FORTNITE: Ryan Lau, Craig Chen, Elizabeth Paperno, Hui Wang
softdev
p1: ohayo
2022-12-06
time spent: 50 hrs
target ship date: 2022-12-19

the idea

A personal dashboard to keep your life in order!

On the dashboard, we will display the weather, sunrise/sunset time, stock data, an inspirational quote for the day, a list of news articles, and an area to write to-do items.

APIs

- Weather API: https://openweathermap.org/api
    - Pull forecast information for the day
- Stock Data: https://alpaca.markets/docs/api-references/market-data-api/stock-pricing-data/historical/
    - Pull stock index information of the day
- Random Quote of the day: https://favqs.com/api
    - Pull random quote to show on homepage
- News API: https://developer.nytimes.com/apis
    - Pull top stories to show on homepage

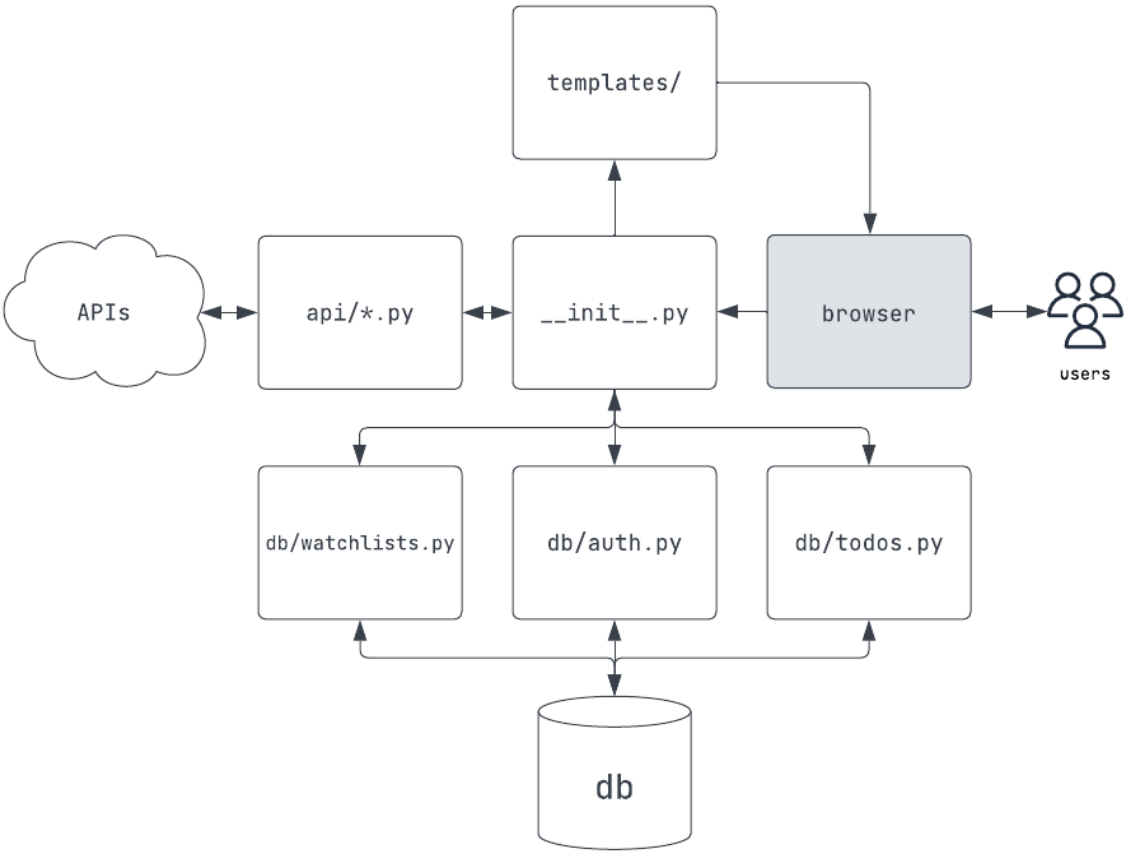
program components

- __init__.py: entry point, flask server, define routes, query **db** with functions imported from **auth.py**, **todos.py**, and **watchlists.py**
- templates/
    - layout.html
        - helper html for every html page
    - login.html
        - shown if not logged in, displays a login form and a link to /register
        - creates user entry in users table if username does not exist, displays error if username is taken
        - Prompts user to enter zipcode as well
        - redirects back to / when user is successfully registered
    - macros.html
        - helper html for dashboard

    - dashboard.html
        - shown if logged in, see mockup in section **FEF**
        - widgets displaying weather, news, stocks, todos
    - settings.html
        - shown if user clicks on settings from the dashboard
        - prompts user to change zip code or password or delete account entirely

- db: sqlite3 database; see section **database structure**
- auth.py: functions that perform SQL queries on **users** table in **db**
    - validate credentials
    - check if username is available
    - create user
    - Get lat, lon, and city from inputted zipcode
    - create table
    - delete table
- todos.py: functions that perform SQL queries on **todos** table in **db**
    - create todo
    - mark todo as done
    - delete todo
    - create table
    - delete table
- watchlists.py: functions that perform SQL queries on **watchlists** table in **db**
    - check if ticker exists
    - add ticker
    - remove ticker
    - create table
    - delete table
- api.py: functions that return parsed data from rest APIs
    - get weather and sunrise/sunset time
    - get stock data
    - get quote of the day
    - get news

component map



database structure

user_info

username	password	lat	lon	city	zip
rhinoceros	0i@8D7Uh3P18	38.8951	-77.0364	New York	11227
tofr	f78Q7&W*71fA	24.2351	-50.4592	Los Angeles	12352

