RESTful Rhinos: Brian Liu, Victor Casado, Danny Mok, Marco Quintero

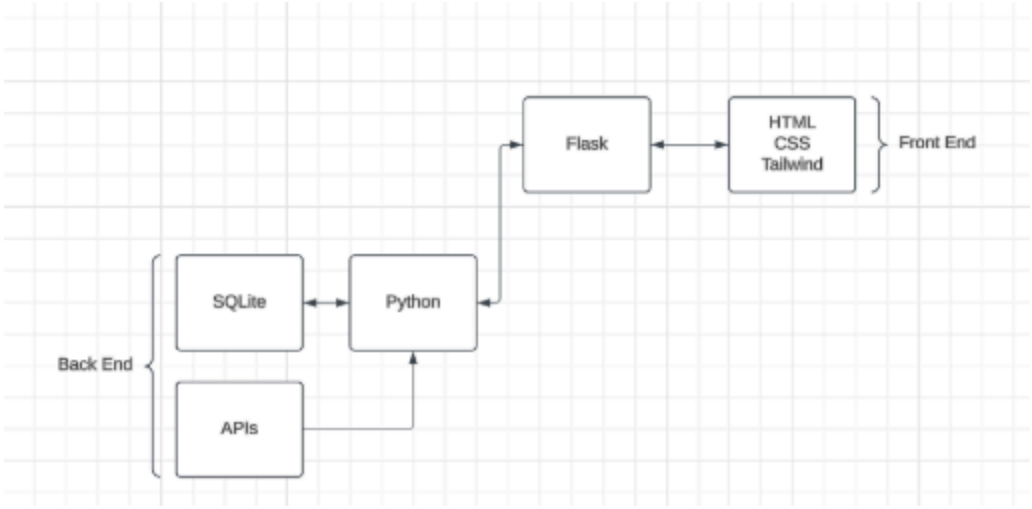
Softdev
P01: ArRESTed Development
2024-11-25
Target Ship Date: 2025-01-24

Design Document

Project Description

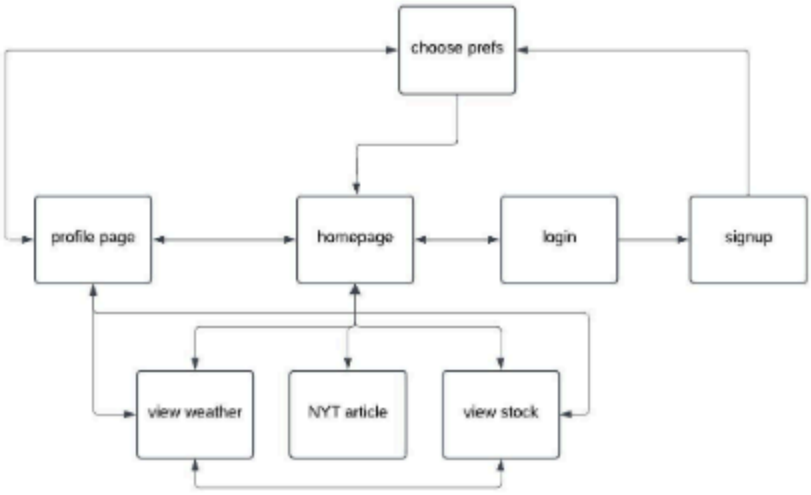
This website aims to provide personalized and real time information to the user—like a daily feed or landing page. Registered users can input their preferences for location and interests, which is then used to showcase personalized stock information, weather, and news to the user. From the homepage, the user can then go to separate pages that contain more detailed information about the articles, stocks, or weather. The user will have access to their profile, where they can change their preferences.

Program Components + Component Map



1. Flask/Python: connects the routes to different pages and connects the backend databases to the frontend.
2. SQLite3: stores information about user info (login information) and their preferences (location, stock preferences, news preferences).
3. HTML + Jinja + CSS + Tailwind: allows for dynamic web pages that are personalized based on the user.
4. API: retrieves real-time and customized information which is then displayed on the website using Flask and Jinja templating

Site Map



*Note NYT article does not go back to the homepage, as it is a link to an actual article

- View Stock will display historical and current stock information
- View Weather will display weather data, including any current weather alerts
- The profile page will display preferences and username
- Homepage, view weather, and view stock will display navbars based on user prefs
- Choose preferences will allow the user to select things they want to follow (will show up in navbars)
- Login/Signup Self-Explanatory

Databases

- Table containing {id, username, password}
- Table containing {id, stock1, stock2, stock3, etc...}
    - Each stock is a boolean indicating whether a user cares about said stock
    - Similar tables for weather regions, news preferences, and holiday regions
- Databases to cache API data as an optional further requirement

Frontend Frameworks

FeF of choice: TailwindCSS

Tailwind provides user customization for accessibility, a clean user interface, and easy in-line styling that allows for different styling amongst the same HTML tags.
Integral features:

- Animations - Pings that notify users new updates on the pages they follow
- Sidebar - Easy access to pages users follow, home, and their profile page
- Overflow and overscroll behavior - Adding scrollbars to each page's info so it is a separate scrollbar from the main page.

Possible features:

- Dark mode - For user accessibility
- Image upload - For user profile pictures
- Resizing based on window size

APIs

Calendarific

- 500 calls/month
- Displays list of holidays based on country

NYT API

- 500 calls/day
- Displays news articles from the NYT based on National/International and topic parameters

Open Weather Map

- 1 million API calls per minute
- Displays weather information based on location

Financial Modeling Prep API

- 250 calls / day
- Displays current and historical stock prices and information

Task Breakdown

1. Brian Liu
    a. API: Calendarific
        i. Implement functionality for the Calendarific API and make sure the correct info is displayed.
    b. Linking pages and other backend stuff through Flask
2. Victor Casado
    a. API: Financial Modeling Prep
    b. Implement functionality for the Financial Modeling Prep API and make sure the correct info is displayed according to the user's preferences
        1. Display prices for stocks / other things user cares about
    c. Connect database to front end through Flask
3. Danny Mok
    a. API: NYT API
        i. Implement functionality for the NYT API and make sure the correct news articles and info about them is displayed according to the user's preferences.
    b. Database implementation
        i. Implement a system for login and user preferences.
        ii. Allow users to change their preferences.
        iii. Possibly store api data
4. Marco Quintero
    a. API: Open Weather Map
        i. Implement functionality for the Weather API and make sure the correct info/images/links are displayed according to the user's preferences.
        ii. Implement location-based data based on the address provided by the user
        iii. Default the homepage widget to the current weather at the default location but clicking on the widget to the full page will allow the user to see forecast in different areas
    b. Front-end framework (Tailwind CSS)

Further Optional Functionality:

- Function to change password
- Profile photo
- Tags system to content so a user can "follow" content without going to update preferences
- Recently viewed section
- Alert / Notification system
- Date/time