

Stonk Exchange

Team 6: Juan Gutierrez, Kyle Absten, Harut Lementsyan

CST336-40 Summer 2023

Project Documentation

Project Description

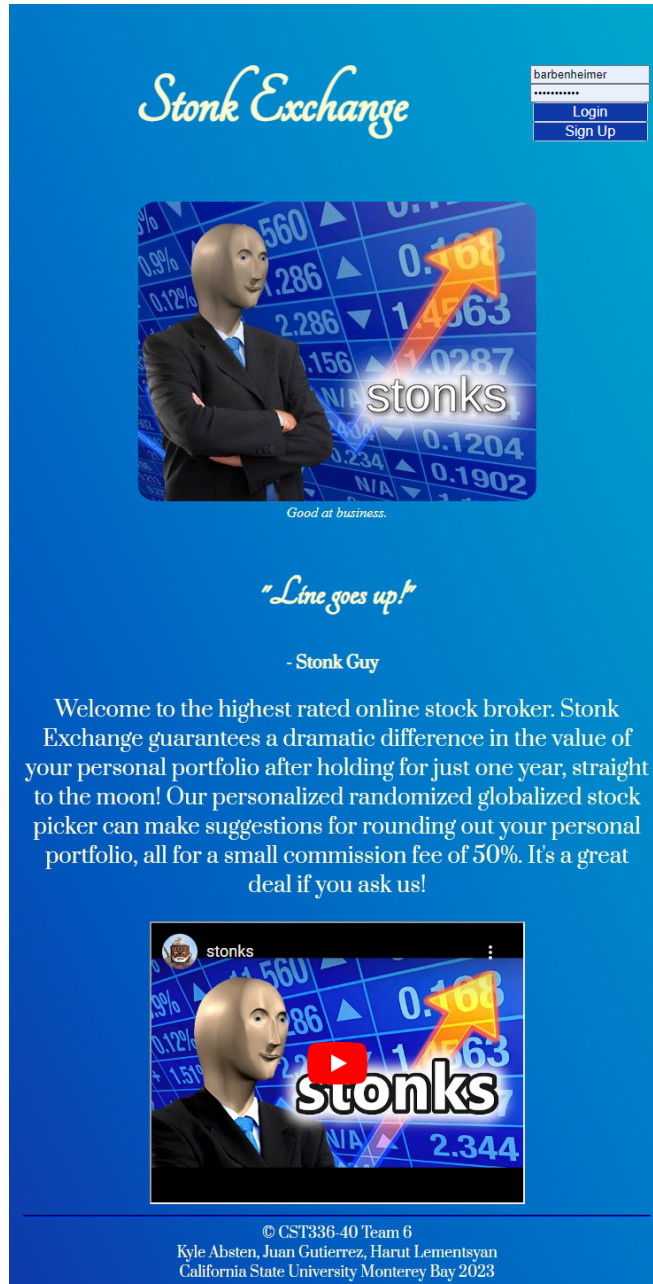
Our team decided to build an online stock broker website in order to meet the requirements for the final project. This is a tongue-in-cheek project that would manage user data, stock market data, and user transactions as a thinly-veiled financial scam. This grift includes stock search and purchase functionality, a dashboard with news stories related to each user's portfolio, a breakdown of said portfolio by number of shares and amount invested, a stock-picker "AI", and a user profile edit page, using Sessions for a persistent user experience across the site. All of this is linked on the back end to an SQL database to store user-generated data. We used two APIs to fetch stock market data. Finnhub.io was used for general stock information including ticker names, value, company information, and news stories related to specific stock. The second API, Senate Stock Watcher, was used to feed our "AI" stock picker the most recent verified stock purchases by sitting US Senators and their spouses. Finally, we included an easter egg page accessible only by manually typing the "/loot" extension in the URL.

