YouTube Wrapped

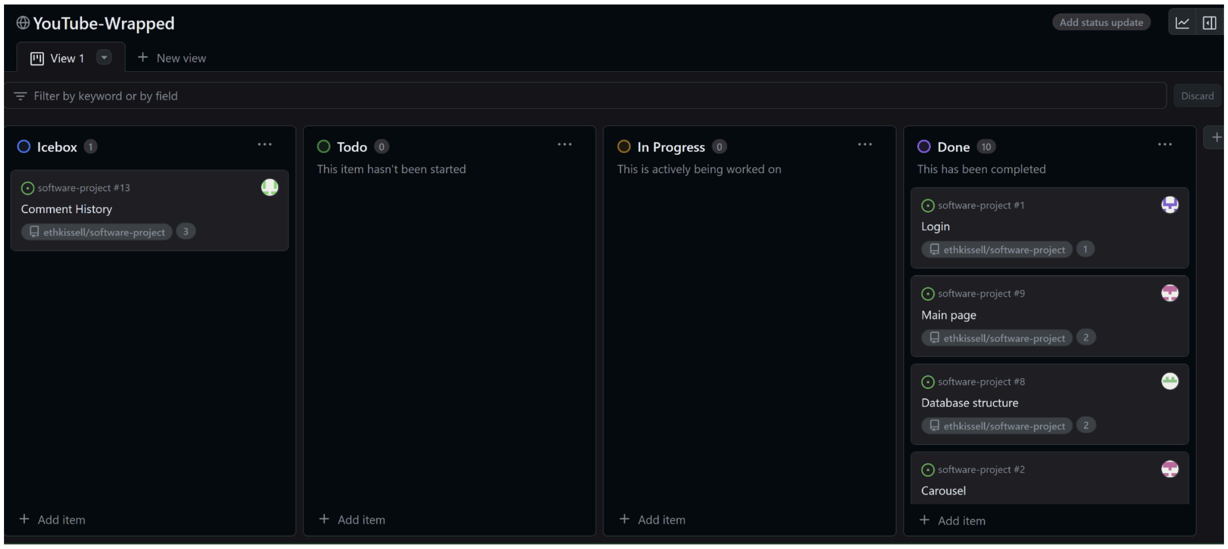
Ethan Kissell, Erik Haller, Araceli Luna, Kyle Johnson, and Renata Zurita
GitHub Usernames: ethkissell, erha2561, Araluna23, kyjo1355, renatazurita
https://github.com/ethkissell/software-project

YouTube Wrapped is an application designed to provide users with insights into their YouTube interaction patterns and preferences. Upon opening YouTube Wrapped the user is prompted to register or login with their YouTube handle. The handle allows us to fetch their channel ID, which in turn allows data to be accessed about the user, such as the video and channel the user has interacted with the most, the user's first and last watched video, and most liked video the user has interacted with, as well as information about those videos and channels. This data is then presented to the user via a carousel. The presentation of data includes interactive links to the videos as well as images of profiles. YouTube Wrapped was made with VS Code as the developers' IDE, SQL for the database structure, Node JS for integrating API calls and routes, Bootstrap/CSS for adding style to the pages and HTML for general UI structure of pages and the YouTube API to access data about the user. YouTube Wrapped provides personalized insights, analytics, and an easy-to-use user interface. Overall, YouTube Wrapped allows users to gain a deeper understanding of their YouTube consumption habits, as well as their personal tastes.

Video Demo

here

Project Tracker
https://github.com/users/ethkissell/projects/2



YouTube-Wrapped

Icebox 1

software-project #13
Comment History
ethkissell/software-project 3

Todo 0
This item hasn't been started

In Progress 0
This is actively being worked on

Done 10
This has been completed

software-project #1
Login
ethkissell/software-project 1

software-project #9
Main page
ethkissell/software-project 2

software-project #8
Database structure
ethkissell/software-project 2

software-project #2
Carousel

Contributions

Ethan Kissell: The things I worked on was creating the repository and the readme for the project, the functionality for hosting the site by filling in the env, index.js, and yaml files. I created the partials for the head, footer, nav, and main. Created the structure and functionality of the login page; and added the final styling of the login page and register pages. Added the functionality for the login and register pages to return the proper error instead of always returning a 200 or 300 code. Created the full functionality of the drawer navbar. Lastly, created a logout route that redirects to the login page. I also hosted the project on Azure.

