YouDash

Group 26

Team Members:
Steven Schonlau
Jose Pagan
Sohaib Qureshi
Abdul Wajid Arikattayil

Table of Contents:

Purpose

- Our generation spends more time than ever on YouTube. While productivity can be gained from this, many spend their time watching what the algorithm recommends, without clear insight into their activity. YouTube has a financial incentive to provide videos that benefit its platform rather than the user's quality of life or goals. We aim to provide transparency along with tools to help users utilize YouTube in more productive ways.

- The purpose of our product is to give users insights into their viewing, provide community for users with common interests, and help users to meet customizable goals. Helping users to reach their goals is accomplished through blocking videos, gamification of goals, and leaderboards. The community will be based on a Reddit style forum, but will be specifically for YouTube viewers, thus meeting a narrower target audience more directly than Reddit. We will give users insights through our website dashboard, which will provide much more than the existing features that YouTube offers. In short, we aim to provide users with an unbiased and informative way to shape their YouTube viewing.

Functional Requirements:

Chrome Extension:

1. As a user, I would like to be redirected to the YouDash website when I want to register for a YouDash account.
2. As a user, I would like to login into my YouDash account on the Chrome extension by being redirected to the YouDash website.
3. As a user, I would add or remove a category of videos from my blocked videos.
4. As a user, if I go to the blocked category of videos, it would redirect me to a screen reminding me to stay focused.
5. As a user, I would like the chrome extension to send my watch history to the YouDash backend.

Profile/Usage Features:

6. As a user, I want to have login functionality that is synchronized across devices.
7. As a user, I want to have signup functionality that is synchronized across devices.
8. As a user, I want to be logged into the chrome extension if I log into the website
9. As a user, I can set the goal of minimizing my watch time on youtube
10. As a user, I can set the goal of improving the quality of videos I watch (improve productivity)
11. As a user, I would like to be able to register for a YouDash account using an email and password.
12. As a user, I would like to be able to register for a YouDash account using Google.
13. As a user, I want to be able to change my profile settings of dark mode
14. As a user, I want to be able to change my profile settings of profile picture
15. As a user, I want the app to have responsive design.
16. As a user, I want the ability to change account settings of email
17. As a user, I want the ability to change account settings of bio
18. As a user, I want to be able to reset my password if needed.

Social Features:

19. As a user, I want to be able to view public profiles and analytics of people I follow to be able to compare my progress with others and gain inspiration.
20. As a user, I want to be able to share an image of my progress to my followers to celebrate the progress I've made.
21. As a user, I want to see weekly leaderboards of watch time per category of video so that I can be motivated socially to work towards my goals if they include watch time.
22. As a user, I want to see weekly leaderboards of goal progress so that I can be motivated socially to meet my goals.
23. As a user, I want to join a per interest group leaderboard
24. As a user, I want to join an overall leaderboard group.
25. As a user, I want to be able to join interest communities so that I can be socially motivated to meet my goals and interact with other users.
26. As a user, I want to be matched with interest communities that share interests in watch categories so that I can join these groups easier
27. As a user, I want to be matched with users that share interests in watch categories so that I can follow these people easier
28. As a group manager, I want to be able to create interest groups
29. As a group manager, I want to be able to invite users in order to moderate communities
30. As a group manager, I want to be able to kick users in order to moderate communities
31. As a group manager, I want to configure the group setting community name
32. As a group manager, I want to configure the group setting description
33. As a group manager, I want to configure the group setting picture
34. As a group manager, I want to be able to delete comments made on posts.
35. As a group manager, I want to be able to give announcements in order to moderate communities
36. As a user, I want to be able to view announcements of the groups I am in
37. As a user, I want to post videos in forums both in and out of communities in order to facilitate discussion
38. As a user, I want to post video timestamps in forums both in and out of communities in order to facilitate discussion

39. As a user, I want to post comments in forums both in and out of communities in order to facilitate discussion
40. As a user, I want to be able to up/down vote comments in order to facilitate discussion
41. As a user, I want to be able to up/down vote videos in order to facilitate discussion
42. As a user, I want to direct message another user
43. As a user, I want to be able to search messages and users to facilitate the community and social organization.
44. As a user, I want to have the option to reformat my messages using AI in order to communicate messages more clearly.
45. As a user, I want to be able to block someone's direct messages
46. As a user, I want to be able to follow someone
47. As a user, I want to be able to unfollow someone
48. As a user, I want to view a list of my followers
49. As a user I want to view a list of those I follow
50. As a user I want to be recommended users to follow based on my followers and followed users networks
51. As a user I want to be able to choose to create a public and private account

Analytics Features:

52. As a user, I want to be able to set goals like limiting my watch time, watching specific ratios of categories or restricting my watching to certain times of the day, in order to better manage my watching habits.
53. As a user, I want to be able to view my Youtube analytics with the help of visual graphics and tables so I can easily view my progress towards my set goals.
54. As a user I should be able to view my data daily/ weekly/ or annually using a heat map
55. As a user, I want to be able to see a month to month comparison of my watchtime, in order to see my progress over time.
56. As a user, I want to be able to view my progress towards my watchtime goals in a pie chart so that I can easily track the percentage of my progress.

57. As a user, I want my YouTube analytics to be viewed over multiple customizable time frames.
58. As a user, I want to see my goal progress through a customizable visual of a cactus growing to motivate meeting my goals.
59. As a user I want to be able to share my profile using the website link (to show my progress)
60. As a user I want to be able to share my profile by downloading an image of my progress.
61. As a user, I want to see AI recommendations on demand on how to meet my goals so that I can gain insights and motivation to meet my goals.
62. As a user, I want to be able to export my data and dashboard as a csv in order to have more portability
63. As a user, I want to be able to export my data and dashboard as a pdf in order to have more portability

Nonfunctional Requirements:

Architecture and Performance:

We're using React for the frontend of our web application, while Spring Boot handles the backend, managing HTTP protocols sent from the frontend. Although Spring Boot is a robust framework, which might seem excessive for a small-scale application, it offers the flexibility to scale with multiple developers in the future. Additionally, we plan to implement a secure login feature using a Google API to ensure user safety. We'll also utilize AWS tools such as DynamoDB for storing user information and EC2 for running the application. Once user data is stored in DynamoDB, a recommendation by Open-AI API will be generated on-demand of the user based on the click of a button..

Security:

The secure storage of users' credentials is a top priority as we'll store users' passwords into our database. We will salt/hash the passwords and then store it inside DynamoDB which is a trusted database. Users will only have access to their own data, with the option to share insights publicly, if they choose to do so. Permission settings will be clearly defined to maintain privacy.

Usability:

The UI should be easy to use and navigate for the users. The app will have smooth transitions between screens like logins and analytics charts to allow the best experience for the users. We will include customizable display options like dark mode, adjustable font sizes, and responsible design for optimal viewing across different devices.

