

VIETNAM INTERNATIONAL UNIVERSITY – HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

WEB APPLICATION DEVELOPMENT

PROJECT VIDEO PLATFORM

By

Vo Hoai Bao – ID: ITITIU21038

Nguyen Phuc Vinh – ID: ITITIU21350

Advisor: Dr. Nguyen Van Sinh

Advisor: Nguyen Trung Nghia

A report submitted to the School of Computer Science
and Engineering in partial fulfillment of the requirements
for the Final Project

in Web Application Development course – Summer 2024

Ho Chi Minh City, Vietnam, 2024

OUTLINE

I. INTRODUCTION:	1
1. ABOUT US:.....	1
2. THE PRODUCT'S INFORMATION:.....	1
3. WORK BREAKDOWN STRUCTURE:	2
4. DEVELOPMENT ENVIRONMENT:	5
II. REQUIREMENT ANALYSIS AND DESIGN:	7
1. REQUIREMENT ANALYSIS:	7
A. FUNCTIONAL REQUIREMENTS:.....	18
B. NON-FUNCTIONAL REQUIREMENTS	23
2. DESIGN:	25
A. ERD Diagram:	25
B. Relational Schema Diagram:.....	26
C. Class-Diagram:.....	27
a. Front-End:.....	27
b. Back-End:	30
c. Logic flow of backend API:.....	36
D. User Case Diagram:	37
E. API:	37
a. Front End:.....	37
b. Back End:	39
III. IMPLEMENTATION:	41
1. Account management:	41
1.1. Login to the application:	41
1.2. Register:	44
1.3. Forgot password:	49
1.4. Account information:.....	52
2. Video management:	54
2.1. Upload video:.....	54

2.2. Mange uploaded video's user:.....	59
2.3. Watching the video:.....	60
2.4. User watching history:	63
3. Search engine:	64
4. Admin role	66
4.1. Manage role for users:	66
4.2. Delete users:	68
4.3. Review upload videos:	69
5. Root role	72
IV. DISCUSSION AND CONCLUSION	73
V. References:	73
1. Front End	73
2. Back End.....	73

I. INTRODUCTION:

In this segment, the essence of our project software development collective is introduced, uTuBe. We delve into the foundational aspects of our endeavor – a video platform webpage project – and articulate the challenges encountered throughout its development journey.

1. ABOUT US:

We, a duo of tech enthusiasts, have joined hands to bring to life a video platform project. Our collaboration harnesses our collective skills and vision, resulting in a user-centric digital area for sharing and engaging with video content.

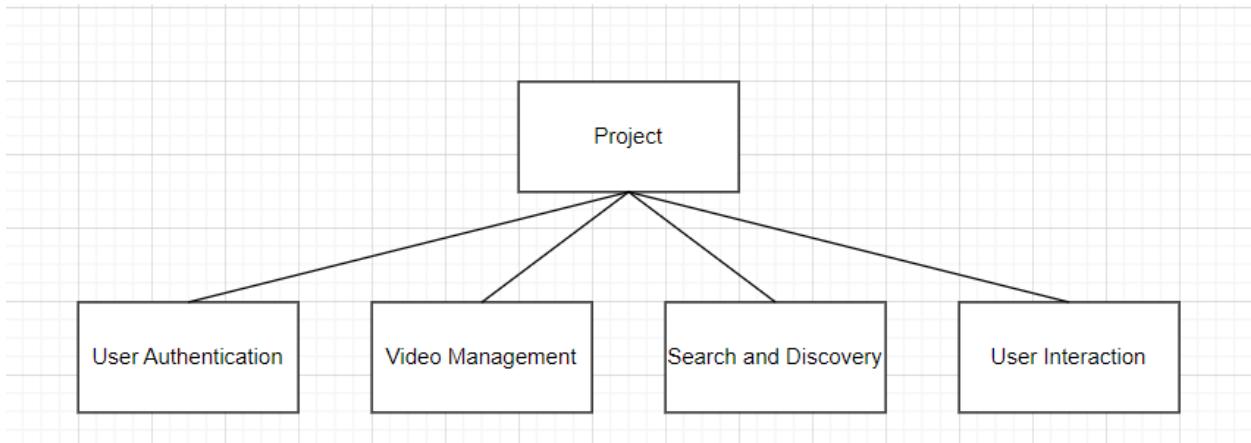
2. THE PRODUCT'S INFORMATION:

In today's digital age, video content is popular, but the vastness of available platforms often leaves users overwhelmed and content creators struggling to stand out. The challenge lies in finding a balance between feature-rich platforms and user-friendly interfaces that cater to diverse audiences and empower creators.

Our proposed solution is a video platform webpage named uTuBe, designed with meticulous attention to addressing these challenges. By delivering a streamlined and intuitive user experience combined with robust functionality, uTuBe aims to foster an environment conducive to creativity and connectivity. This initiative is driven by our commitment to democratizing video sharing and establishing a global community where individuals can share their experiences and perspectives with the world.

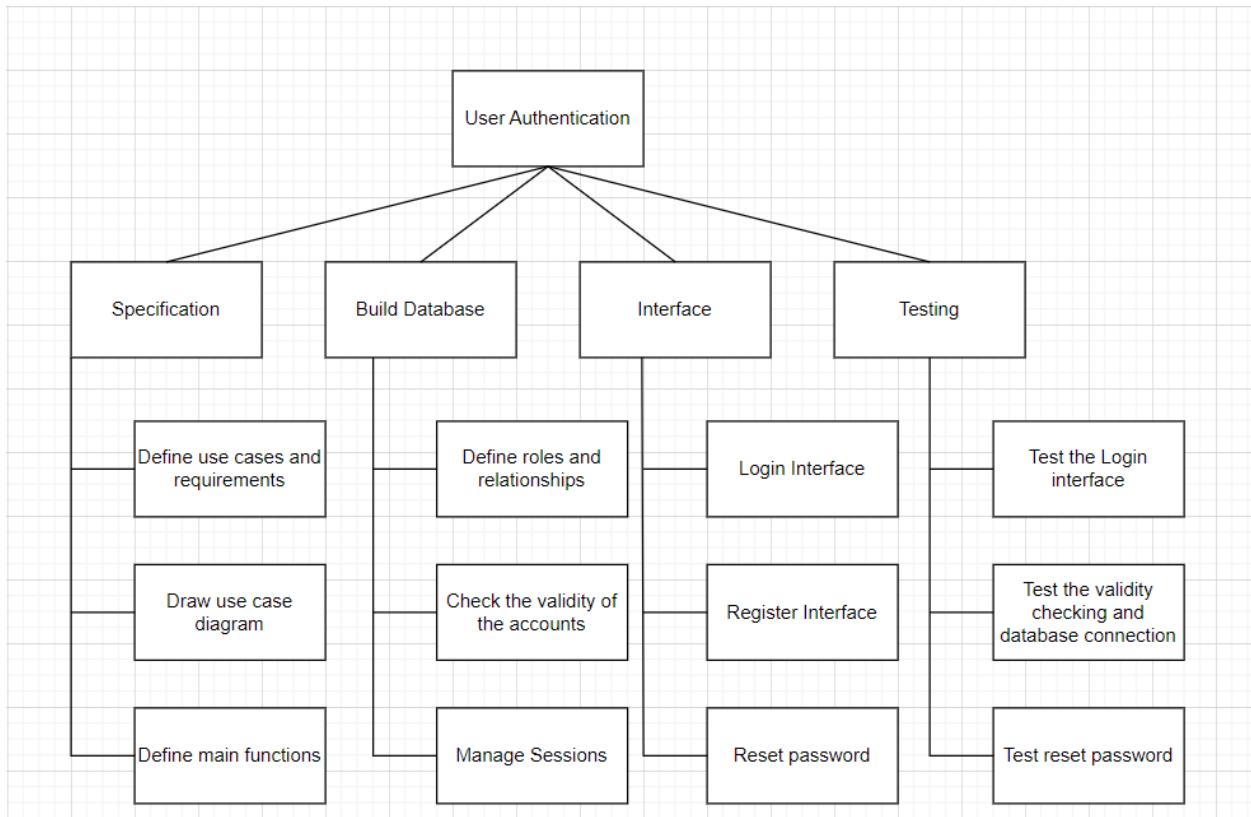
3. WORK BREAKDOWN STRUCTURE:

