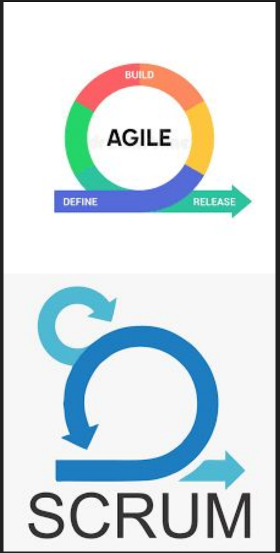
YouTube Wrapped

Erik Haller, Kyle Johnson, Ethan Kissell,
Araceli Luna-Cabral, Renata Zurita

Description

- Basically a Spotify Wrapped for YouTube
- Uses YouTube’s Data API to show information about a user such as:
  - Most watched channel
  - Most watched video
  - Their first video watched
  - Last video watched
  - Most liked video watched
- These cards also have links to the video themselves
- Users are anyone interested in learning more about their YouTube viewing habits

Methodology



Description

- We had weekly group agile meetings where we would optimize production and collaborate.
- Weekly simulated client meetings with TA.

Rating: 4

- Convenient to have a set meeting time weekly.
- Weekly TA feedback was necessary to maximize progress on our features.

Project Tracker: GitHub Projects

Description:

- Keep track of work currently being done on project
- Show completed work

Rating: 2

- Useful to organize and plan at beginning of project
- Did not utilize as project progressed

VCS Repository: GitHub

Description:

- Allow users to work on their own branch and push to a main to merge code
- Resolve merge conflicts when working on the same code
- Keeping track on what changes within the codebase

Rating: 4

- Very reliable, allows users to see and access previous commits
- Resolving merge conflicts can be kind of annoying

Database: PostgreSQL

Description:

- Created a database to store registered usernames (Channel IDs) and passwords.
- Checked login inputs with registered usernames and passwords to determine if account exists.
    - If username or password does not exist in the database, the login would fail.

Rating: 3

- This was simple enough to use and create and our register and login functions depend on the database.
- However, didn’t have any use for making databases anywhere else.

IDE: VsCode

Description:

- Environment for coding the full stack of the project

Rating: 5

- Useful for working between files such as creating a feature on the front-end and adding functionality on the back-end
- Keeps files organized while coding
- Helps catch syntax errors

UI Tools: HTML, Bootstrap

Description:

- Tools used to build the UI for each page
- HTML for page layout, and CSS/Bootstrap for stylizing (background color, text size, etc.)

Rating: 5

- These tools were used for every single page.
- Very simple tools to use with a lot of resources online telling you how to use them

Application Server: NodeJS

Description:

- Used to code API calls and routes

Rating: 3.5

- Project couldn’t work without it
- But, it does require coding in JavaScript
- And Axios has inconvenient quirks

Deployment Environment: Azure

Description:

- Hosting the site on the cloud so it can be accessed by anyone

Rating: 4

- Easy to set up, and rehost site when changes are made
- Limits users to 750 hours before charging $100 if not taken down

