livecrylove: Yinwei Zhang, Sascha Gordon-Zolov, Jackie Zeng, Princeden Hom
Mr. Mykolyk
Software Development
P01: ArRESTed Development
Project: Stock Market Manager
11/25/2024
TARGET SHIP DATE: 2024/12/17
Time Spent: 2 hours

P01 Design Doc

Overview:

Project Objective: Develop Flask-based web application that provides users with tools for managing stock portfolios, viewing market data, and currency conversions through various APIs.

Primary Features:

1. View stock portfolio and account balances
2. Dynamic marke updates using APIs
3. Currency exchange rate conversion
4. Persistent user data storage in db
5. User auth (login, registration, logout)

Components:

Frontend:

Templates Folder

- home.html
  - If logged in should display current stock portfolio
  - Otherwise should display a login button, as well as buttons to still see currency exchange and market data
- login.html
  - Login user given valid username & password
- register.html
  - Allows the user to register an account given a unique username
- logout.html
  - Allows the user to logout
- currencyExchange.html
  - Allows user to select a currency and amount and display a conversion to any currency of their choice
- marketData.html
  - Displays market data retrieved from apis as well as the ability to convert that data to their chosen currency
- bank.html
  - Allows user to see their current balance as well as the value of their stocks

Tailwind CSS to ensure consistent and responsive UI/UX

Middleware:

app.py

- Serves the website, includes routing and rendering templates

currency.py

- Python file that contains helper functions that utilize our currency conversion APIs market.py

- Python file that contains helper functions that utilize our market data APIs

keys_<api_name>.txt

- Text file that contains the keys for APIs for secure access to external data sources. Each key is in a separate file under the folder keys

Database:

app.db

- SQLite db storing persistent user information and stock data and connects to backend for data retrieval

Database Organization:

- Users Table
    - Contains username (text), email (text), password (text), current money in account (float), and stocks owned (text)

Column Name	Data Type	Constraints	EX
username	TEXT	PRIMARY KEY, UNIQUE	“livecrylove123”
email	TEXT	UNIQUE, NOT NULL	“livecrylove@gmail.com”
password	TEXT	NOT NULL	“safepassword”
current_balance	FLOAT	DEFAULT 0.0	314159.26
stocks_owned	TEXT (JSON)	NOT NULL	REFERENCE

- User Stocks
    - Contains ticker symbol (text) and quantity of stock (int)

