Assignment1

Group: loop

Team members: Qiming Ye, Zixiao Ren

Project choices

Stock Analysis Tool

Description

A website that analyzes the stock from different perspectives. For example, finding what stock has the most growth potential, finding the most growth potential categories(energy, healthcare, industrial...), etc.

Group Decision

At first, we chose to analyze stock trends, but then we thought there were a lot of financial APPs, and analyzing these stocks didn't have much value because these APPs already had quite a lot of data analysis. So we did not choose to make a stock analysis tool.

Reason for the Decision

- Cost
  - Too many stocks in the current stock market. The database would be too large to be managed as a course project if we add all the stocks in the world into the database. If we don't include all the stocks, then the data analysis result might not be accurate.
  - None of the team members is familiar with stocks, it might take us lots of time to learn how to analyze stock data.
  - There are only two people in the group. The workload would be too heavy if we have a high expectation of the accuracy of the stock analysis
- Benefits
  - It has a good market prospect if this tool can be developed to provide very accurate stock data analysis

Youtube Analysis Tool:

Description:

A website that could help people, especially YouTubers, to analyze how to create a popular youtube channel. The user would need to provide a channelId to the website and the website would all the channel information including channel title, channel description, playlists, videos, etc. The most important part is that it would help the user to find what are the keywords they can put into the video title and description to catch people's eyes. The keywords are generated based on the previously published popular videos on this channel.

Group Decision:

We decide to work on this youtube analysis tool after having a meeting to prove that this is a feasible idea for the project. We found that all kinds of data that are needed for the analysis could be get from the youtube APIs. (More detail about youtube APIs in week1 meeting notes).

Reason for the Decision:

● Cost
  ○ We found that the youtube APIs are not very stable when running them. Sometimes it responds with a 403 error for no reason. This might cause issues when the user runs our tool. (With further discussion, we decided to use a database for backup. If the API responds with a 403, then we can find if our database has previously stored data to use.)
● Benefits
  ○ All kinds of data we need for analysis could be easily obtained from Youtube APIs. This could increase the accuracy of the analysis since all the data are real-time data.
  ○ All of our teammates have experience using APIs in javascript, which would save us time in figuring out how to get our data.

Project Detail (Youtube Analyzing Tool)

Feature:

  a. The user may get channel, playlists, and videos info by searching ChannelID
    i. Channel: channel title, description, published at, custom url, followers count
    ii. Playlist: playlist title, description, published at, video number, total # of likes and view count
    iii. Video: video title, description, view count, like count, actual start time, scheduled start time(if available), duration

      b. Find a recommended list of keywords for video title (based on the keywords with high frequency in the title of previous popular videos)
- i.   The recommended keywords based on all the videos
- ii.   The recommended keywords based on videos in a single playlist

      c. Find a recommended list of keywords in the description of popular videos (based on the keywords with high frequency in the description of previous popular videos)
- i.   The recommended keywords based on all the videos
- ii.   The recommended keywords based on videos in a single playlist

Add-ons(features good to add if we have extra time):

- Sort playlists by publishing time/video number/total # of likes of videos
  - (Note that not all channels have playlists. This feature is only for channels that have playlists created)
- Sort video by view count/like count/duration/published time
- Find a recommended duration time for videos (based on the duration times of previous popular videos)
- Find a recommended publish time for videos

Tech Stack and Toolchain

Container: Docker

Frontend: React

Backend: NodeJs

Source of data: Youtube API

Database: Mongodb (This is a database for backup. If there's any issue with the Youtube API, for example, the API responds with a 403 error, then searches the data in the database. The data in the database are updated whenever API responds with real-time data)

CI/CD: Github Actions+Mocha/Heroku

How we arrived at this design

Summary: Based on the educational and work background listed below, we decided to use React for frontend, NodeJs for backend, Mongodb for the database, and Github actions, Mocha, and Heroku for CI/CD since we are both very familiar with these languages/ tools. The reason why we choose Docker as our container is that Zixiao had experience using Docker in CSCC01 and CSCD27. We have 7 features listed initially. However, since we only have 2 people in the

team and we both have very heavy workloads this semester, we do not think we could finish implementing all the features, so we picked 3 of them to the feature list and the rest are the add-ons if later we have time to implement.