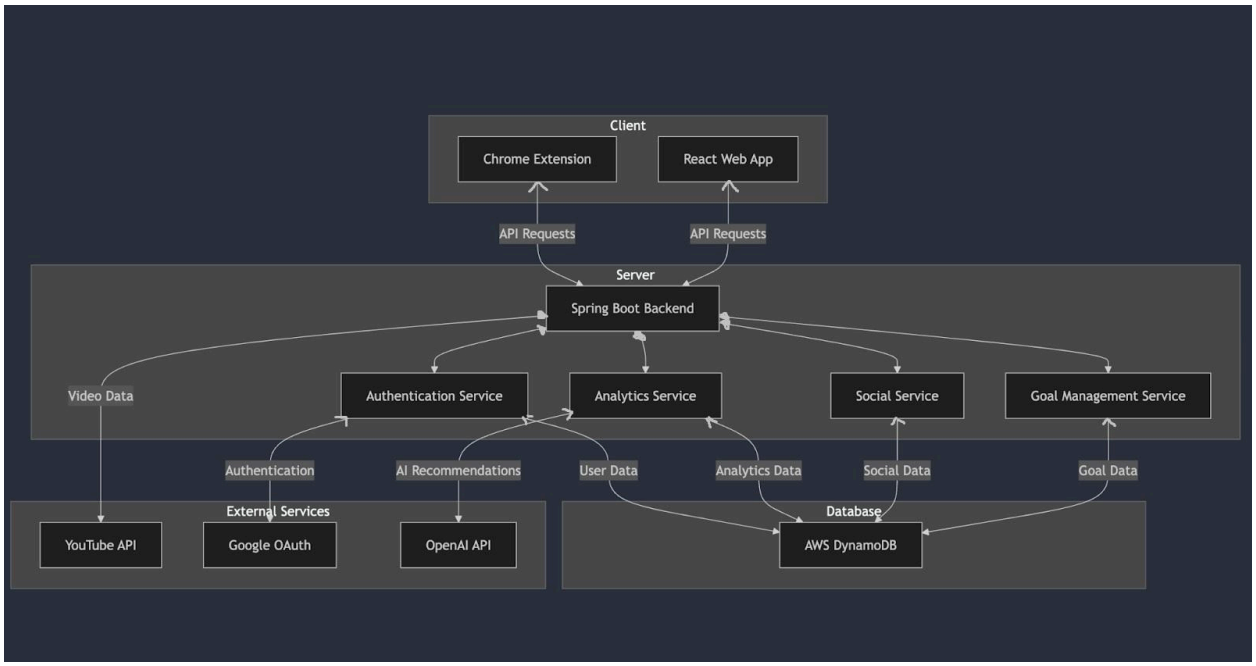
Hosting/Deployment:

Hosting and deployment can be done on AWS EC2 instance. We would choose the t2.micro instance which is given on the AWS Free Tier will be able to cover all of the

functionalities this project needs by balancing out the number of users entering the website and how fast the data is retrieved and added.

The free 1 year plan of AWS includes 25 GB on indexed data storage each month. Also, it has the ability to do 2.5 million read requests and 1 million write requests.

Design Outline:



1. Client - Server Interaction:
    - The client (React Web App and Chrome Extension) communicates with the server using RESTful API calls allowing the client to send requests and receive responses from the server.
    - The client sends user actions, such as login attempts, goal settings, or analytics requests to the server.
    - The server processes these requests and sends back appropriate responses, such as authentication tokens, user data, or analytics results.
2. Server - NoSQL Database Interaction:
    - The server interacts with the NoSQL Database (AWS DynamoDB) using NoSQL queries.
    - The database responds with the requested data or confirmation of successful data storage.
3. Client - YouTube API Interaction:

- The client through the server, particularly the Spring Boot Backend, interacts directly with the YouTube API.
- It sends REST API requests to fetch video data, such as video details, categories, or watch time.
- The YouTube API responds with the requested information, which the client can then use or send to the server for processing.

4. Client - Google OAuth Interaction:
    - The client interacts with Google OAuth through the backend for user authentication.
    - When a user chooses to log in with Google, the client queries the backend for the OAuth flow.
    - Google OAuth response with authentication tokens, which the server can then use to authenticate with the database credentials.
5. Server - OpenAI API Interaction:
    - The server interacts with the OpenAI API to generate AI recommendations.
    - When a user requests AI-powered insights or recommendations, the server sends relevant data to the OpenAI API.
    - The OpenAI API processes this data and returns AI-generated recommendations, which the server can then send back to the client.
6. Chrome Extension - React Web App Interaction:
    - While not directly connected, these two client components can indirectly interact through the server.
    - For instance, watch history data collected by the Chrome Extension can be sent to the server, which then makes it available for viewing in the React Web App.

Design Issues:

Functional Issues:

How should the user be able to view their usage?

- Line graph
- Pie graph
- Heatmap
- Daily, weekly, or monthly views of above?
- Month to month comparison
- Combination of above?

Choice: All the above (time permitting)
Justification: We decided to let the user view all of the above (time permitting) as each graph will allow us to give more insights to the viewer, allowing us to meet our purpose more fully. The only disadvantages to having more is the added time of creating the charts and having more complex code. Having the user be overloaded with information shouldn't be an issue as long as it is organized in a viewer friendly way.

How should groups be created?

- Based on categories
- Based on specific videos
- Based on moderators creating a group

Choice: Based on moderators, with the option to select a category
Justification: We felt that this option has the most flexibility for the types of groups users could create as a moderator, but still allow us to recommend groups based on category. The only downside is that it is more work to create/find a group based around a video, because a moderator has to create the group.

How should leaderboards be created?

- Based on goals

- Based on categories
- Based on groups
- Combination of above

Choice: All of the above (time permitting)
Justification: We felt that this choice only adds more to the leaderboards and allows us to meet our purpose more fully. The only downside to having leaderboards with more metrics is if it becomes unorganized, but we will address this proactively.

How should a user log into the Chrome extension?

- Redirected to website to log in and use API key afterwards
- Log in through chrome extension and access database directly

Choice: Redirected to website to log in and use API key afterwards
Justification: We chose this because it reuses the existing functionality of the login mechanics. The only work required is to create an API key between the chrome extension and the server to allow for transfer of user data to be stored in the database. The only disadvantage is this extra work and the added complexity of having the website as a middleman between the chrome extension and the database.

Non-Functional Issues:

What framework should be used?

- React
- Vue.js
- Flutter

Choice: React

Justification: We chose React because Vue has fewer plugins and Flutter uses Dart which is a less-used language compared to JavaScript. We feel that React will leverage some of the groups' existing react experience. The only disadvantage is React is more complex than we need, but as a team we would like more experience with it.

How should the application obtain user's watch history?

- Chrome extension
- YouTube API
- Web Scraping

Choice: Chrome Extension

Justification: We chose to use a Chrome Extension to filter, copy, and send the user's YouTube usage in real time because the YouTube API v3 stopped supporting accessing a user's watch history, and the Chrome Extension would be simpler than Web Scraping the user's history, which would require storing their login (which could be google login, email login, etc). The only downsides to this is that it only records after they install it, and it only would work for Chrome users. However, we feel that this is outweighed by the relative complexities of the other approaches.