The structure of project can be divided into 4 parts in the Figure 3.1:

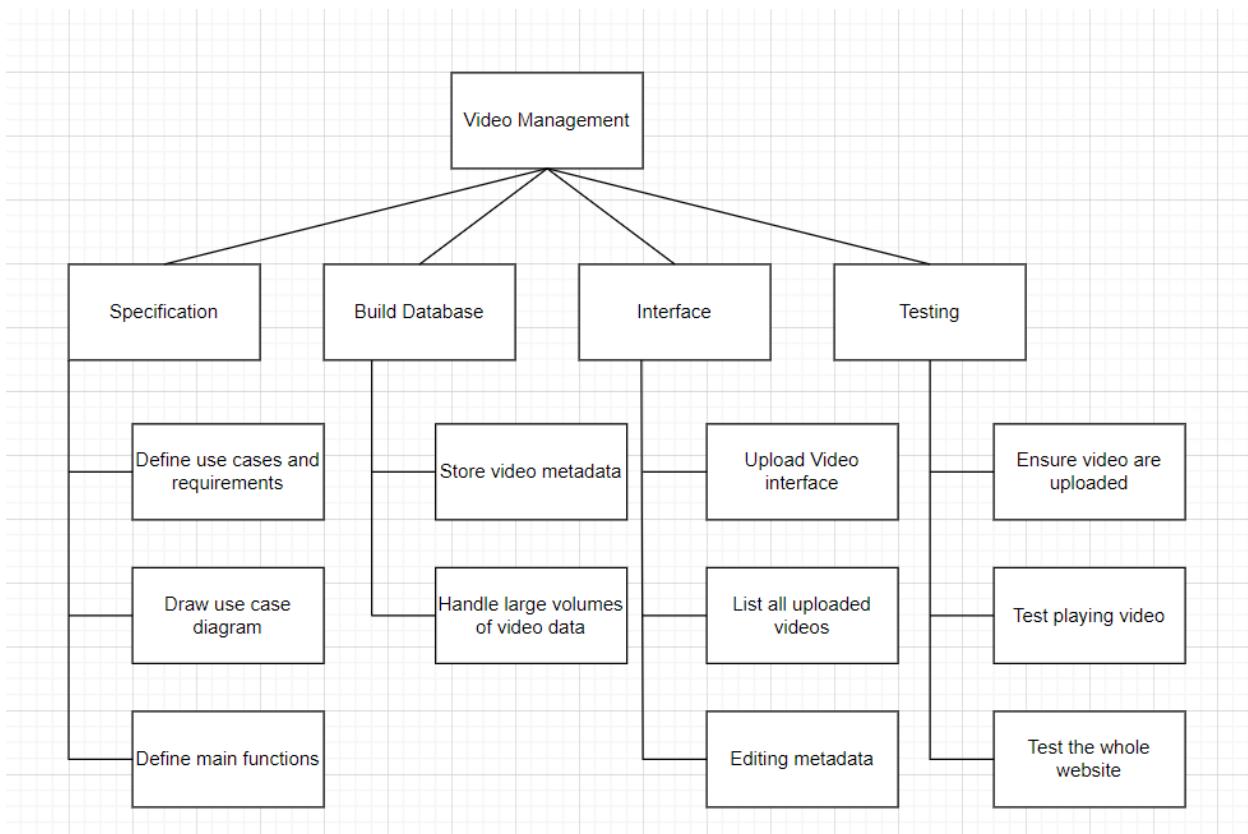


In each branch of this tree, subtrees were also provided to describe the tasks needed to be accomplished.

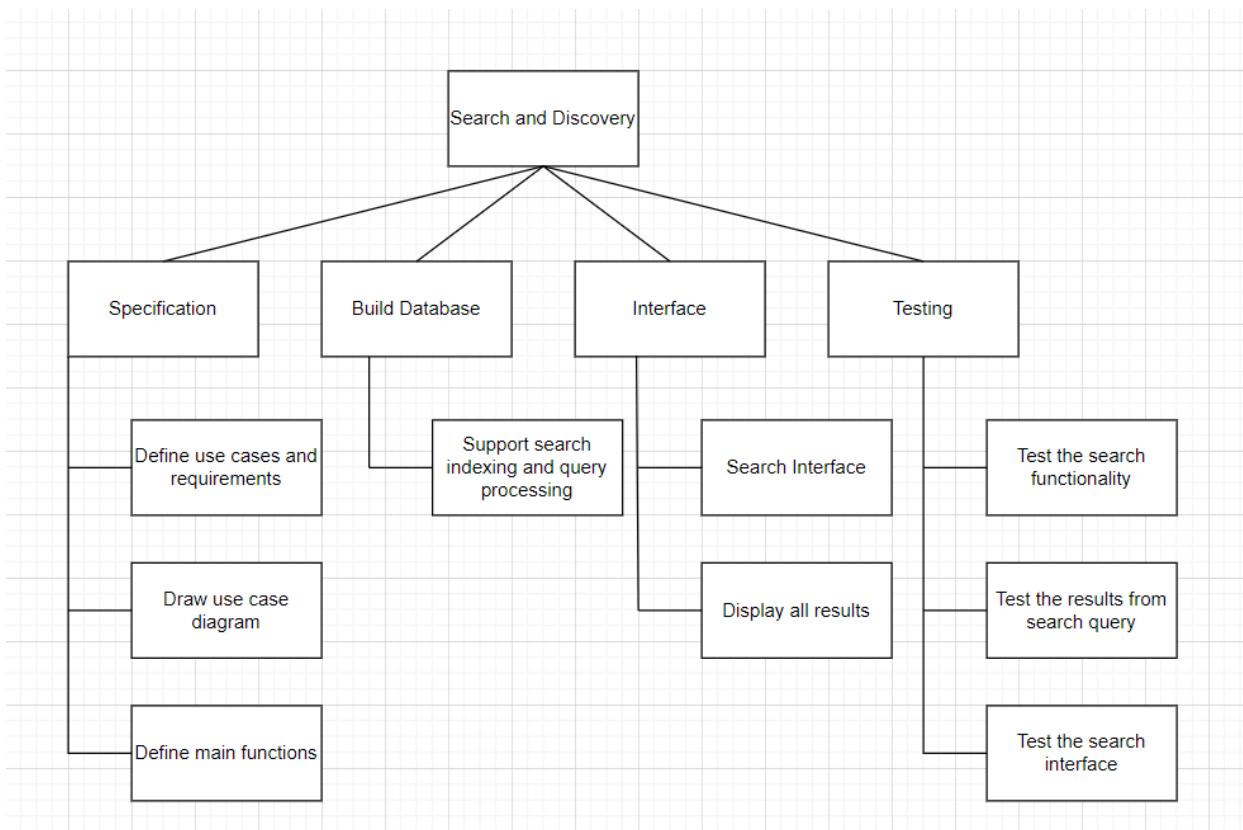
The task for the User Authentication is described in this figure below:



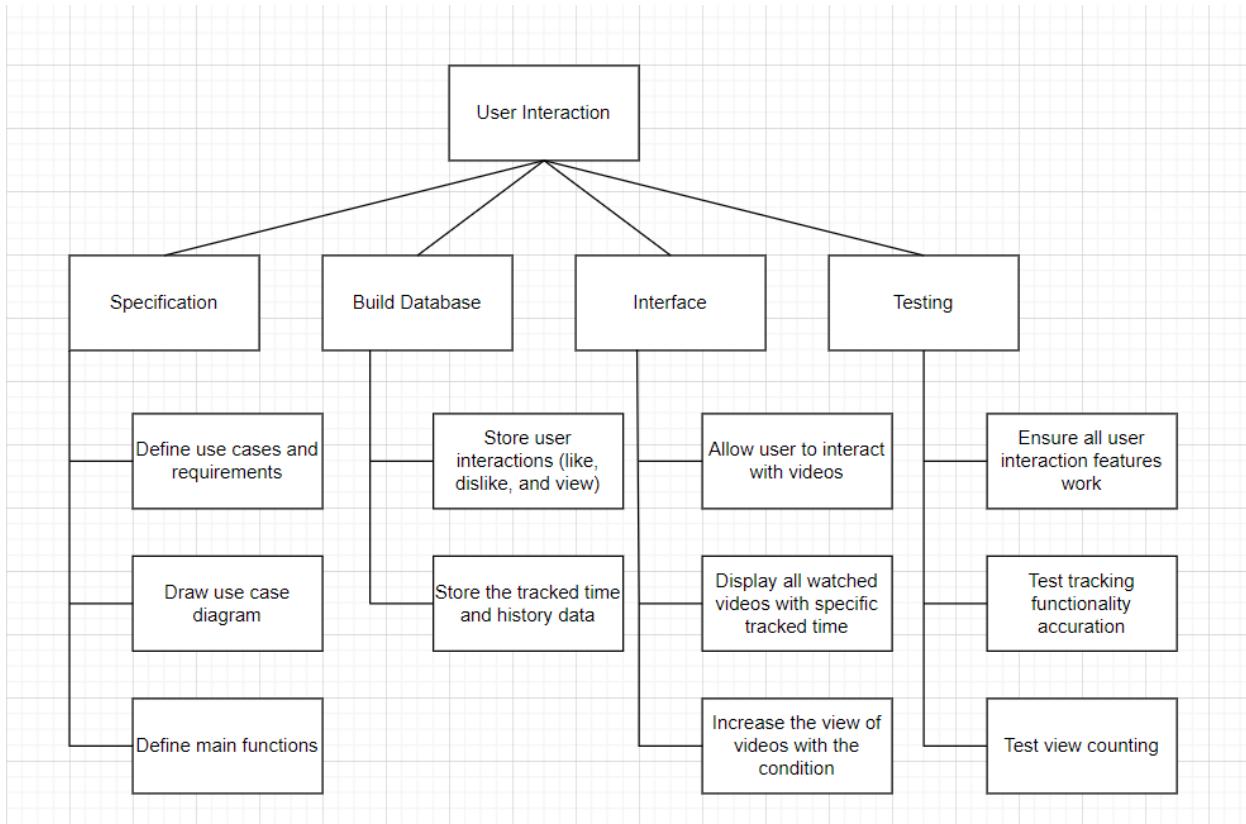
The task for the Video Management is described in this figure below:



The task for the Search and Discovery is described in this figure below:



The task for User Interaction is described in this figure below:



In our agile development approach, we anticipate the completion and meticulous testing of these tasks at the end of each iteration, as opposed to the traditional methodology that typically segments these steps.

4. DEVELOPMENT ENVIRONMENT:

The development process utilized various Programming Languages, structured around the Model-View-Controller (MVC) architecture.

- **Front End:** Utilizing **React** along with the **Ant Design** (antd) library for UI components.
- **Back End:** Implementing **Servlets** to handle server-side logic.
- **Database:** Employing **MySQL** for data storage and management.
- **Architecture:** Following the **Model-View-Controller** (MVC) for structuring code.
- **Version Control:** Managing code with **GitHub** to track changes and collaborate.
 - GitHub: <https://github.com/vinhngph/uTube.git>
- **Deployment:** Deploying our web application on **Amazon Web Services** (AWS).
- **Containerization:** Using **Docker** to create, deploy, and run our application by using containers, simplifying the development process.
- **Communication:** Using **Discord** for team meetings and discussions.

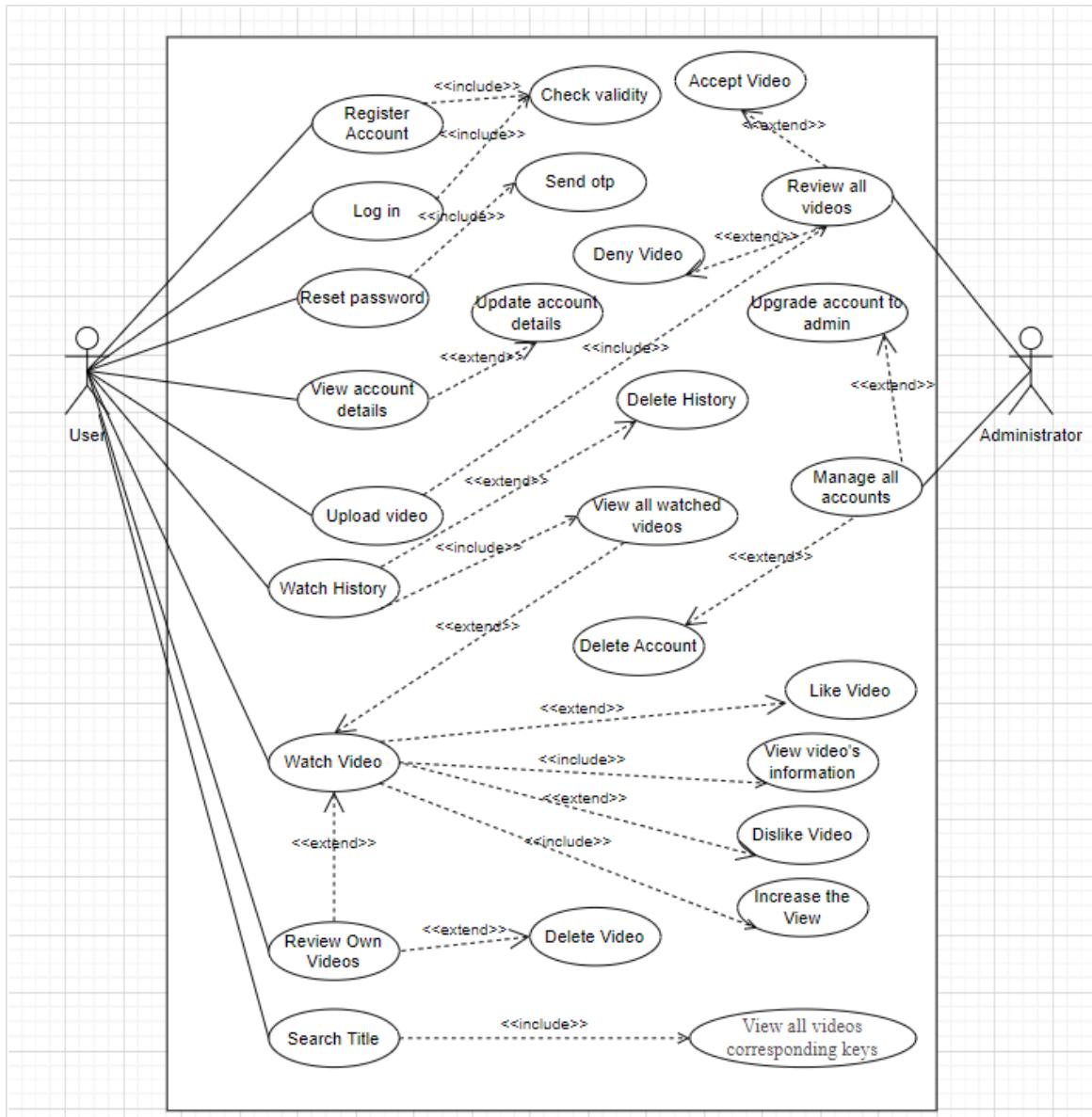
- **Task Management:** Keeping track of work with **Microsoft Excel** and **Trello**.
- **Documentation:** Storing all the API documentation in **Google Docs**.