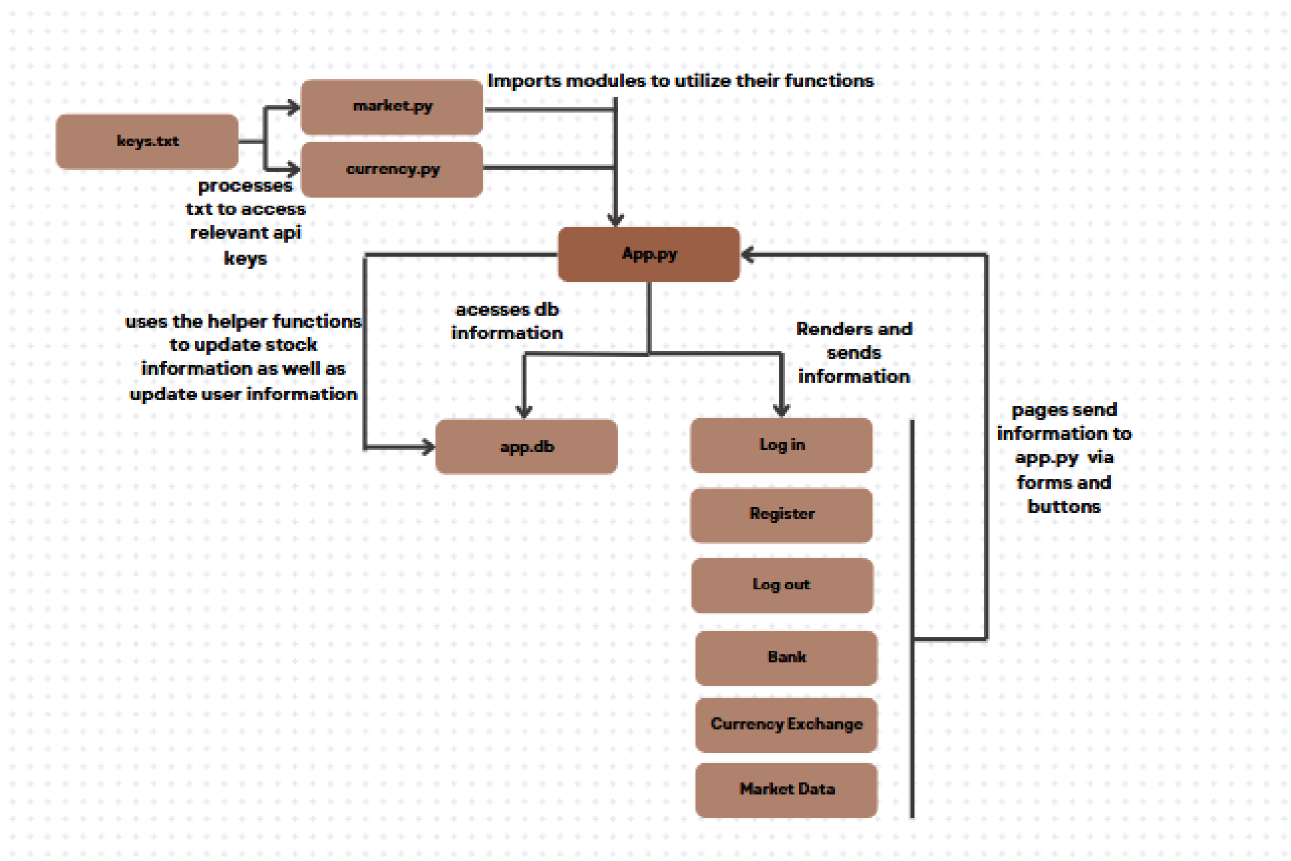
Column Name	Data Type	Constraints	EX
field	TEXT	PRIMARY KEY, UNIQUE, NOT NULL	“TSLA”
quantity	INT	NOT NULL	2

- Stocks Table
    - Contains a stock name (text), ticker symbol (text), price (float), sector (text) and time last updated (datetime)

Column Name	Data Type	Constraints	EX
stock_name	TEXT	NOT NULL	“Apple Inc.”

ticker_symbol	TEXT	PRIMARY KEY, UNIQUE	“APPL”
price	FLOAT	NOT NULL	239.4
stock_sector	TEXT		“Tech”
last_updated	DATETIME	DEFAULT CURRENT_TIMESTAMP	"2024-12-03 12:00:00"

Component Map


Imports modules to utilize their functions
market.py
keys.txt
currency.py
processes txt to access relevant api keys
App.py
uses the helper functions to update stock information as well as update user information
acesses db information
Renders and sends information
pages send information to app.py via forms and buttons
app.db
Log in
Register
Log out
Bank
Currency Exchange
Market Data

Directory:

```
p01/
├── app/
│   ├── templates/
│   │   ├── home.html
│   │   ├── login.html
│   │   ├── register.html
│   │   ├── logout.html
│   │   ├── currency_exchange.html
│   │   ├── market_data.html
│   │   └── bank.html
│   ├── __init__.py
│   ├── app.py
│   ├── currency_utils.py
│   ├── market_utils.py
├── keys/
│   ├── 1forge.txt
│   ├── currencylayer.txt
│   ├── fixer.txt
│   ├── financialmodelingprep.txt
│   ├── marketstack.txt
│   └── iexexchange.txt
├── database.db
├── requirements.txt
└── README.md
```

Chosen APIs:

- currencylayer.com/documentation: Exchange rates and currency conversion
- fixer.io/: Exchange rates and currency conversion
- site.financialmodelingprep.com/: Stock, Futures and Forex Market Data
- marketstack.com/: Market data provider

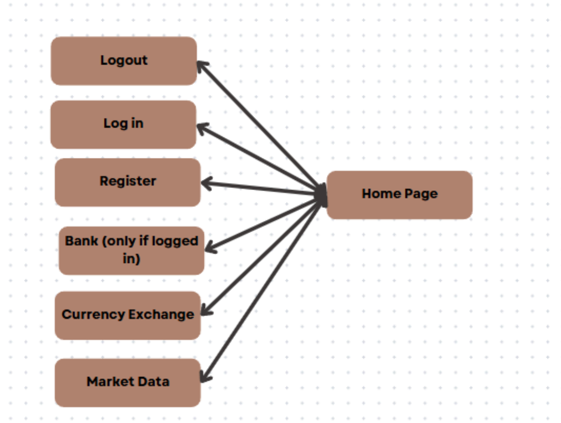
Functionalities:

- Users can log in and log out
- Users will be able to input and save their current bank data
- Users can place money in the stock market, using this website as a proxy for interacting with the stock market
- Users do not have to log in, but certain features will be inaccessible. For instance, users will not be able to observe how differing amounts change over time. However, guests will be able to check currency conversions and view graph data on the current stock market.
- A database will store a user's data, password, and banking data
- Stock market data will be stored in a database, with each stock given a different column
- Users will be able to edit and watch their banking data
- We note that money in the stock market is dynamic, or that the database with market data will be updated as the API data changes

Front End Framework:

- We are using **Tailwind** as our frontend framework because we like it's ease of use combined with the amount of easy customization available for each class and the amount of possible styling options

Sitemap



Pages:

Home (home.html)

- Purpose: Acts as the entry point to the site.
- Features:
  - Displays options for guest users:
    - Login
    - Register
    - Links to Market Data and Currency Exchange pages.
  - For logged-in users: Displays portfolio summary.

Login (login.html)

- **Purpose**: Allows users to log in to their accounts.
- **Features**:
  - Redirect to **Home** after successful login.
  - Includes a link to **Register** for new users.

Register (register.html)

- **Purpose**: Enables new users to create an account.
- **Features**:
  - Takes user details (username, email, password).
  - Redirects to **Login** upon successful registration.

Market Data (marketData.html)

- **Purpose**: Provides real-time and historical stock market information.
- **Features**:
  - Guest and logged-in users can view stock prices and trends.
  - Button to go to **Currency Exchange, Home,** and **Portfolio** (Only if logged in)

Currency Exchange (currencyExchange.html)

- **Purpose**: Enables real-time currency conversions.
- **Features**:
  - Select base and target currencies.
  - View historical exchange rates.
  - Button to go to **Market Data, Home,** and **Portfolio** (Only if logged in)

Portfolio (bank.html)

- **Purpose**: Displays the logged-in user's financial summary.
- **Features**:
  - Shows account balance.
  - Displays stocks owned and their current values.
  - Button to go to **Market Data, Home,** and **Currency Exchange**

Logout (logout.html)

- **Purpose**: Ends the user session.
- **Features**:
  - Redirect to **Home** after logging out.

Visual Layout Notes::

- Every page will contain the same heading to provide users with a sense of pleasant consistency. This heading will consist of our team name and a set of links to different pages

- General visual design (CSS) will be standardized across every page to allow for comfort and recognition in within the website
- Buttons from Tailwind will be used to move from page to page to allow for navigation
- There will be instructions for how to use the heading on every page, and the built-in back-button can be used in place of an additional coded back-button

Roles:

	Jackie	Sascha	Yinwei	Princeden
Flask setup and routing	✔	✔		
API integration			✔	✔
HTML template creation and CSS styling			✔	✔
Database setup and integration	✔	✔		
Testing and debugging	✔	✔	✔	✔