What data should be stored for each user in the database?

- Youtube URL
- All video information (JSON)

Choice: All video information (JSON)

Justification: Storing all of the videos information as a JSON would decrease the amount of API calls and allow for faster data analysis. This is important in order to meet our requirement of response time. The downside is that this requires more space, but since our first priority is our customers we feel this is outweighed by the benefits.

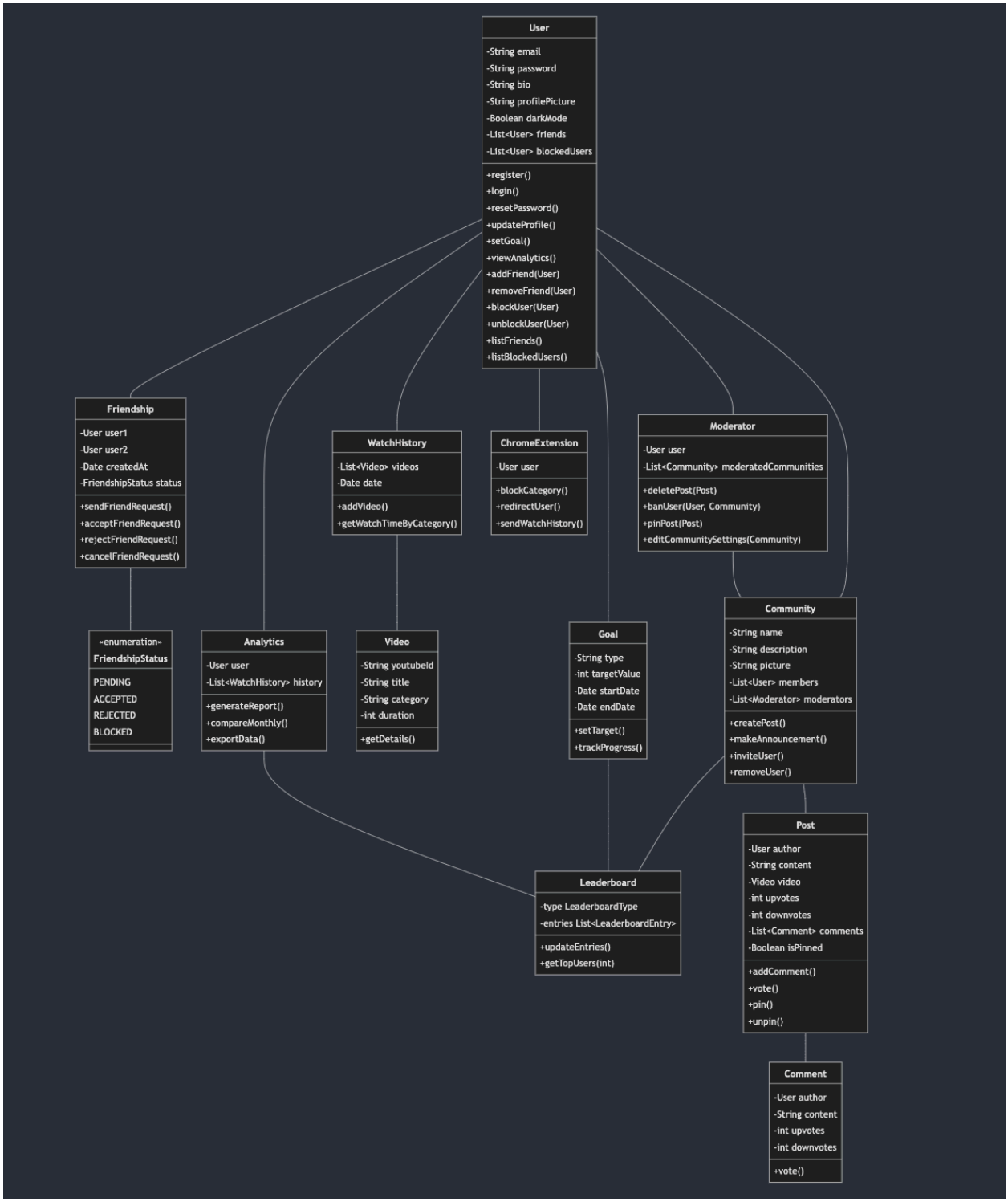
What type of database should be used?

- Relational
- Non-Relational
    - Document Data Store
    - Key Value Stores

Choice: Document Data Store

Justification: Since we are storing video information as a JSON, it makes the most sense and is the easiest to access through a Document Data Store. Having a Key Value store would be unnecessary work and flexibility, and a relational database would also be more work to convert from JSON. However, one of the disadvantages would be that it would be slightly slower to access data compared to a relational database.

Design Details



User:

- Attributes: username, email, password, profilePicture, bio, watchHistory, goals, blockedCategories
- Methods: setGoal, updateProfile, blockCategory, unblockCategory, getAnalytics, addToWatchHistory
- Interactions: Has many Videos (watchHistory), many Goals, one Analytics, belongs to many Communities, has one ChromeExtension

Video:

- Attributes: videoId, title, category, duration, watchedDuration
- Methods: getCategory, updateWatchedDuration, isCompleted
- Interactions: Belongs to User's watchHistory

Goal:

- Attributes: type, target, progress, startDate, endDate
- Methods: updateProgress, checkCompletion, getProgressPercentage
- Interactions: Belongs to a User

Analytics:

- Attributes: user, watchTimeData, categoryData, goalProgressData
- Methods: generateReport, exportData, getWatchTimeStats, getCategoryStats, getGoalProgressStats
- Interactions: Belongs to a User

Community:

- Attributes: name, description, members, posts, managers
- Methods: addMember, removeMember, createPost, addManager, removeManager
- Interactions: Has many Users (members), many Posts

Post:

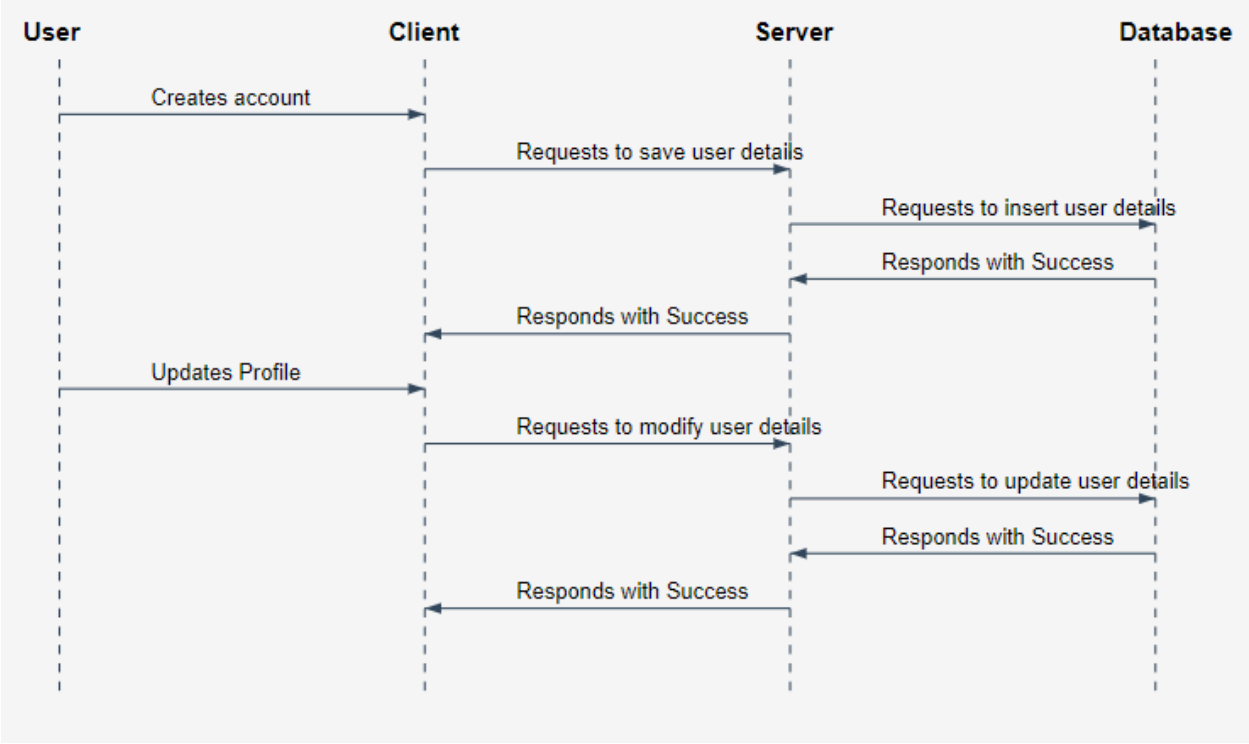
- Attributes: id, author, content, timestamp, comments, votes
- Methods: addComment, upvote, downvote
- Interactions: Belongs to a Community, authored by a User

Leaderboard:

- Attributes: type, entries, timeFrame
- Methods: updateEntries, getTopUsers, getUserRank
- Interactions: Contains many Users

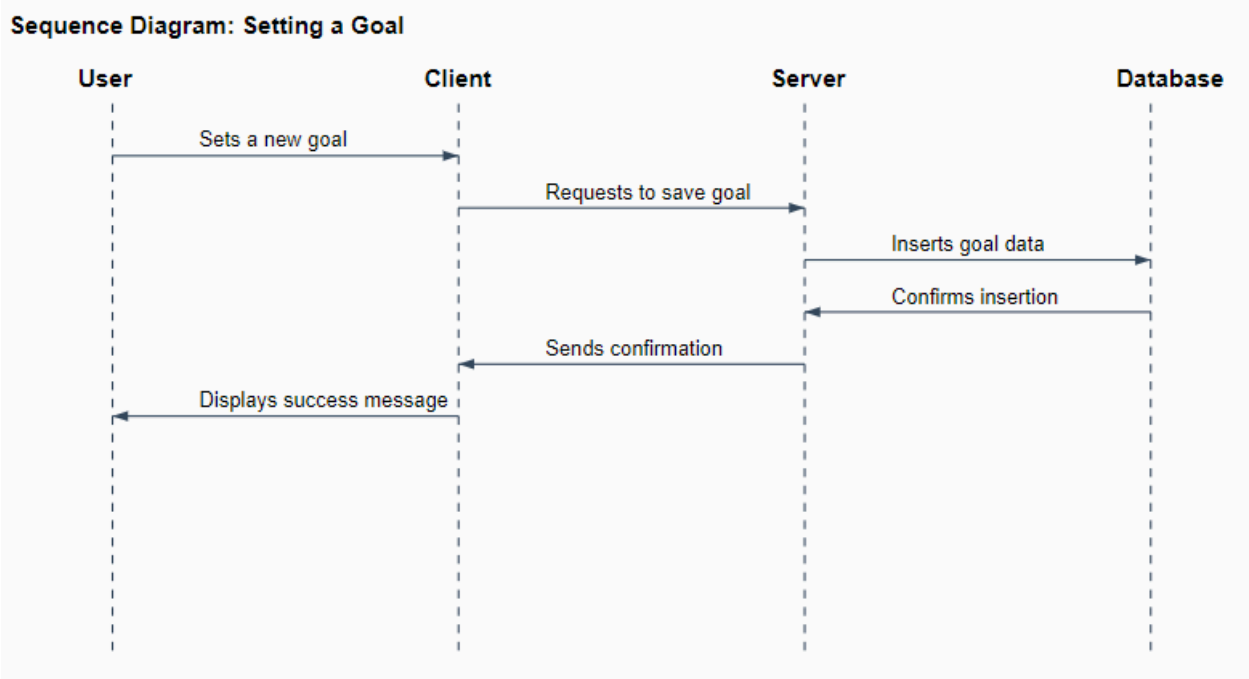
ChromeExtension:

- Attributes: user, isActive
- Methods: redirectUser, sendWatchHistory, toggleActive, isVideoBlocked
- Interactions: Belongs to a User