II. REQUIREMENT ANALYSIS AND DESIGN:

This section outlines the analysis of requirements and the design framework, tailored for the upcoming development of a video platform. This part serves as a roadmap for the project's execution, detailing each feature along with the stipulated conditions, and both functional and non-functional requirements as provided by the clientele. Throughout the development phase, we will actively refine and enhance subsequent editions, ensuring that we maintain an accurate record of the project's advancement.

1. REQUIREMENT ANALYSIS:

USE CASE DIAGRAM:



Use case 1:

Name: Register Account

Identifier UC1

Inputs:

1. Email
2. Username
3. Password
4. Full name
5. Date of Birth

Outputs:

1. The login page (if successful).
2. The register page included with the message and border of the input field turn red (username or email already existed).

Basic Course:

Actor: User	System
1. Open the register page	1.1. Display the stage 1 of register page
2. Enter email, username, and password	2.1. Check email and username if it already existed.
3. Click 'Next', move to stage 2 of Register	3.1. Display the stage 2 of register page
4. Enter Full name and choose Date of Birth	
5. Submit	5.1. Show the message that account is successfully created 5.2. Display the login page

Precondition:

1. User does not have a registered account with the system.

Post-condition:

1. User obtains a unique ID and password for future logins.

User story: As a new visitor to the new video platform, I want to register an account so that I can enjoy features such as uploading videos, liking, and personalizing my viewing experience.

Use case 2:

Name: Login to the uTuBe

Identifier UC2

Inputs:

1. Username
2. Password

Outputs:

1. The home page (if successfully).
2. The login page (if failed).

Basic Course:

Actor: User	System
1. Open the login page	1.1. Display the stage 1 of register page
2. Enter username, and password	2.1. Check username and password if it is already valid.
3. Submit	5.1. Show the message that account is successfully logged 5.2. Display the home page

Precondition:

1. User has a registered account on the uTuBe platform with an existing username and password.

Post-condition:

1. User gains access to their account and can utilize all platform features.

User story: As a user of the uTuBe platform, I want to log into my account so that I can engage with the community by watching videos, liking content, and exploring things all around the world.

Use case 3:

Name: Reset password in page ‘Forgot Password’

Identifier UC3

Inputs:

1. Email
2. OTP
3. New password

Outputs:

1. The login page (if successful).
2. The forgot password page (if failed).

Basic Course:

Actor: User	System
1. Click the forgot password in Log in page	1.1. Display the interface of forgot password page with email input field
2. Enter email	2.1. Check the validity of email 2.2. Send OTP to email in mailbox
3. Enter OTP and new password	
4. Submit	4.1. Check OTP 4.2. Display the Log in page

Precondition:

1. Users cannot recall their account password for the uTuBe platform.

Post-condition:

1. Users can reset their password and regain access to their account.

User story: I want to be able to reset my forgotten password so that I can regain access to my account and continue to enjoy features such as watching videos and interacting with the community.

Use case 4:

Name: View account's detail

Identifier UC4

Inputs:

1. None

Outputs:

1. Display all information of account (if successful).
2. The error message (if failed).

Basic Course:

Actor: User	System
1. Click the avatar in the navbar	1.1. Drop down list
2. Click the Account Information	2.1. Display all the information of the user

Precondition:

1. Users are logged into their account on the uTuBe platform.

Post-condition:

2. Users can view and manage all their account information.

User story: As a registered member of the uTube platform, I want to be able to view all my account information so that I can manage my personal details.

Use case 5:

Name: Update account's information

Identifier UC5

Inputs:

1. New full name
2. New date of birth

Outputs:

1. The successful updated message (if successful).
3. The error message (if failed).

Basic Course:

Actor: User	System
1. In the Account Information page	1.1. Display all the information of the user
2. Click the Edit	2.1. The field 'full name' and 'date of birth' can editable
3. Enter new full name and/or new date of birth	
4. Submit	4.1. Show the successful updated message

Precondition:

1. Users are logged into their account on the uTuBe platform.

Post-condition:

2. User's account information is updated with the new changes.

User story: I want to be able to update my account information so that I can keep my personal details current, ensuring a personalized and up-to-date experience on the platform.

Use case 6:

Name: Upload Video into uTuBe

Identifier UC6

Inputs:

1. File video
2. File image
3. Title
4. Description

Outputs:

1. The successful message with 100% loading bar (if successful).
2. The error message (if failed).

Basic Course:

Actor: User	System
1. Open the Upload Video page	1.1. Display the Upload interface
2. Choose files video and image, and enter the title and description	
3. Upload	3.1. Loading bar running 3.2. Show the successful uploaded message

Precondition:

1. Users are logged into their account on the uTuBe platform.

Post-condition:

1. User successfully uploads a video to the platform

User story: As a content creator on the uTuBe platform, I want to upload videos so that I can share my content with viewers, engage with my audience, and grow my channel's presence.

Use case 7:

Name: Watch videos in uTuBe system

Identifier UC7

Inputs:

1. None.

Outputs:

1. A window with video and information of video.
2. The error message (if failed).

Basic Course:

Actor: User	System
1. Click the video card on the Home page, History page, Search page, or Manage Channel page.	1.1. Display the Video Playing page 1.2. Show the information of video
2. Watch video	2.1. The view of video is increase by 1

Precondition:

1. Users are browsing the uTube.

Post-condition:

1. Users watch a video of their choice.

User story: As a viewer on the uTuBe platform, I want to watch videos so that I can enjoy diverse content, learn new things, and be entertained by creators from around the world.

Use case 8:

Name: Interaction in video

Identifier UC8

Inputs:

1. Like/ Dislike the Video

Outputs:

1. The number of likes/ dislikes of the video has increased by 1.

Basic Course:

Actor: User	System
1. In Video Playing page	
2. Click Like/ Dislike the video	2.1. The number of likes/ dislikes of the video has increased by 1.

Precondition:

1. Users are watching a video on the uTuBe platform.

Post-condition:

1. Users express their opinion by liking or disliking the video.

User story: As an engaged viewer on the uTuBe, I want to like or dislike videos so that I can provide feedback to creators and help tailor content recommendations to my preferences.

Use case 9:

Name: Watch history

Identifier UC9

Inputs:

1. None.

Outputs:

1. Display all the watched videos.

Basic Course:

Actor: User	System
1. Open the History page	1.1. Display all the watched videos.
2. Click one of the watched videos	2.1. Play the video that seek at the time user exits it previously.