Task Distribution

At the suggestion of feedback from the Team Plan assignment, we made sure each member of the team got the chance to develop at least one full route from the database at the back end to the front-end user interface level. Additionally, each team member took on different primary responsibilities. Juan Gutierrez was team coordinator, and routinely kept the other members up to date on current work being done by each other as well as TODO items through discussions on our Discord chat, as well as writing much of the CSS code and front-end user interface. Kyle Absten worked on the bulk of back-end database management and storage, as well as implemented the Sessions functionality for the site and API interactions. Harut Lementsyan contributed to both, implementing the portfolio chart analysis, as well as performing thorough testing and debugging of the site.

Project Screenshots

Front Page

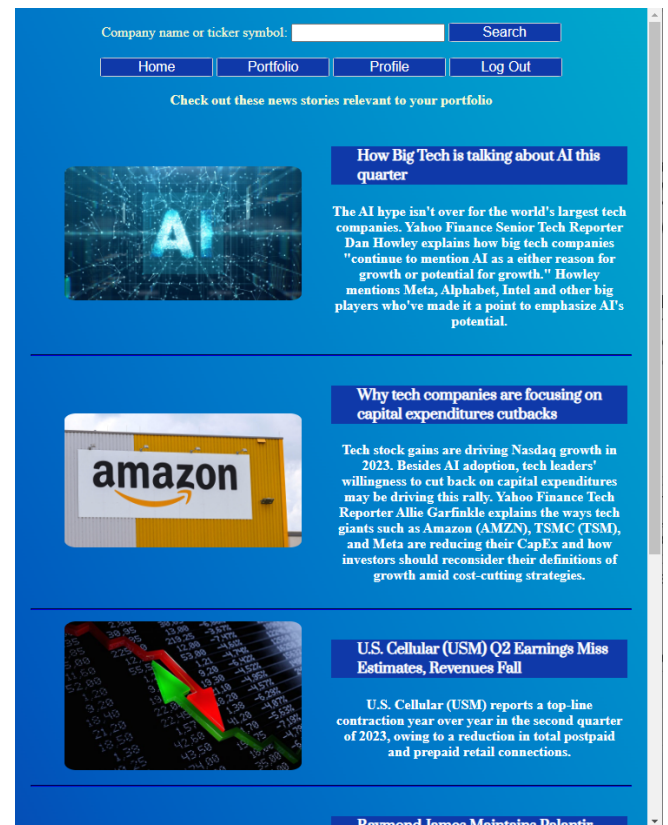


Home Dashboard links 3 recent news items related to each stock in a user's portfolio and shows the first instance of the search and navigation bar on the user side of the site at the top of the page.

Signup Page

Front page passes username and password to the Signup page in a friendly way. The submit button remains disabled until the fields are all validated.

Dashboard



Portfolio Page

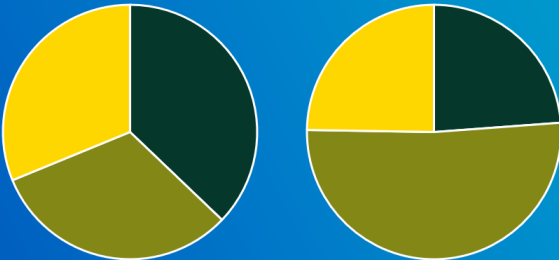
Company name or ticker symbol:

Portfolio

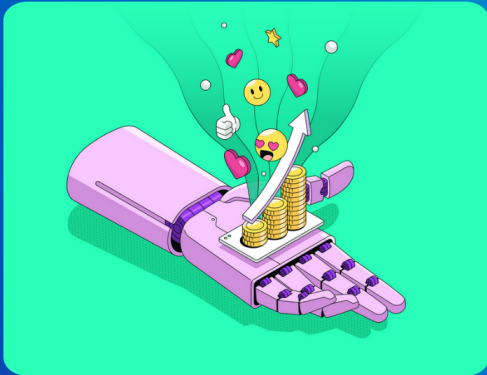
- F : 75 shares valued at \$966.00
- VZ : 64 shares valued at \$2084.48
- CLF : 63 shares valued at \$1002.96

Total Amount Invested: \$4053.44

☐ F ☐ VZ ☐ CLF ☐ F ☐ VZ ☐ CLF



Click to get a helpful suggestion from our robot overlord stock-picking savant Superintelligent Artificial General Intelligence Blockchain AI for a guaranteed increase in the total value of your portfolio investments!