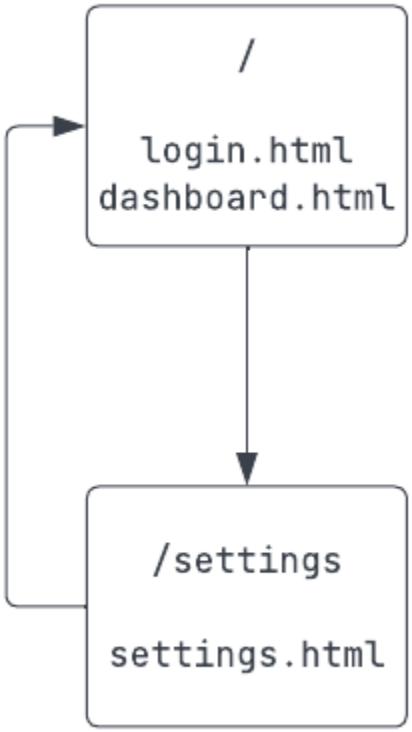
todos

todo_id	username	item	completed
0	tofr	walk the dog	FALSE
1	tofr	do softdev homework	FALSE
2	tofr	read	TRUE

watchlists

username	ticker	company_name
tofr	GOOG	Alphabet Inc Class C
tofr	AAPL	Apple Inc
bob	AAPL	Apple Inc

frontend flow



Our **login.html** also handles user registration and because our app is a personal dashboard, everything is shown on **dashboard.html**!

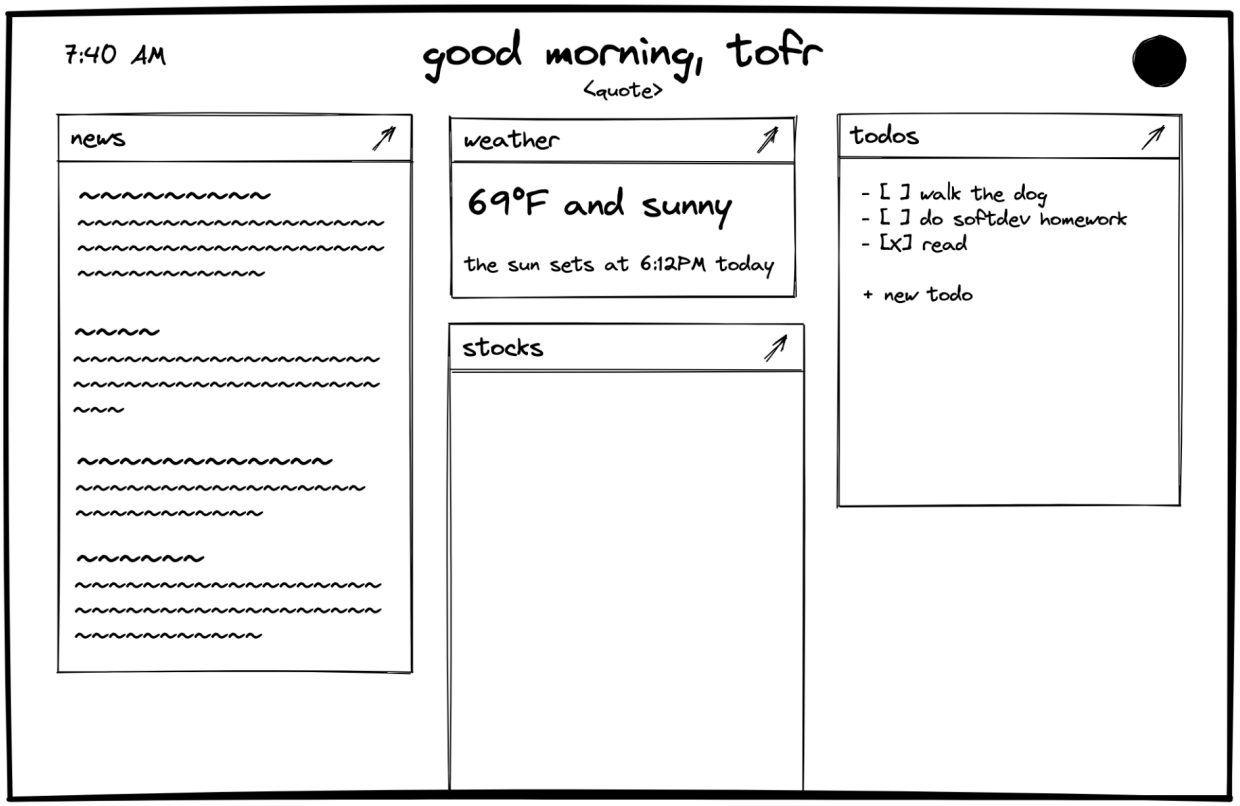
FEF

We plan on using Bootstrap because it offers better documentation and more pre-made components compared to Foundation.

We plan on using these following features:

- Grid system
- Buttons
- Card
- Dropdown
- Spacing utility classes
- Nav and tabs

This is what our front end will look like:



post MVP features

- Animate to do list to allow users to mark their tasks as completed.
- ~~Add a settings page to change password and zip code~~

Tasks

- db work
    - assigned to **Elizabeth**
- frontend
    - assigned to **Hui** and **Ryan**
- Flask server
    - assigned to **Craig**
- api.py + api card (if necessary)
    - weather: **Craig**
    - stocks: **Ryan**
    - quote: **Elizabeth**
    - news: **Hui**