Precondition:

1. Users have watched videos on the uTuBe platform.

Post-condition:

1. Users can review their watch history.

User story: As a user of the uTuBe platform, I want to access my watch history so that I can keep track of the videos I've viewed and easily revisit any content I found interesting or informative.

Use case 10:

Name: Manage channel

Identifier UC10

Inputs:

1. None.

Outputs:

2. Display all the user's own videos.

Basic Course:

Actor: User	System
1. Open the Manage Channel page	1.1. Display all the uploaded videos.
2. Click one of the uploaded videos	2.1. Display the Video Playing page of that video

Precondition:

1. Users have uploaded videos on the uTuBe platform.

Post-condition:

1. Users can review and manage their uploaded videos.

User story: As a content creator on the uTuBe platform, I want to review my uploaded videos so that I can monitor engagement, analyze viewer feedback, and optimize my content strategy.

Use case 11:

Name: Search title

Identifier UC11

Inputs:

1. Title

Outputs:

1. Drop down the recommended titles
2. Display all videos matching with key search (if matched)
3. Display a message that no search be found (if null or not matched)

Basic Course:

Actor: User	System
1. Click on the search bar in navbar	
2. Enter the search key	2.1. Drop down the list of title that matched with the key
3. Enter or click the title in the list	3.1. Display all videos matched with key search

Precondition:

1. User is browsing the uTuBe platform.

Post-condition:

1. User finds a video by searching for its title.

User story: As a viewer on the uTuBe platform, I want to search for videos by title so that I can quickly find specific content I'm interested in watching.

Use case 12:

Name: Manage All Accounts

Identifier UC12

Inputs:

1. None

Outputs:

1. Display all accounts of users.

Basic Course:

Actor: Administrator	System
1. Open the Manage Account page	1.1. Display all accounts of users
2. Modify the account (Upgrade or Delete)	2.1. If 'Upgrade' then account from user turn to admin 2.2. If 'Delete' then account of user is deleted

Precondition:

1. Admin is logged into the uTuBe platform.

Post-condition:

2. User finds a video by searching for its title.

User Story: As an administrator of the uTuBe platform, I want to manage user accounts so that I can ensure a safe and enjoyable experience for all users by overseeing account activity.

Use case 13:

Name: Censor videos

Identifier UC13

Inputs:

1. None.

Outputs:

1. Display all uploaded but not published into video platform.

Basic Course:

Actor: Administrator	System
1. Open the Censor page	1.1. Display title Admin Page and list all videos are waiting
2. Accept or deny the video	2.1. If 'Accept', then video disappeared from the list and published into Home page 2.2. if 'Deny', then video deleted

Precondition:

1. Admin is logged into the uTuBe platform.

Post-condition:

2. Admin can review and censor videos as per content.

User story: As an administrator of the uTuBe, I want to be able to censor videos so that I can remove inappropriate content and maintain a positive and safe environment for all viewers.

A. FUNCTIONAL REQUIREMENTS:

Use case 1: Register Account

1. Product Scope:

- Integral to the account registration functionality for new users

2. Requirements:

- a. Functional:

- Present the registration form for new users to enter their details.
 - Store new user account information in the database.
 - Confirm the validity and uniqueness of the new user's registration details.
- b. Data:
- Registration details (username, email, full name, date of birth, and password) must be valid.

Use case 2: Login to the uTuBe

1. Product Scope:

- Critical component of the user login feature.

2. Requirements:

- a. Functional:

- Display login interface for user credential input.
- Retrieve matching account information from the database using the username.
- Validate the legitimacy of the user's login credentials

- b. Data:

- Login credentials (username and password) must be authentic and valid.

Use case 3: Reset password in page 'Forgot Password'

1. Product Scope:

- The ‘Forgot Password’ feature is a critical security function that allows users to reset their passwords if they cannot remember them, ensuring they can regain access to their accounts without compromising account security.

2. Requirements:

- a. Functional:
 - Prompt the user to enter their registered email to receive an OTP.
 - Validate the entered OTP to verify the user’s identity.
 - Allow the user to set a new password upon successful OTP verification.
 - Redirect the user to the login page after a successful password reset.
 - Return the user to the ‘Forgot Password’ page with an error message if any step fails.
- b. Data:
 - The email must be associated with an existing user account.
 - The OTP must be correct and valid within the stipulated time frame.
 - The new password must comply with the platform’s password policy.

Use case 4: View Account Detail

1. Product Scope:

- Essential aspect of user profile management, providing users with the ability to view and verify their personal information stored on the platform.

2. Requirements:

- a. Functional:
 - Provide a user interface element (such as an avatar) in the navigation bar for user interaction.
 - Include an ‘Account Information’ option within the dropdown list.
 - Show all user account information.
 - Present an error message if the account details cannot be displayed.
 - Retrieve matching account information from the database using the username.
 - Validate the legitimacy of the user’s login credentials
- b. Data:
 - All displayed account information must be accurate and up to date.

Use case 5: Update Account Information

1. Product Scope:

- Vital function that allows users to maintain the accuracy of their personal details on the platform, ensuring their profile remains current and relevant.

2. Requirements:
 - a. Functional:
 - Display the user's current account information on the 'Account Information' page.
 - Provide an 'Edit' option that enables the 'full name' and 'date of birth' fields to become editable.
 - Accept new input for the full name and/or date of birth from the user.
 - b. Data:
 - The new full name and date of birth must be valid and adhere to any format requirements specified by the platform.

Use case 6: Upload Video

1. Product Scope:
 - Core functionality of the uTuBe platform, enabling users to share their content with the community and contribute to the diversity of available videos.
2. Requirements:
 - a. Functional:
 - Provide an 'Upload Video' page with an intuitive user interface for uploading content.
 - Allow users to select video and image files for thumbnail and input a title and description for their content.
 - Initiate the upload process when the user confirms their selection and inputs.
 - Display a loading bar indicating the progress of the upload.
 - b. Data:
 - The video file must be in a supported format and within size limits specified by the platform.
 - The image file, typically used as a thumbnail, must also meet format and size criteria.
 - The title and description should comply with community guidelines and not contain prohibited content.

Use case 7: Watch Videos

1. Product Scope:
 - Primary service of the uTuBe platform, providing users with the ability to view a wide range of video content uploaded by other users.
2. Requirements:
 - a. Functional:

- Display video cards on various pages such as Home, History, Search, or Manage Channel.
- Open the ‘Video Playing’ page when a video card is clicked.
- Show all relevant information about the video on the ‘Video Playing’ page.
- Play the video for the user to watch
- Increment the view count of the video by 1 each time it is watched.
- b. Data:
- None.

Use case 8: Interaction with video

1. Product Scope:
 - The ‘Like/Dislike’ feature is an interactive function that allows users to express their opinions on videos, contributing to the community feedback and engagement on the uTuBe platform.
2. Requirements:
 - a. Functional:
 - Provide ‘Like’ and ‘Dislike’ buttons on the ‘Video Playing’ page.
 - Register the user’s action when they click on either the ‘Like’ or ‘Dislike’ button.
 - Update the like/dislike count of the video accordingly by incrementing it by 1.
 - b. Data:
 - Like or Dislike action of user.

Use case 9: Watch History

1. Product Scope:
 - The ‘Watch history’ feature allows users to revisit their previously watched videos, enhancing user experience by providing easy access to their viewing history on the uTuBe platform.
2. Requirements:
 - a. Functional:
 - Display a ‘History’ page accessible to the user.
 - Show all videos that the user has watched on the ‘History’ page.
 - Allow users to click on a watched video to play it from where they left off.
 - b. Data:
 - The track time must be accurate.