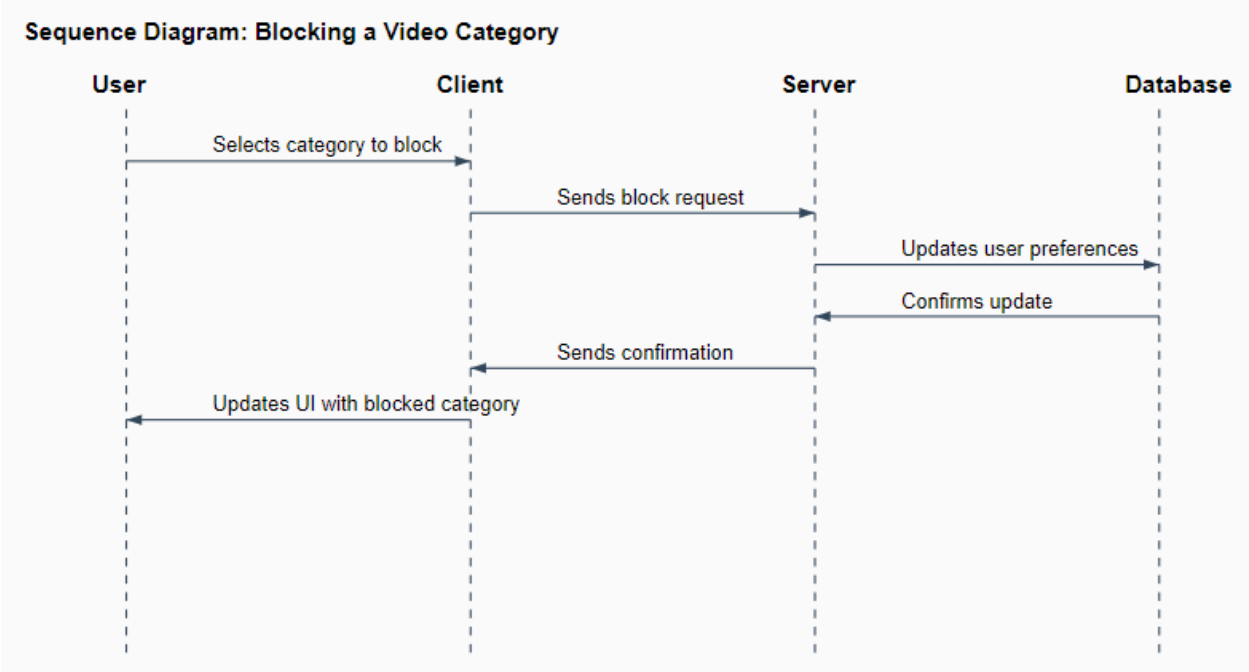


In these sequence diagrams we have shown the interaction between the user, client (website), server, and database. The sequence begins when the user creates a new account or signs into their YouDash account; this will be given to the client which is then handed to the server so it can

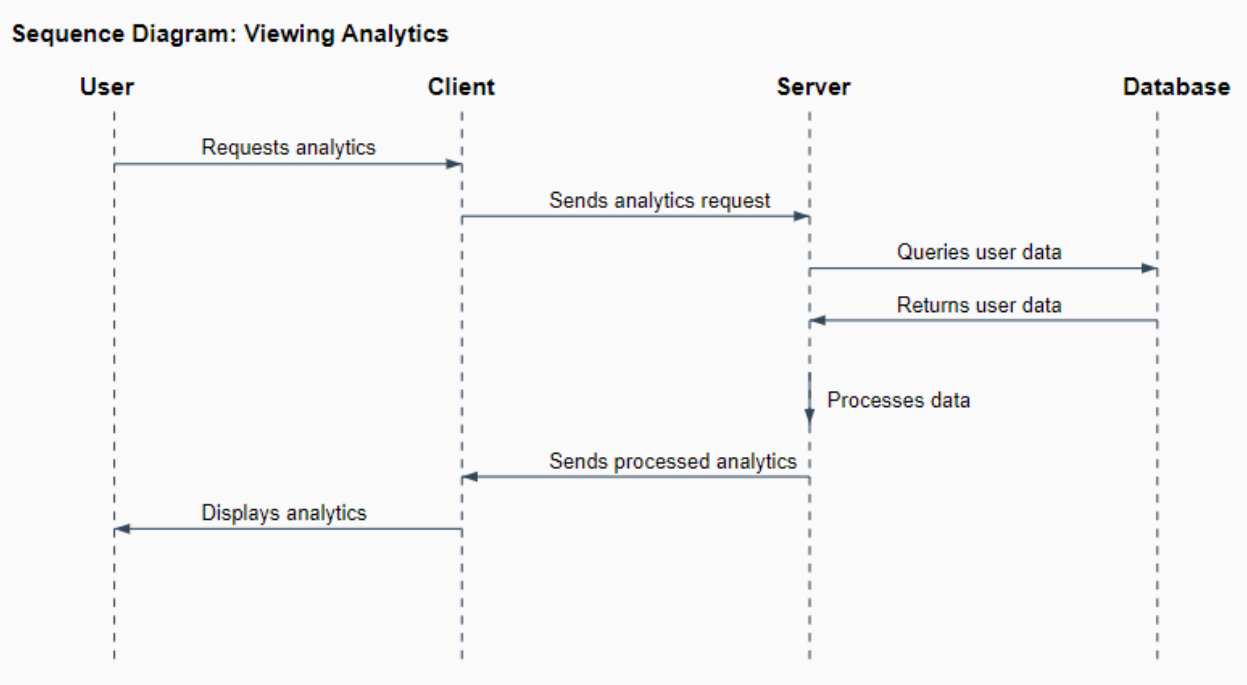
create a query for getting the users information from the database. Once the account has been created, the user can choose to add or modify goals, block video categories, view their watch history, participate in communities, create their weekly watch plan, and view their analytics. Each addition or modification will start at the client and then will be sent as a request to the server which creates a query that requests to store or change information in the database. Once the database has received its new data, a success message will be displayed on the client side. Whenever the client needs to retrieve information, the request goes through the server which queries the database and returns the resulting data



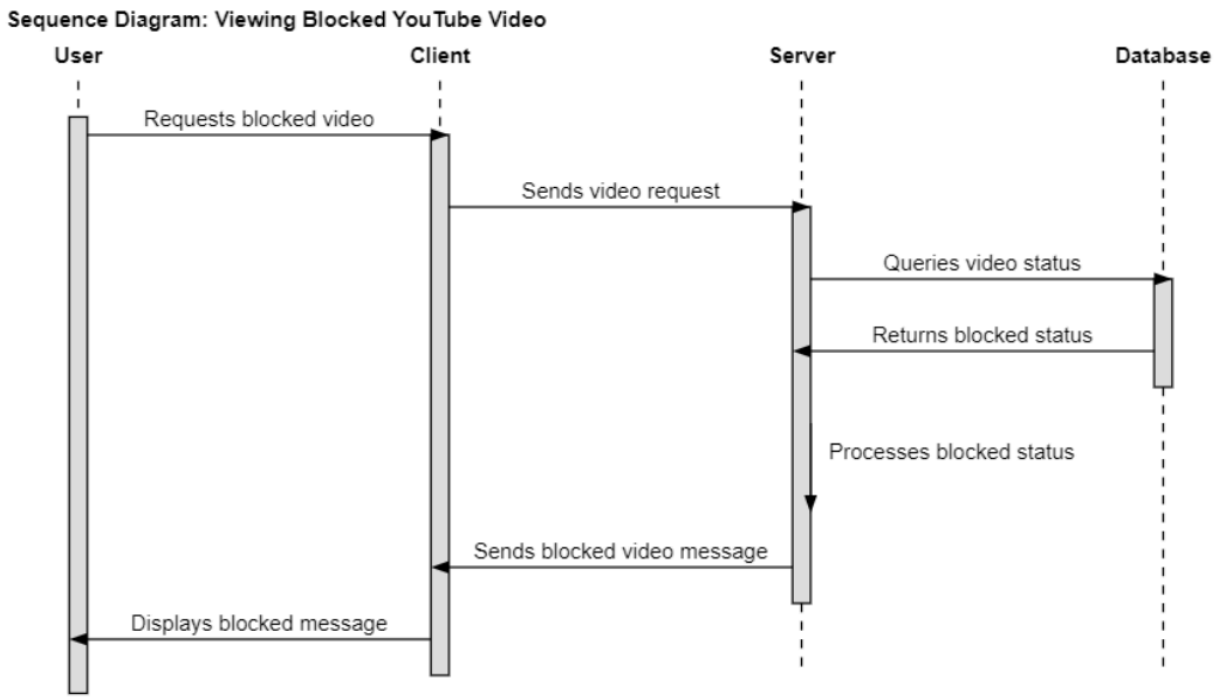
Setting a Goal: This sequence diagram illustrates the process of a user setting a new goal in YouDash. The user initiates the action through the client interface. The client then sends a request to the server to save the goal. The server processes this request and inserts the goal data into the database. Once the database confirms the successful insertion, the server relays this confirmation back to the client. Finally, the client displays a success message to the user, completing the goal-setting process. This flow ensures that the user's goal is properly recorded and stored in the system, allowing for future tracking and analysis.