External API: YouTube Data API

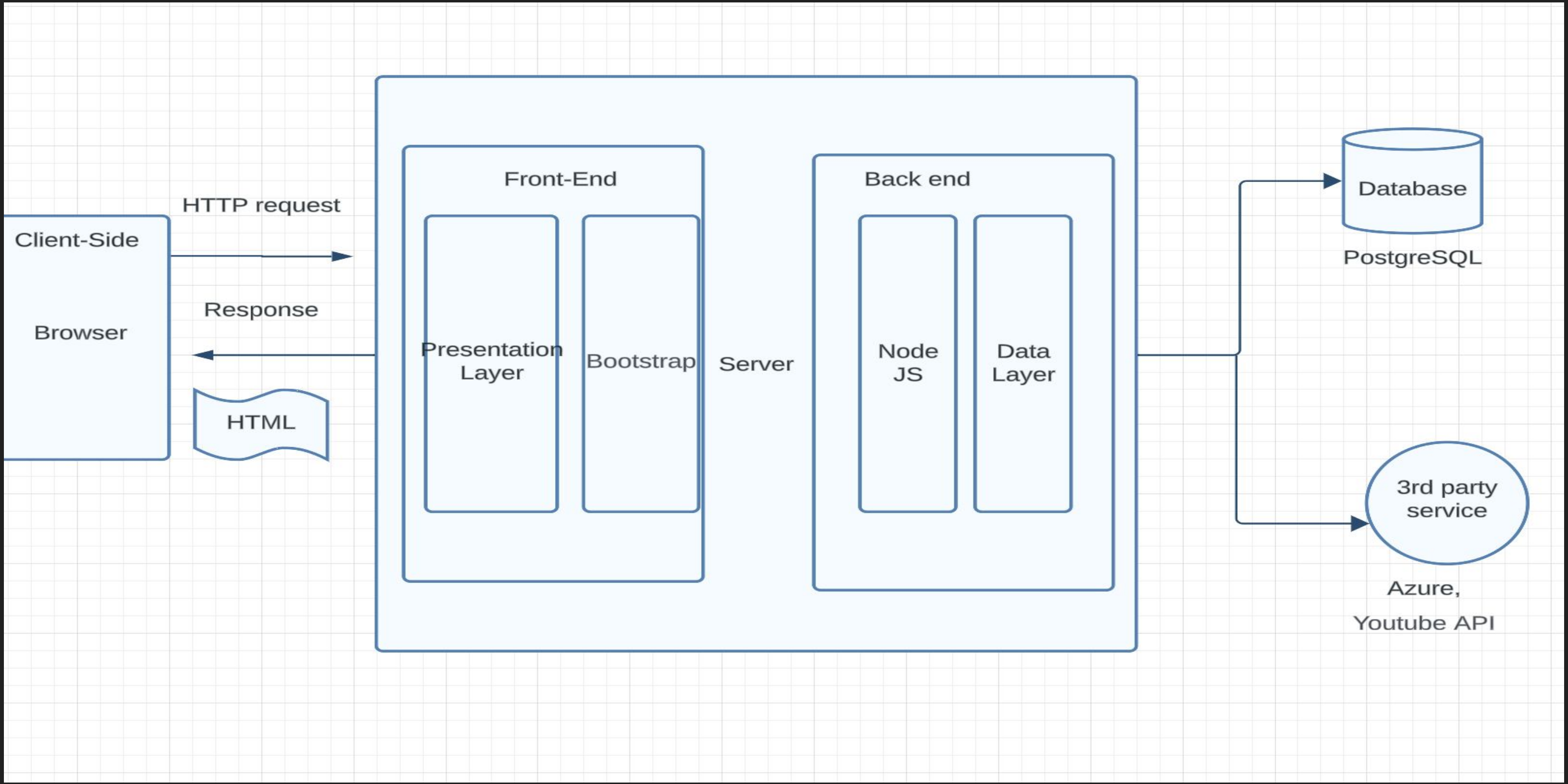
Description:

- Used to get data from YouTube
  - activities of an account, information about videos, etc

Rating: 4

- Needed for our project
- But some issues lower the rating
  - mainly, some parameters were inconsistent

Architecture Diagram



Challenges

1. Google Authentication
   a. Make it work with handles
2. YouTube Data API Limitations
   a. Use of Object arrays, functions, multiple calls
3. Test Cases Error Codes
   a. Return status 400 when test cases fail in the POST APIs for relevant pages.

Future Scope

1. Implementing Google Authentication for login
   a. Instead of asking for channel ids
2. QoL Improvements
   a. Way for users to change their password
3. Adding more statistics
   a. Most liked comment
   b. Fun ones like most disliked video or comment

Demo

http://recitation-13-team-07.eastus.cloudapp.azure.com:3000/login