Use case 10: Manage Channel

1. Product Scope:
 - The ‘Manage channel’ feature provides content creators with the ability to oversee their uploaded videos, facilitating better content management and user engagement on the uTuBe platform.
2. Requirements:
 - a. Functional:
 - Display a ‘Manage Channel’ page accessible to the content creator.
 - Show all videos that the content creator has uploaded on the ‘Manage Channel’ page.
 - Allow content creators to click on an uploaded video to view its ‘Video Playing’ page.
 - b. Data:
 - None.

Use case 11: Search Title

1. Product Scope:
 - The ‘Search title’ feature enables users to quickly find videos by entering keywords, enhancing discoverability and user experience on the uTuBe platform.
2. Requirements:
 - a. Functional:
 - Provide a search bar in the navigation bar.
 - Offer a dropdown list of recommended titles that match the entered search key.
 - Display all videos that match the search key if any are found.
 - Show a message indicating no search results found if there are no matches.
 - b. Data:
 - Search query key.

Use case 12: Manage All Account

1. Product Scope:
 - The ‘Manage All Accounts’ feature provides administrators with the tools to oversee user accounts, ensuring proper account management and security on the uTuBe platform.

2. Requirements:
 - a. Functional:
 - Display a ‘Manage Account’ page accessible to administrators.
 - Show all user accounts on the ‘Manage Account’ page.
 - Allow administrators to modify accounts by upgrading them to admin status or deleting them.
 - b. Data:
 - None.

Use case 13: Censor videos

1. Product Scope:
 - The ‘Censor videos’ feature allows administrators to review and decide whether uploaded videos meet the platform’s content guidelines, ensuring the quality and appropriateness of videos on the uTuBe platform.
3. Requirements:
 - a. Functional:
 - Display a ‘Censor’ page accessible to administrators.
 - Show all uploaded but not yet published videos on the ‘Censor’ page.
 - Allow administrators to accept or deny videos for publication.
 - b. Data:
 - None.

B. NON-FUNCTIONAL REQUIREMENTS

- 1. Usability requirements:**
 - It is essential to ensure that all software components function correctly and efficiently.
 - The hosting environment must maintain a valid license to effectively manage all incoming requests.
 - An administrator should be appointed to regularly update the software and monitor for any system errors.

- 2. Usability requirements:**
 - Well-organized and easy to follow.
 - Clear, and formative.
 - Intuitive, facilitating easy learning and navigation
 - Effortlessly and without errors when achieving objectives.

3. Humanity Requirements:

- Comprehensive documentation is produced that enables users to understand the system thoroughly by reading it.
- Intuitively designed GUI for ease of learning and use, allowing users to become familiar with the system quickly.
- Responsiveness of the interfaces are essential components of the requirements, contributing to an enhanced user experience.

4. Maintainability Requirements:

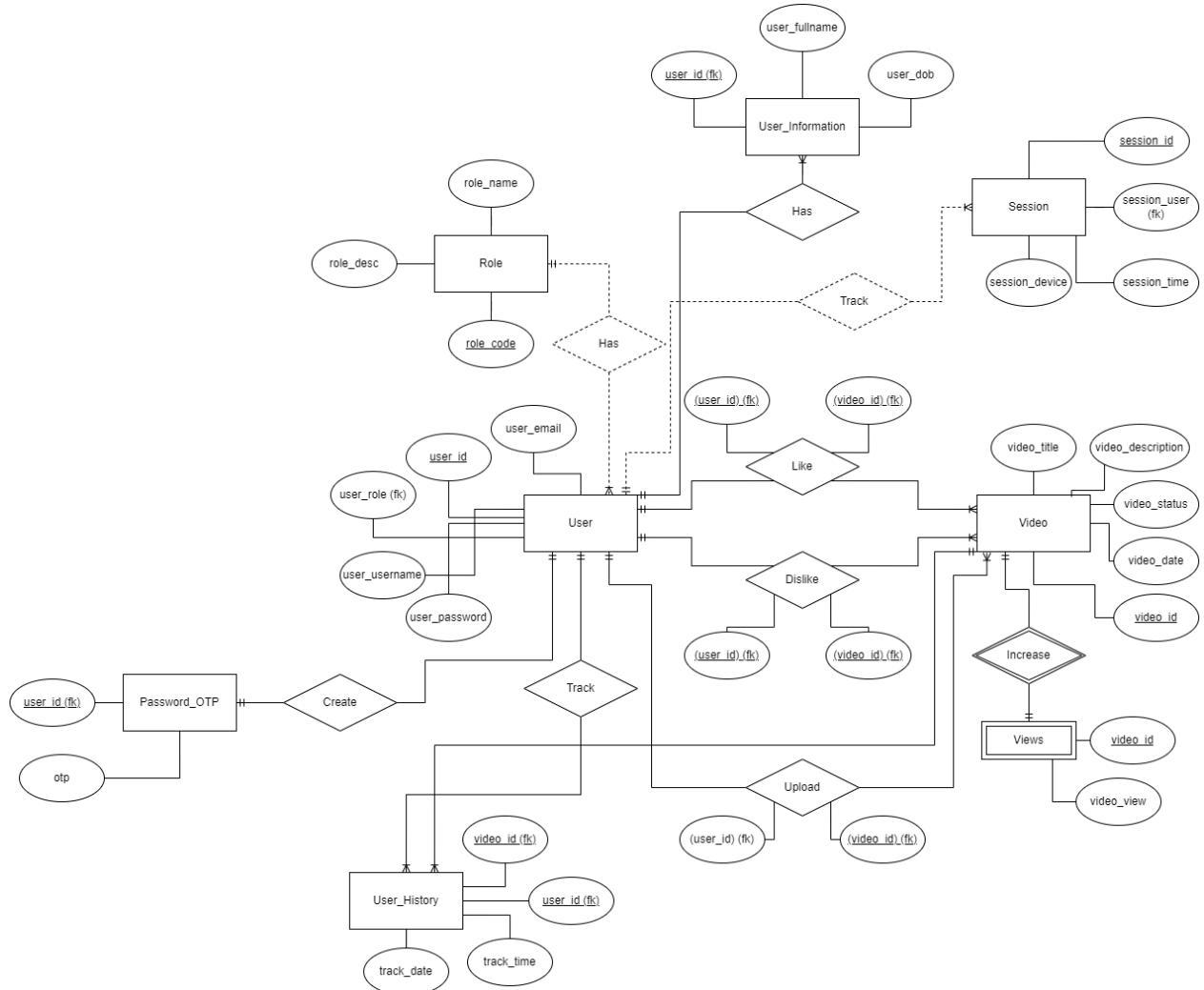
- The system is designed for ongoing updates and modifications. Given that users may have limited insight into the system's inner workings, it is imperative that the development team constructs the platform to be user-friendly and straightforward for both end-users and subsequent maintenance personnel.
- The platform should include intuitive functions that allow users to update and manage information related to videos, channels, and viewer interactions with ease.
- The source code must be well-structured and meticulously commented on to enhance comprehension and facilitate future updates or troubleshooting.
- A clearly defined and stable database architecture is crucial, providing a solid foundation for maintenance tasks and scalability.

5. Interface Requirements:

- Utilize Ant Design's (antd) comprehensive library of UI components to create intuitive buttons and controls that visually communicate their functionality.
- Leverage React's flexibility alongside antd's theming capabilities to create a visually appealing and customizable user interface.
- Incorporate language options within the interface to accommodate both native and international users, enhancing accessibility.