Blocking a Video Category: The sequence for blocking a video category begins with the user selecting a category to block via the client interface. The client sends this block request to the server, which then updates the user's preferences in the database to include the newly blocked category. After the database confirms the update, the server sends a confirmation back to the client. The client then updates its user interface to reflect the newly blocked category, providing immediate feedback to the user. This process allows users to customize their YouTube experience by filtering out unwanted content categories, a key feature of the YouDash application.



Viewing Analytics: The analytics viewing process starts when a user requests to see their analytics through the client interface. The client forwards this request to the server, which then queries the database for the relevant user data. Once the database returns the requested data, the server processes this raw information into meaningful analytics. The processed analytics are then sent back to the client, which displays them to the user in an understandable format. This sequence demonstrates how YouDash provides users with insights into their YouTube usage patterns, helping them understand their viewing habits and track progress towards their goals.

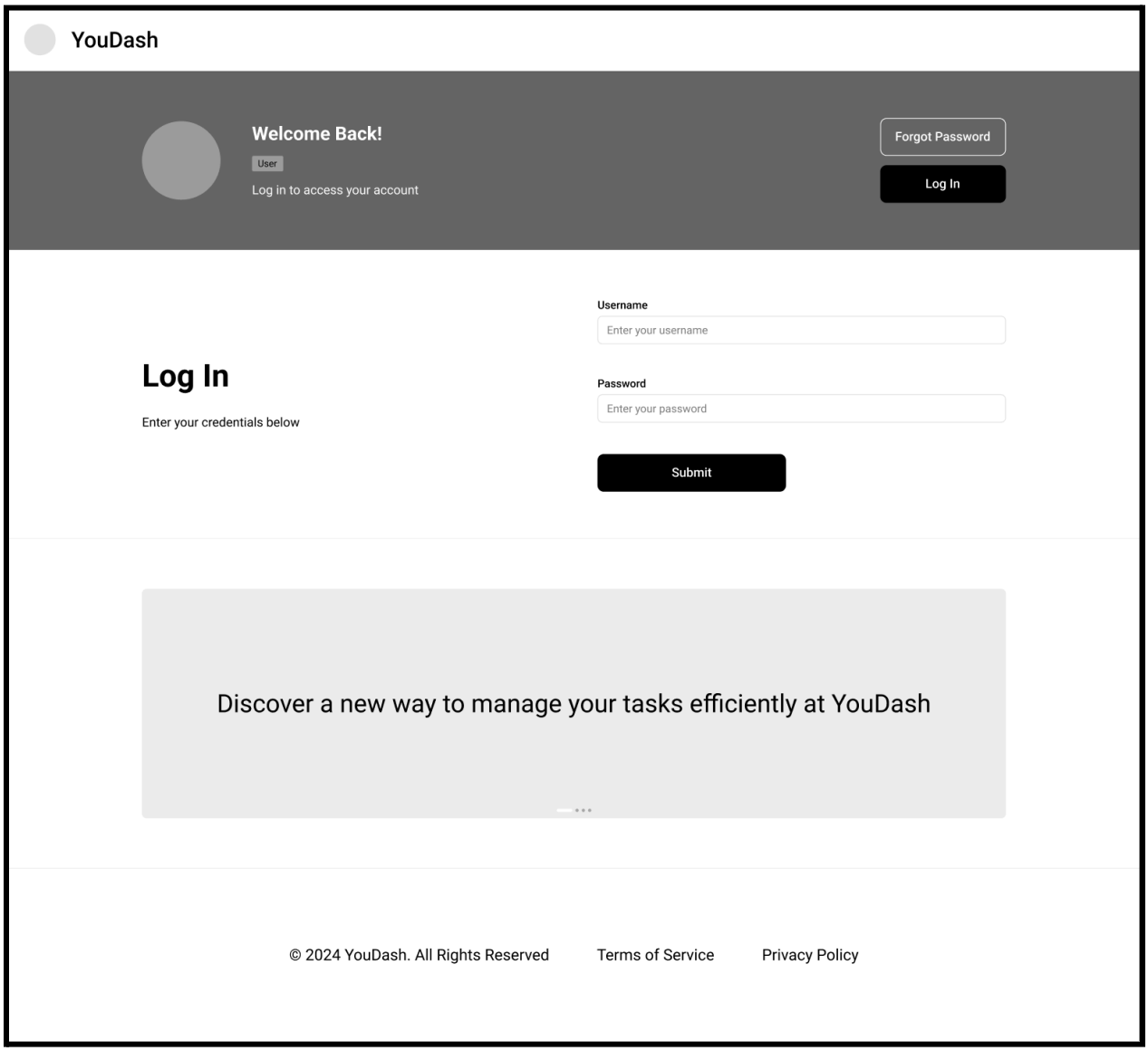


Viewing Blocked YouTube Video:

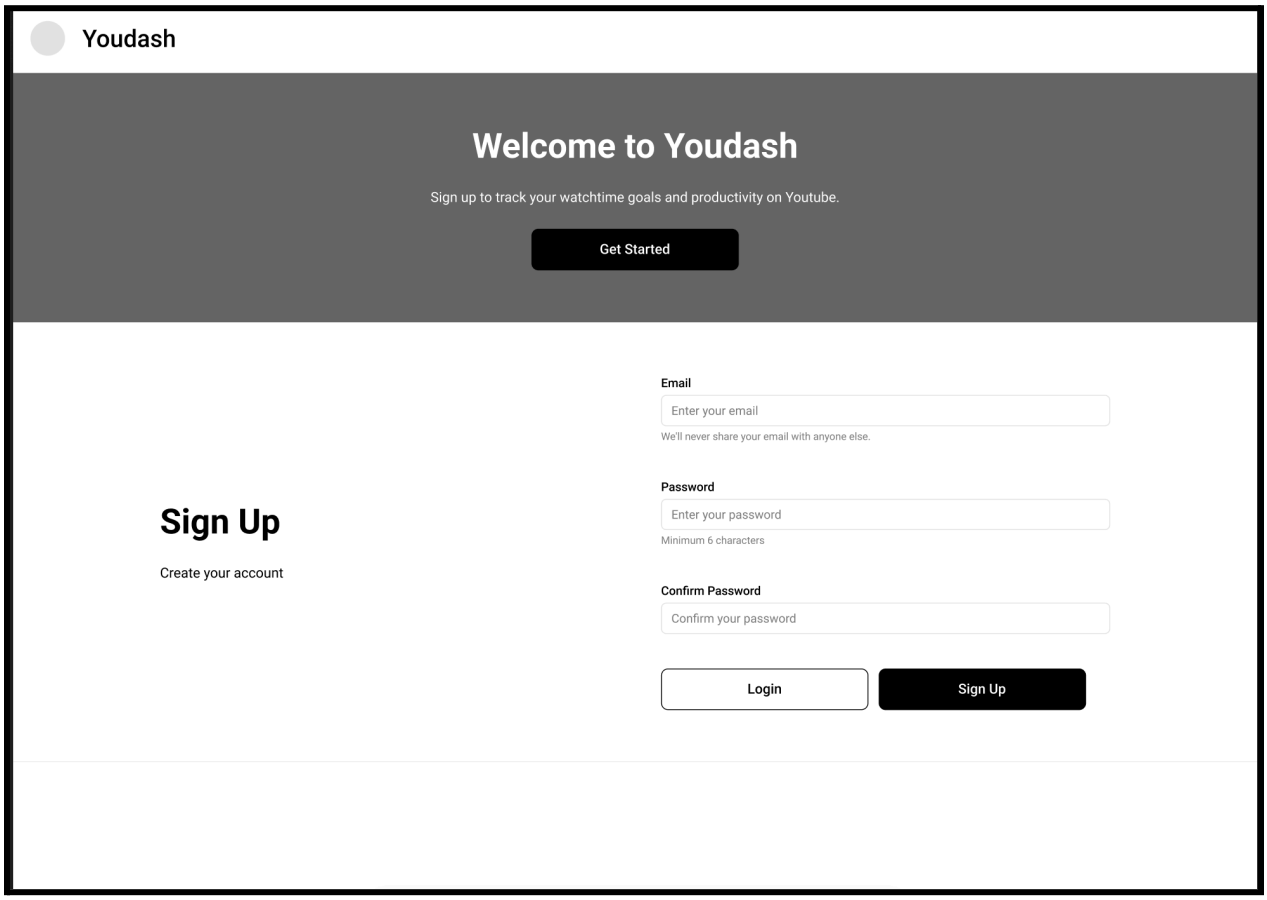
User requests a blocked video from the client. The client the sends this request to the server. The server queries the database to check the video's status, which returns that the video is blocked. The server processes this information internally, then sends a message back to the client indicating that the video is blocked. Then finally, the client displays this blocked message to the User.