© CST336-40 Team 6
Kyle Absten, Juan Gutierrez, Harut Lementsyan
California State University Monterey Bay 2023

Profile Edit Page

Company name or ticker symbol:

Profile Info

First Name:

Last Name:

Birthdate:

Sex:

Current Password:

New Password:

Re-type New password:

Credit Card (16 digits):


© CST336-40 Team 6
Kyle Absten, Juan Gutierrez, Harut Lementsyan
California State University Monterey Bay 2023

The user profile edit page prefills with information from the database.

The portfolio page displays the amount and current value of shares purchased by the user, along with a graphical breakdown of the data. It also prompts the user to try the AI stock picker tool, which sends them to a modified version of the search results page with suggested stocks instead of those related to the user query.

Search Page

Company name or ticker symbol:



Symbol: GME
Price at open: \$20.70/share
Current price: \$20.74/share

Buy GME

of shares to purchase:


At \$20.74/share, plus our small commission, brokerage fees and exchange rate adjustments (including federal, state and local taxes) it comes out to a mere \$1306.62! What a steal! Literal theft!

[GMET Website](#)

Symbol: GMET
Price at open: \$26.95/share
Current price: \$27.25/share

Buy GMET

of shares to purchase:



Symbol: GMER
Price at open: \$0.02/share
Current price: \$0.02/share

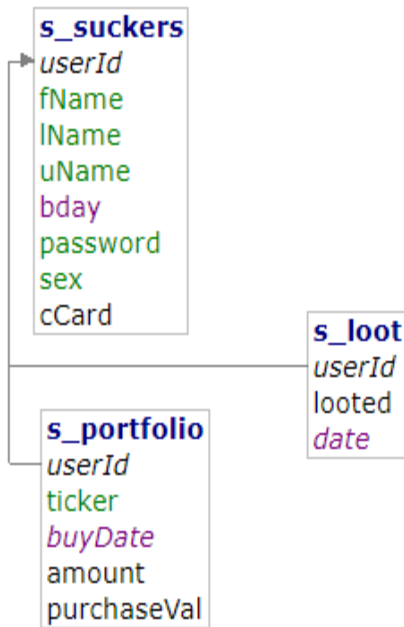
Buy GMER

of shares to purchase:

The search results page displays stocks related to the user query. The query entered for this screenshot was “GME”, which results in the matching ticker stock first, followed by related options. The image is a clickable link to the company website, which defaults to a hyperlinked indication when the stock does not come with an accompanying logo image. Next to the company logo the user will see the basic financial information for the stock, including the ticker name, open price, and current price per share. When the user inputs or changes the number of shares to purchase for any given stock among the results, a dedicated message will display under each one showing the total charge for the transaction, with the brokerage fees included, of course.

SQL Database

Schema and Description



Our database consists of three tables, linked by the `userId` from the database containing personal user information. Another table keeps track of each user's stock portfolio. Originally we did not have the stock's price at purchase "purchaseVal", it was added once we realized it would be difficult to retrieve historical data on stock values using the API around which we had built the site, so we added it to be able to track the individual progress of stocks in each user's portfolio. The final table registers our corporate profits, which include the bloated fees and commission we get from each purchase.