2. DESIGN:

A. ERD Diagram:



B. Relational Schema Diagram:

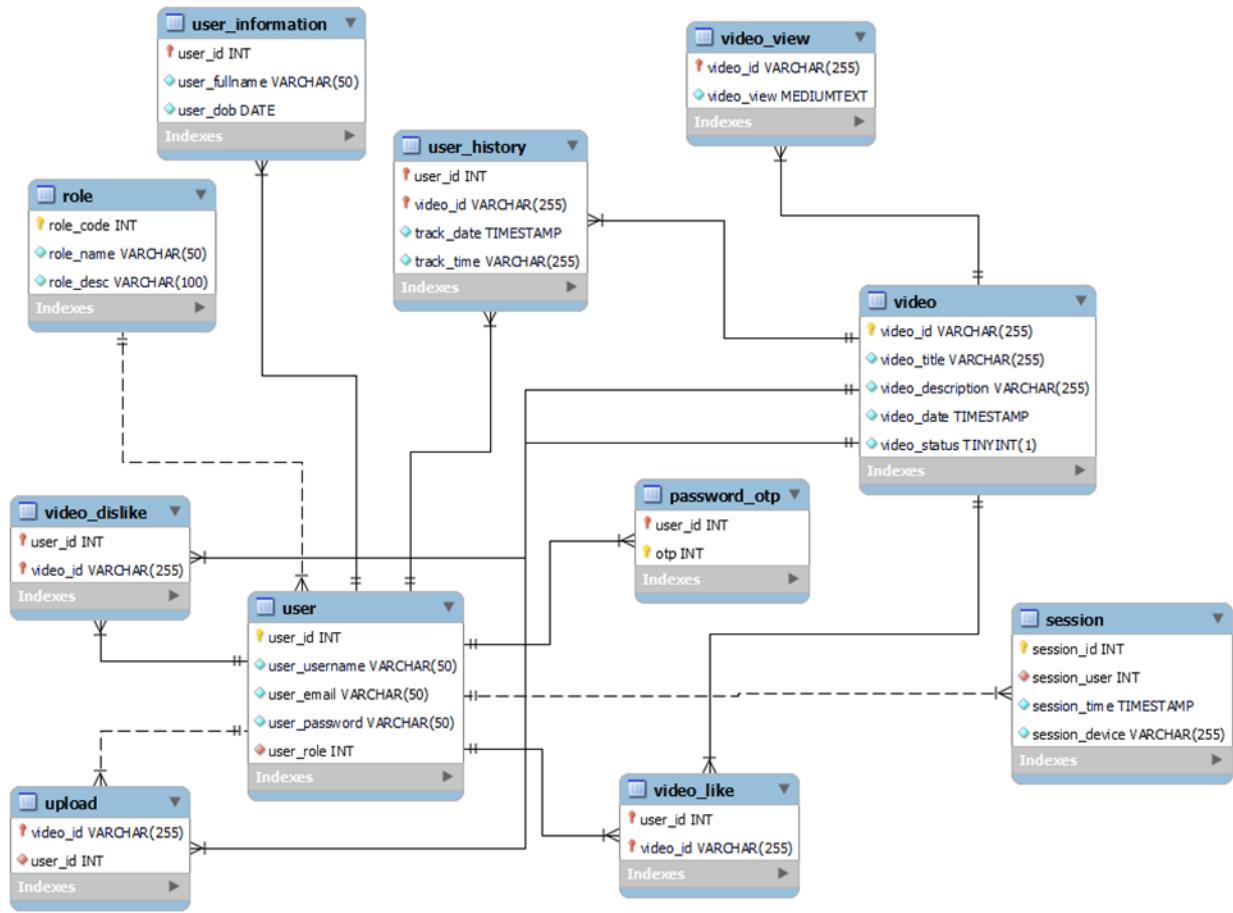
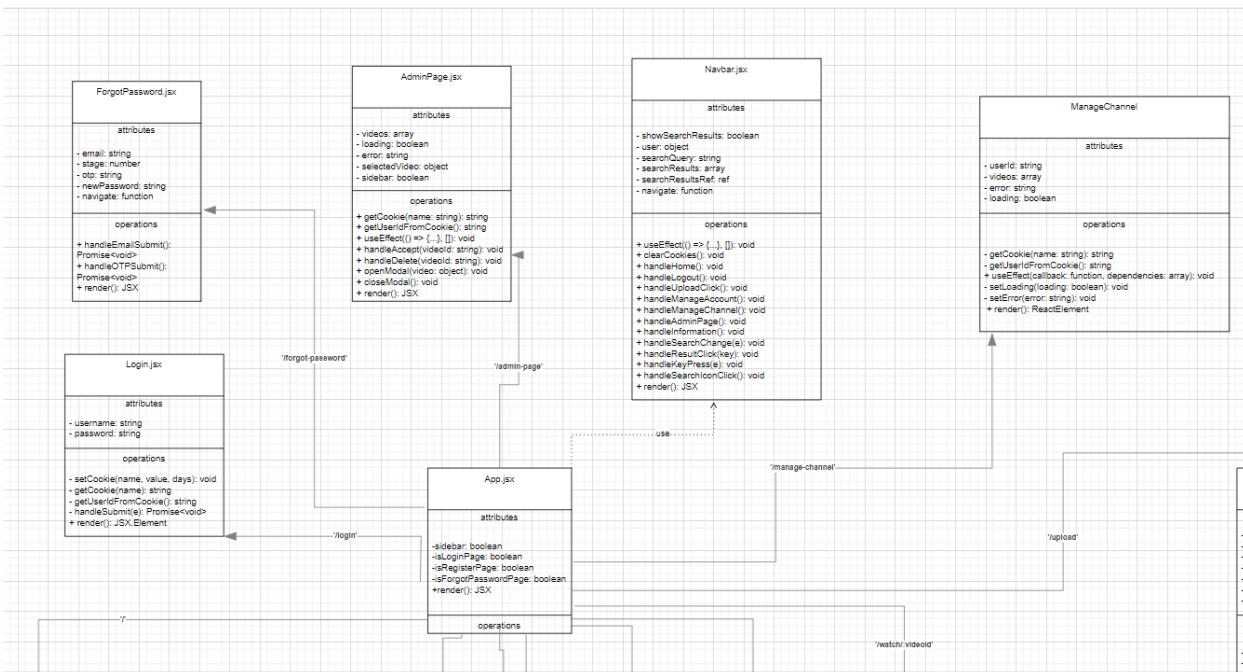