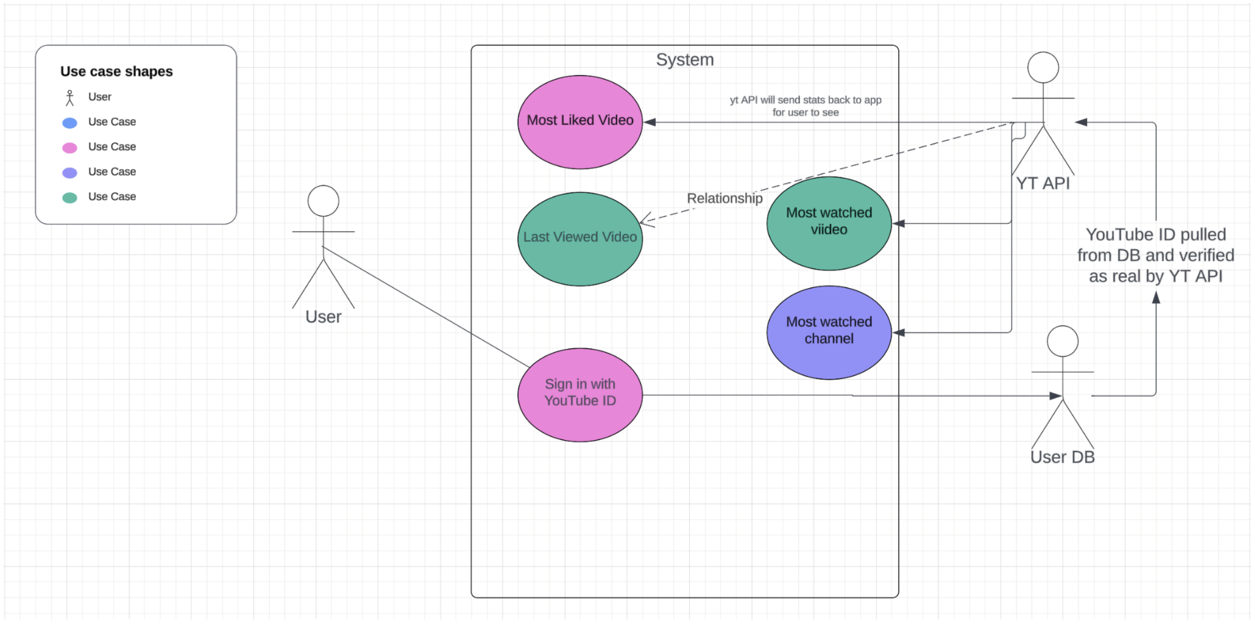
Erik Haller: I mainly worked on the back-end stuff, and I implemented the back-end functionality for the stats page and the home page, including all the calls to the YouTube Data API. I also added functionality for the YouTube handles, making it so that they would fetch the appropriate channel ID so we could fetch data for that page. So, I interacted with the YouTube Data API, as well as NodeJS, and Handlebars (to make the appropriate data display on the HTML pages).

Kyle Johnson: Things I added to the project included most of the necessary files and folders to the repository so the directory structure matched the one outlined in the project description. I also did a lot of front-end work for the stats and profile pages to boost visual appeal (this included work on the cards, card colors, and background color for stats page, and table and table color for profile page). I also added some functionality to the profile page (rendering and ability for the page to display the user's YouTube ID/username and censored password) and made the 4 test cases.

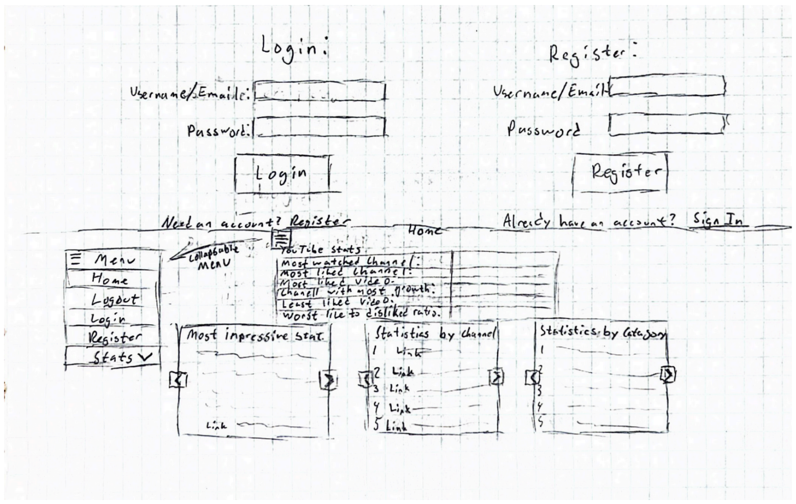
Araceli Luna: In the project, I helped create the functionality of the register page that seamlessly inserts data into a SQL table. I mostly focused on front-end development, aiming to make the project appealing to the client. I helped scale thumbnail images and adjust fonts, and colors on the home page for improved presentation. I selected appropriate logos and favicons to make the website similar to a YouTube page. I also helped redesign the profile page to look similar to the login and register pages.