CREATE Statements

```
SET NAMES utf8;
SET time_zone = '+00:00';
SET foreign_key_checks = 0;
SET sql_mode = 'NO_AUTO_VALUE_ON_ZERO';
```

```
SET NAMES utf8mb4;
```

```
CREATE DATABASE `jruj86yjj0mdl6ko` /*!40100 DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */;
USE `jruj86yjj0mdl6ko`;
```

```
DROP TABLE IF EXISTS `s_loot`;
CREATE TABLE `s_loot` (
  `userId` int NOT NULL,
  `looted` float NOT NULL,
  `date` timestamp NOT NULL ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`userId`,`date`),
  CONSTRAINT `s_loot_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `s_suckers`
  (`userId`))
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `s_loot` (`userId`, `looted`, `date`) VALUES
```

```
(8, 5868.15, '2023-08-01 21:55:43'),  
(11, 102.83,'0000-00-00 00:00:00'),  
(11, 4497.08, '2023-07-29 23:53:39'),  
(11, 3432, '2023-08-01 13:03:55'),  
(11, 765, '2023-08-01 13:04:25'),  
(11, 2146.2,'2023-08-01 13:04:42'),  
(11, 10077.6, '2023-08-01 13:05:05'),  
(11, 293.41,'2023-08-02 05:41:07'),  
(11, 1799.4,'2023-08-07 07:26:10'),  
(11, 3187.22, '2023-08-07 07:27:23'),  
(11, 1933.5,'2023-08-07 07:28:11'),  
(11, 3054, '2023-08-07 07:31:50'),  
(11, 1341.38, '2023-08-07 21:22:35'),  
(11, 1455.75, '2023-08-08 04:38:05'),  
(11, 1500.75, '2023-08-08 04:39:57'),  
(11, 1659.03, '2023-08-08 06:00:09'),  
(11, 1465.65, '2023-08-08 23:27:46'),  
(11, 1504.44, '2023-08-08 23:28:19'),  
(12, 399660000, '2023-07-30 09:51:20'),  
(12, 90580200, '2023-07-30 09:52:29'),  
(12, 169524000000, '2023-08-01 13:21:54'),  
(13, 198.31,'2023-07-31 04:37:21'),  
(13, 594.94, '2023-07-31 10:52:23'),  
(13, 799.32,'2023-07-31 11:05:12'),  
(14, 1175.88, '2023-08-01 17:54:27'),  
(14, 1067.4,'2023-08-01 17:55:03'),  
(14, 1176.95, '2023-08-01 17:55:52'),  
(14, 1055.97, '2023-08-01 17:56:09'),  
(14, 1049.63, '2023-08-01 21:51:50'),  
(14, 586.81, '2023-08-01 23:12:48'),  
(14, 1513.58, '2023-08-03 08:52:12'),  
(14, 1513.58, '2023-08-03 08:52:29'),  
(14, 378.68,'2023-08-03 11:28:28'),  
(14, 1277.91, '2023-08-07 23:16:01'),  
(14, 1580.25, '2023-08-08 03:53:59'),  
(14, 1508.7,'2023-08-08 03:54:25'),  
(16, 51964500000, '2023-08-02 04:27:40'),  
(16, 51002000000, '2023-08-08 04:40:57'),
```

```
(16, 1517.76, '2023-08-08 04:41:39'),
(16, 1508.7,'2023-08-08 04:42:00'),
(17, 1611.3,'2023-08-08 04:43:18'),
(17, 1703.88, '2023-08-08 04:43:32'),
(18, 211759000000, '2023-08-08 05:01:37'),
(18, 1770.72, '2023-08-08 05:02:09'),
(20, 159693000000, '2023-08-08 05:12:36'),
(20, 1646.08, '2023-08-08 05:13:02');
```

```
DROP TABLE IF EXISTS `s_portfolio`;
CREATE TABLE `s_portfolio` (
  `userId` int NOT NULL,
  `ticker` varchar(4) NOT NULL,
  `buyDate` timestamp NOT NULL ON UPDATE CURRENT_TIMESTAMP,
  `amount` int NOT NULL,
  `purchaseVal` float NOT NULL,
  PRIMARY KEY (`userId`,`buyDate`),
  KEY `ticker` (`ticker`),
  CONSTRAINT `s_portfolio_ibfk_1` FOREIGN KEY (`userId`) REFERENCES
`s_suckers` (`userId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `s_portfolio` (`userId`, `ticker`, `buyDate`, `amount`, `purchaseVal`)
VALUES
(11, 'F', '2023-08-08 04:38:05', 75, 19.41),
(11, 'VZ', '2023-08-08 06:00:09', 34, 48.8),
(11, 'VZ', '2023-08-08 23:27:46', 30, 48.86),
(11, 'CLF', '2023-08-08 23:28:19', 63, 23.88),
(14, 'GME', '2023-08-08 03:53:59', 50, 31.61),
(14, 'TSLA', '2023-08-08 03:54:25', 4, 377.18),
(16, 'MSFT', '2023-08-08 04:40:57', 103000000, 495.17),
(16, 'TQQQ', '2023-08-08 04:41:39', 24, 63.24),
(16, 'TSLA', '2023-08-08 04:42:00', 4, 377.18),
(17, 'RMBS', '2023-08-08 04:43:18', 20, 80.56),
(17, 'SPOT', '2023-08-08 04:43:32', 8, 212.99),
(18, 'AMZN', '2023-08-08 05:01:37', 992633688, 213.33),
(18, 'TQQQ', '2023-08-08 05:02:09', 28, 63.24),
(20, 'META', '2023-08-08 05:12:36', 336310000, 474.84),
(20, 'PLTR', '2023-08-08 05:13:02', 61, 26.98);
```

```
DROP TABLE IF EXISTS `s_suckers`;
```

```

CREATE TABLE `s_suckers` (
  `userId` int NOT NULL AUTO_INCREMENT,
  `fName` varchar(50) NOT NULL,
  `lName` varchar(50) NOT NULL,
  `uName` varchar(50) NOT NULL,
  `bday` datetime DEFAULT NULL,
  `password` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci
  NOT NULL,
  `sex` varchar(1) CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci NOT NULL,
  `cCard` bigint NOT NULL,
  PRIMARY KEY (`userId`),
  UNIQUE KEY `uName` (`uName`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