- Members' educational and work background
  - Both finished CSC309 and are familiar with Express(React and NodeJs), Mongodb, and Heroku
  - Qiming(Austin) Ye:
    - 4-month internship in a startup last summer (familiar with full stack development)
  - Zixiao(Shirley) Ren,
    - 4-month coop at Tencent as an automated test developer (familiar with test developing)
    - 8-month coop at the National Bank of Canada. Developed an internal website with React.
    - A research assistant(React developer) of Interactive Media Lab. Currently developing automated test using Mocha framework.
- Members' workloads, schedules, and educational goals
  - Workloads: Extremely heavy workload
    - Zixiao: Course work(CSC302, CSCD03, CSCC46, CSCC63, CSCD27)+research
    - Qiming(Ausin):
      Fall term course: CSC302; CSC369; CSC454; CSC491; ENT200; AST101
  - Schedules:
    - Monday 10am~12am weekly progress meeting
    - At least 5 hours for development every week
    - Use Zoom share screen for code review
  - Goals
    - Both:
      - Learn how to analyze data from existed API
      - Learn how to design the workflow and techflow of a project to minimize the cost
      - Learn how to communicate with the team to maximize efficiency
- Understanding the potential risks of this approach
  - Might not have enough time since we both have very heavy workloads. This is the reason why we put some of the features to 'pending add-ons rather than in the milestones.
  - We only have 2 team members. The workload for this project might still be heavy even though we put some of the features into add-ons.
  - There might be a case that API responds 403 error and the provided channelId is not yet in our database. Need to handle edge cases carefully.

Preliminary Project Development Plan

Milestone1

Implement feature a:

        The user may get channel, playlists, and videos info by searching ChannelID

                i. Channel: channel title, description, published at, custom url, followers count

                ii. Playlist: playlist title, description, published at, video number, total # of likes and view count

                iii. Video: video title, description, view count, like count, actual start time, scheduled start time(if available), duration

Team members' responsibilities:
- Both: Analyze youtube API, MongoDB database design, UI/UX design
- Zixiao: playlist info
  - Obtain playlist data from youtube API and store/update it in the database
  - Database structure design for playlist
  - Backend internal API design for playlist
  - Frontend for playlist
- Qiming(Austin): show channel and video info
  - Obtain playlist data from youtube API and store/update it in the database
  - Database structure design for channel and videos
  - Backend internal API design for channel and videos
  - Frontend for video

Milestone2

Implement feature b:

        Find a recommended list of keywords for video title (based on the keywords with high frequency in the title of previous popular videos)

Team members' responsibilities:
- Both: Design algorithm for finding recommended keyword
- Zixiao: The recommended keywords based on videos in a single playlist
  - Frontend and backend implementation
- Qiming(Austin): The recommended keywords based on all the videos
  - Frontend and backend implementation

Milestone3

Implement feature c:

        Find a recommended list of keywords in the description of popular videos (based on the keywords with high frequency in the description of previous popular videos)

Team members' responsibilities:
- Zixiao: The recommended keywords based on videos in a single playlist
  - Frontend and backend implementation
- Qiming(Austin): The recommended keywords based on all the videos
  - Frontend and backend implementation

Milestone4

Testing (and add-ons if have time)
- Zixiao: Write test script for playlist-related backend APIs
- Qiming(Austin): Write test script for video-related backend APIs

Implementation of The Technology Stack

Heroku link: https://csc302-assignments.herokuapp.com/

GitHub: https://github.com/Austin618/CSC302-Assignments/tree/main

Run dev environment: docker-compose up (then open localhost:5000 to view the website)