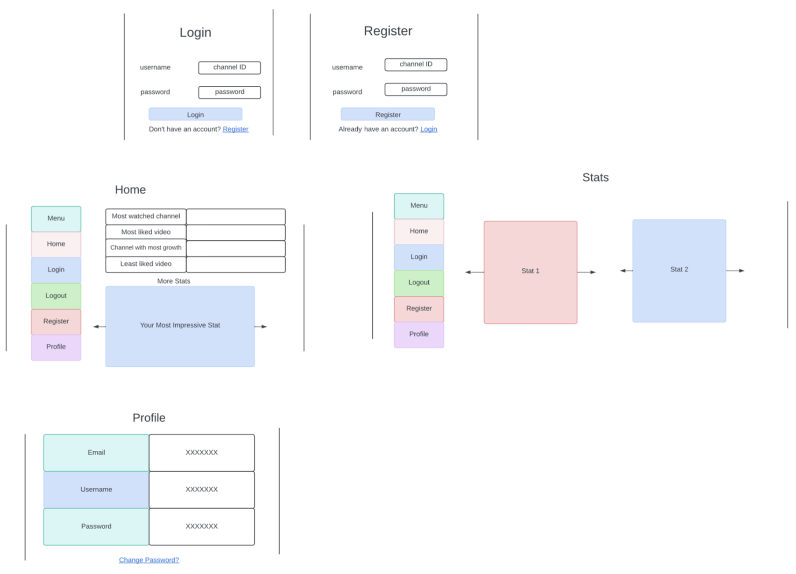
Renata Zurita: The additions I made to the project was connecting the app to Google's console cloud in order to access APIs. I added credentials to the cloud. I also specified the APIs consent screen. I adjusted the user type to external, added authorized redirect JS origins and page URIs and I added the non-sensitive and sensitive scopes of the OAuth key to our application. I integrated the client secret into the API calls and backend functions. I also integrated our cloud console with Firebase. I overall made edits to the backend of our app.

Use Case Diagram


Use case shapes
User
Use Case
Use Case
Use Case
Use Case
System
Most Liked Video
yt API will send stats back to app for user to see
Relationship
Last Viewed Video
Most watched viideo
Most watched channel
Sign in with YouTube ID
User
YT API
YouTube ID pulled from DB and verified as real by YT API
User DB

Wireframes


Login:
Username/Email:
Password:
Login
Need an account? Register
Register:
Username/Email
Password
Register
Already have an account? Sign In
Home
Menu
Home
Logout
Login
Register
Stats
Collapsable Menu
YouTube stats
Most watched Channel:
Most liked Channel:
Most liked video:
Chanell with most growth:
Least liked video:
Worst like to disliked ratio.
Most impressive stat.
Link
Statistics by channel
1 Link
2 Link
3 Link
4 Link
5 Link
Statistics by Category
1
2
3
4
5



Test Results

Feature 1: Login Page

- Test Case: When the user enters a username and password that has been registered, the log in should work and redirect the user to the home page of their account. If the user enters an incorrect username or password, the log in button will not function and an error message will be displayed.
- Redirects properly if correct credentials are provided, If credentials are provided that are not in database redirects to the register page, if incorrect password then page does not redirect and allows re-entering of credentials.

Feature 2: Stats Page

- Test Case: When the user navigates to the stats page it should propagate with information about specific channels that you have watched the most.
- Stats page is properly generated using info from the API, this includes; most watched channel, most watched video, their first video watched, last video watched, and most liked video watched.

Feature 3: Home Page

- Test Case: Home Page should display general YouTube statistics for that year. Home page should be able to connect to google through user login and grab data.
- Home page is able to connect to the users information using the API and links are working that take users to videos or the channel. However, users are not able to login through Google as functionality is not available for this.

Deployment

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

If Azure link does not work:

- Open up the GitHub repository in vscode or another IDE
- Make sure you are in the ProjectSourceCode folder (cd ProjectSourceCode) and that Docker is up and running
- Enter “docker compose up” into your terminal
- Go to your web browser and enter “localhost:3000”