```

INSERT INTO `s_suckers` (`userId`, `fName`, `lName`, `uName`, `bday`,
`password`, `sex`, `cCard`) VALUES
(3, 'Juan','Gutierrez', 'trasgu', NULL, 'password', '', 2147483647),
(6, 'John Jacob','Jingleheimer-Schmidt', 'jingleschmells', NULL,
'ihatemyname', '', 1234567898765432),
(8, 'Kyle','Absten', 'kabsten', NULL, 'password', '', 1234567812345678),
(10, 'Adam', 'Smith', 'MaoWasWrong', NULL, 'butnotaboutlandlords',
'', 7894561234567894),
(11, 'Zaria', 'Kokiri', 'kakarikokid', '1998-11-21 00:00:00', 'imisslink',
'F', 1231231234512121),
(12, 'Elmo','Tusk', 'emeraldboy',NULL, 'buyingtwitterwasagoodidea', '',
1111111111111111),
(13, 'Brian','Johns', 'bjones', '0000-00-00 00:00:00', '123brian', 'M',
1234567891234567),
(14, 'Barbie', 'Heimer', 'barbenheimer', '2023-04-20 00:00:00',
'bombassbabe', 'F', 8888888888888888),
(15, 'Kevin', 'Absten', 'kabsten2', NULL, 'password', '',
1234567812345678),
(16, 'William', 'Gates', 'logicgates', NULL, 'fosssucks', '',
8888888888888888),
(17, 'Frank', 'Zappa', 'fzaps', '1940-12-21 00:00:00',
'moustacheman', 'M', 3333333333333333),
(18, 'Jeffrey', 'Bezos', 'amazingjezos', '1964-01-12 00:00:00',
'pee bottle', '', 9999999999999999),
(20, 'Mark','Zuckerberg', 'metanym', '1984-05-14 00:00:00', 'definitelyhuman',
'', 6666666666666666);

```


Conclusions

This project was very fun to work on, and gave us all experience with full stack development of a web application. We practiced interacting with an SQL database, coding with Node.js and Express, advanced our familiarity with HTML, JavaScript and CSS, and presented a holistic opportunity to put into practice everything we learned in this course. 10/10 would code again.

To date, we have looted
\$644,432,695,119.74 from 9
absolute rubes. To the
Moon! 🚀 🌙



© CST336-40 Team 6
Kyle Absten, Juan Gutierrez, Harut Lementsyan
California State University Monterey Bay 2023