UI Mockups

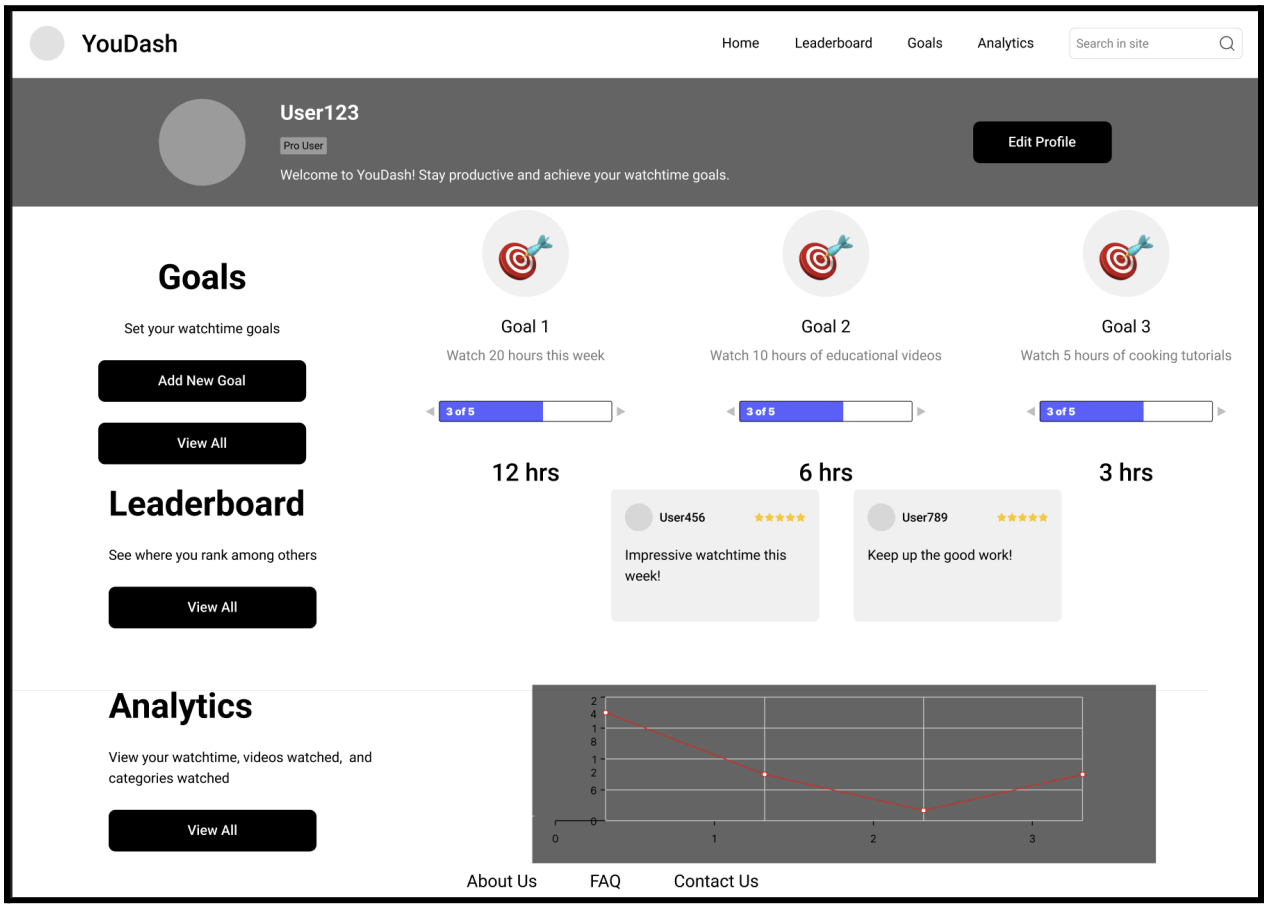
Login Page



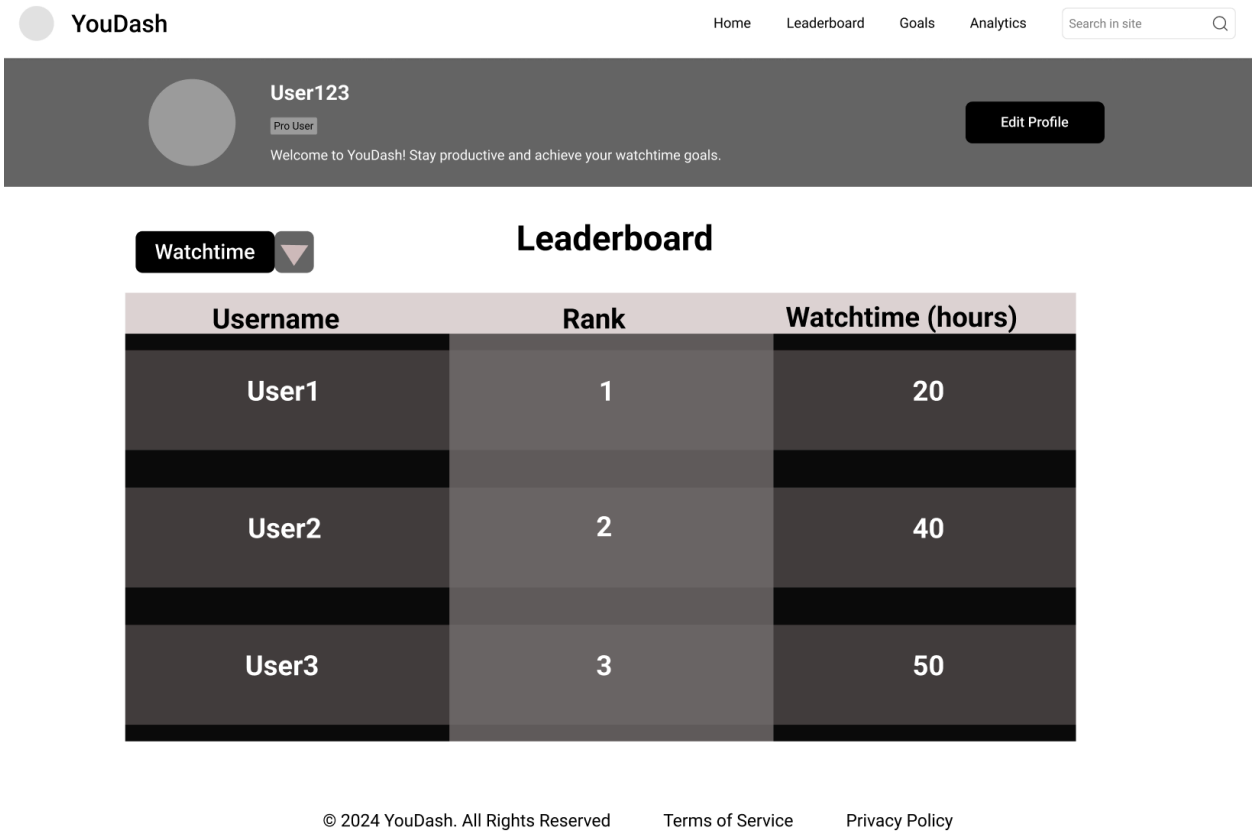
Sign Up Page


Youdash
Welcome to Youdash
Sign up to track your watchtime goals and productivity on Youtube.
Get Started
Sign Up
Create your account
Email
Enter your email
We'll never share your email with anyone else.
Password
Enter your password
Minimum 6 characters
Confirm Password
Confirm your password
Login
Sign Up

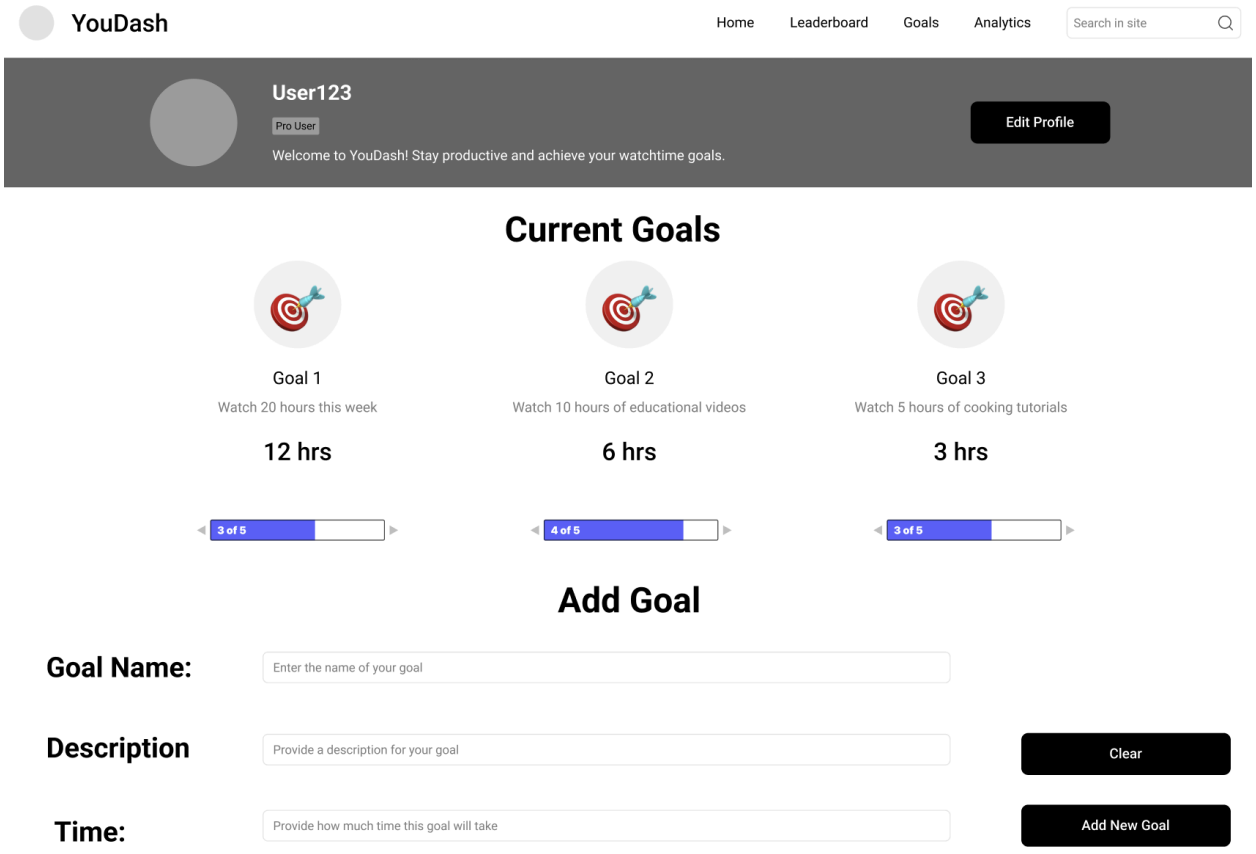
YouDash Dashboard



Leaderboard Page



Goal Page


YouDash
Home
Leaderboard
Goals
Analytics
Search in site
User123
Pro User
Welcome to YouDash! Stay productive and achieve your watchtime goals.
Edit Profile
Current Goals
Goal 1
Watch 20 hours this week
12 hrs
3 of 5
Goal 2
Watch 10 hours of educational videos
6 hrs
4 of 5
Goal 3
Watch 5 hours of cooking tutorials
3 hrs
3 of 5
Add Goal
Goal Name:
Enter the name of your goal
Description
Provide a description for your goal
Clear
Time:
Provide how much time this goal will take
Add New Goal

Analytics Page