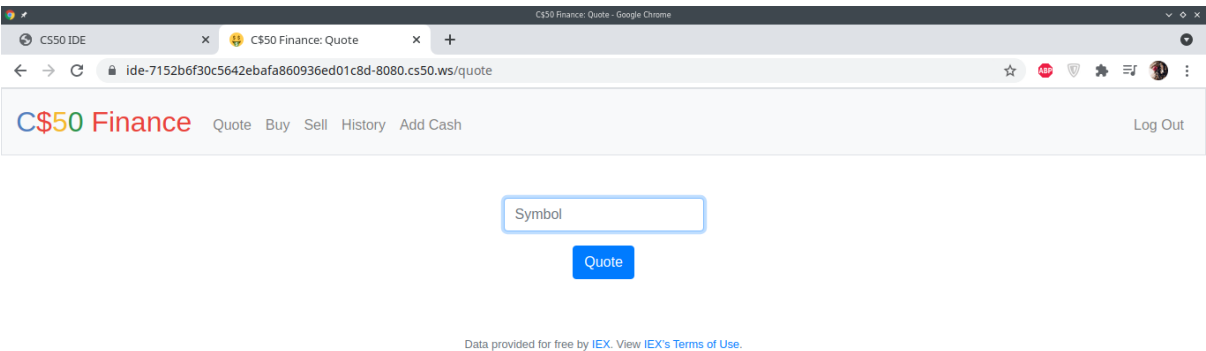
Data provided for free by IEX. View IEX's Terms of Use.

Create Account Screen:



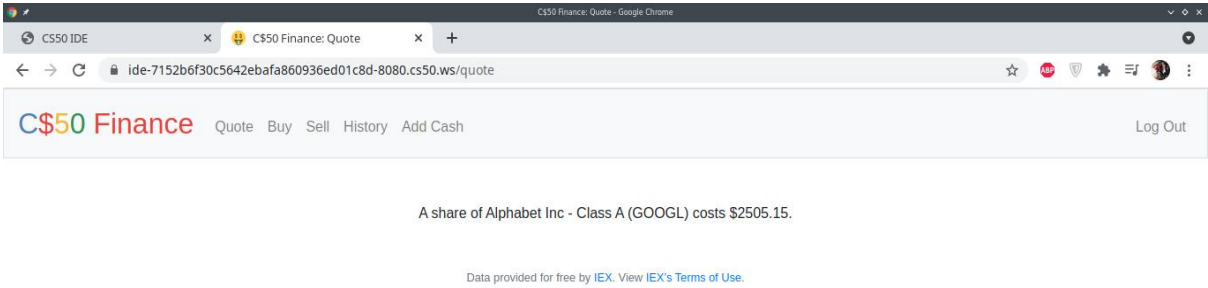
A screenshot of a web browser showing the 'C\$50 Finance: Register' page. The browser's address bar displays the URL 'ide-7152b6f30c5642ebafa860936ed01c8d-8080.cs50.ws/register'. The page header includes the 'C\$50 Finance' logo on the left and 'Register' and 'Log In' links on the right. The main content area features three input fields labeled 'Username', 'Password', and 'Confirmation', stacked vertically. Below these fields is a blue 'Register' button. At the bottom of the page, a small line of text reads: 'Data provided for free by IEX. View IEX's Terms of Use.'

Checking Share Price Screen:

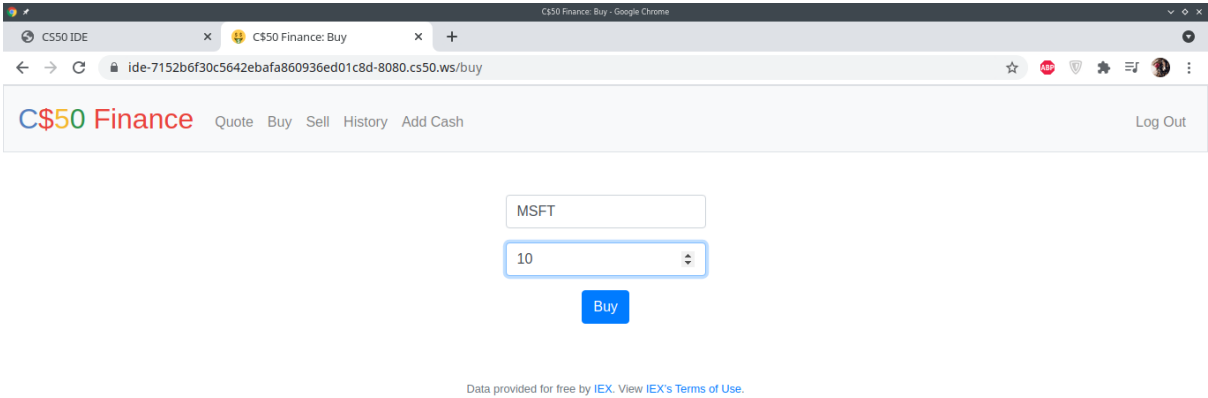


A screenshot of a web browser showing the 'C\$50 Finance: Quote' page. The browser's address bar displays the URL 'ide-7152b6f30c5642ebafa860936ed01c8d-8080.cs50.ws/quote'. The page header includes the 'C\$50 Finance' logo on the left, a navigation menu with 'Quote', 'Buy', 'Sell', 'History', and 'Add Cash' in the center, and a 'Log Out' link on the right. The main content area features a single input field labeled 'Symbol'. Below this field is a blue 'Quote' button. At the bottom of the page, a small line of text reads: 'Data provided for free by IEX. View IEX's Terms of Use.'

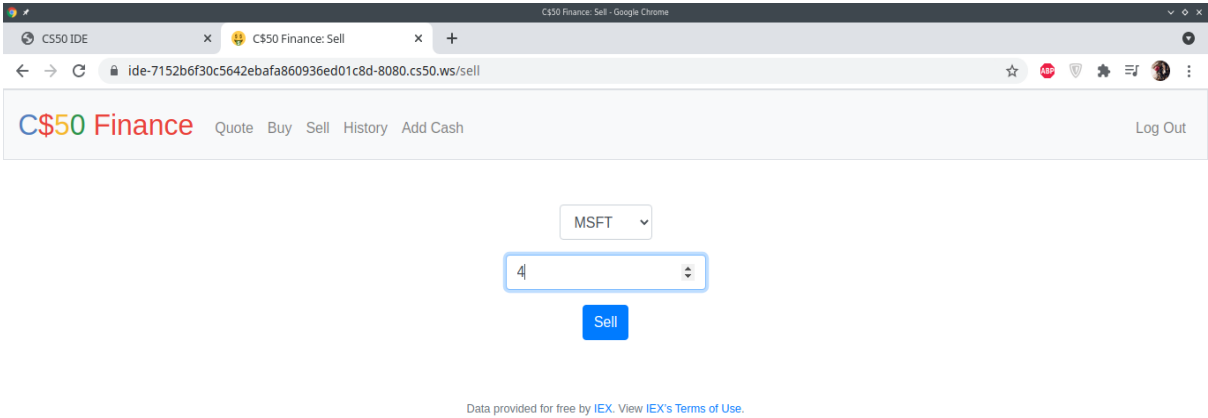
Share Price Screen:



Buying Share Screen:



Selling Share Screen:



Adding Cash Screen:

History Screen:

Symbol	Name	Shares	Price	TOTAL
MSFT	Microsoft Corporation	12	\$277.65	\$3331.80
GOOGL	Alphabet Inc - Class A	3	\$2505.15	\$7515.45
AAPL	Apple Inc	3	\$139.96	\$419.88
CASH				\$2456.14
				\$13723.27

CHAPTER 4 – CONCLUSION

4.1 CONCLUSION

Stock trading is a skill many want to practice in real time. It's impossible to trust yourself with your money or someone else's money as a novice without experience. Hence, this web application is of paramount importance to these individuals and we try our best to cater to their needs. This application lets an individual buy and sell stocks without using real money which is essentially the main ingredient which makes this project valuable to everyone interested in stocks but hesitates to risk their own money. However, this has to be done in real time otherwise this project will be of no

real value to anyone, which is done using IEX cloud API service, and this is the succeeding ingredient that adds value to this project.