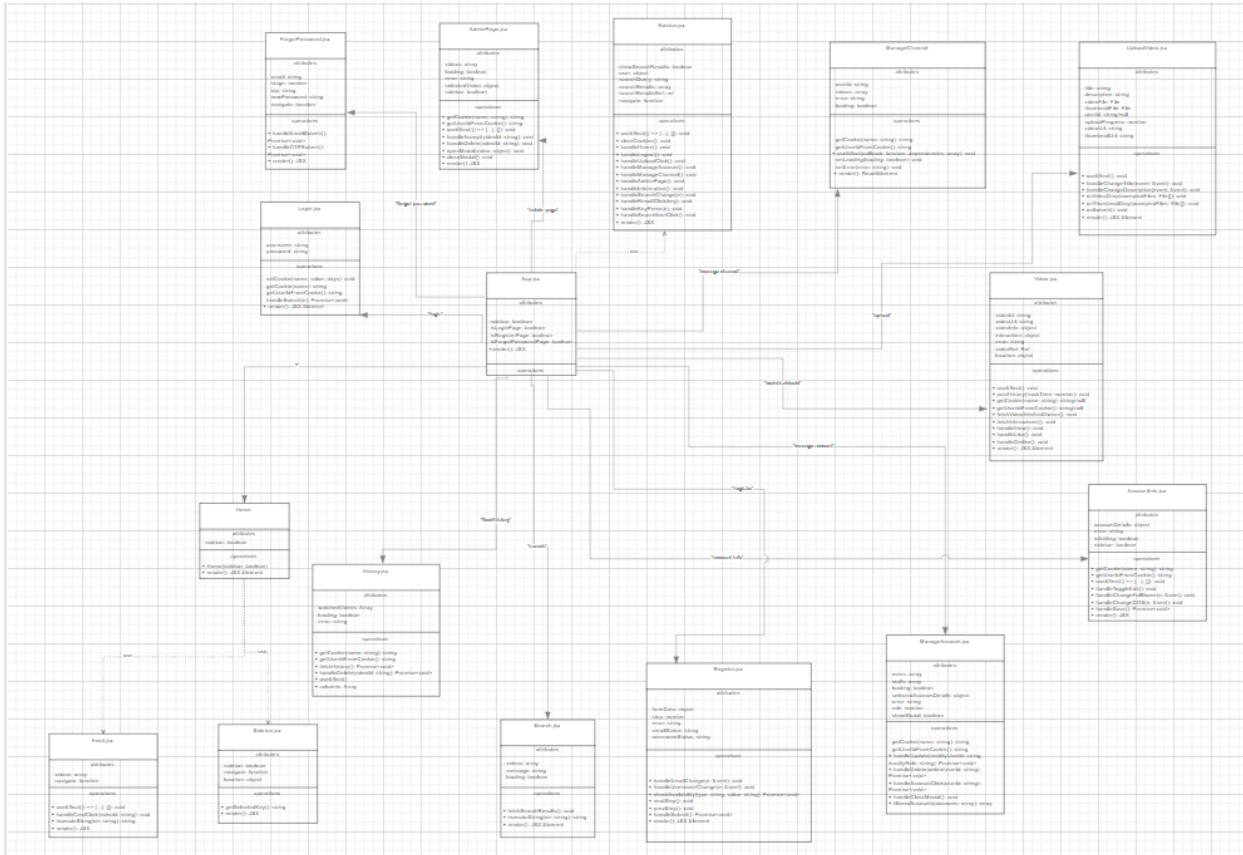
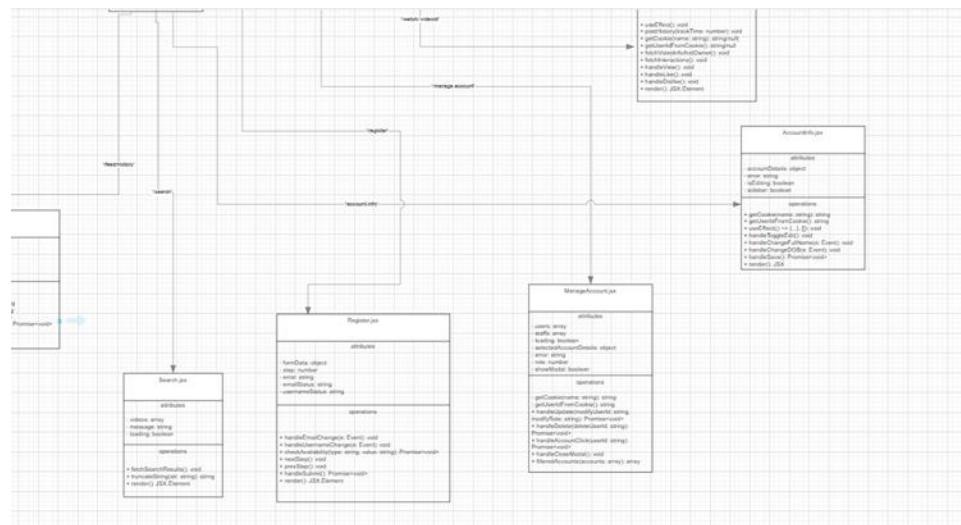
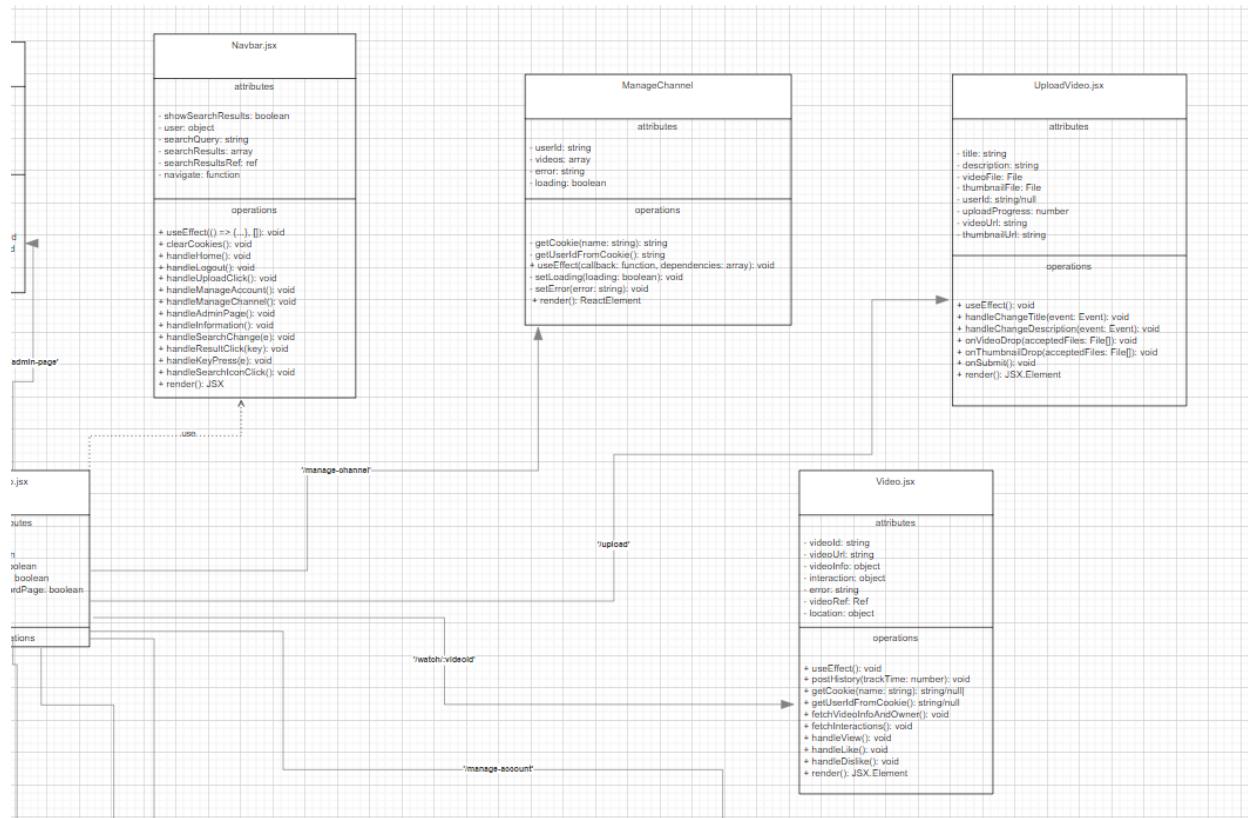


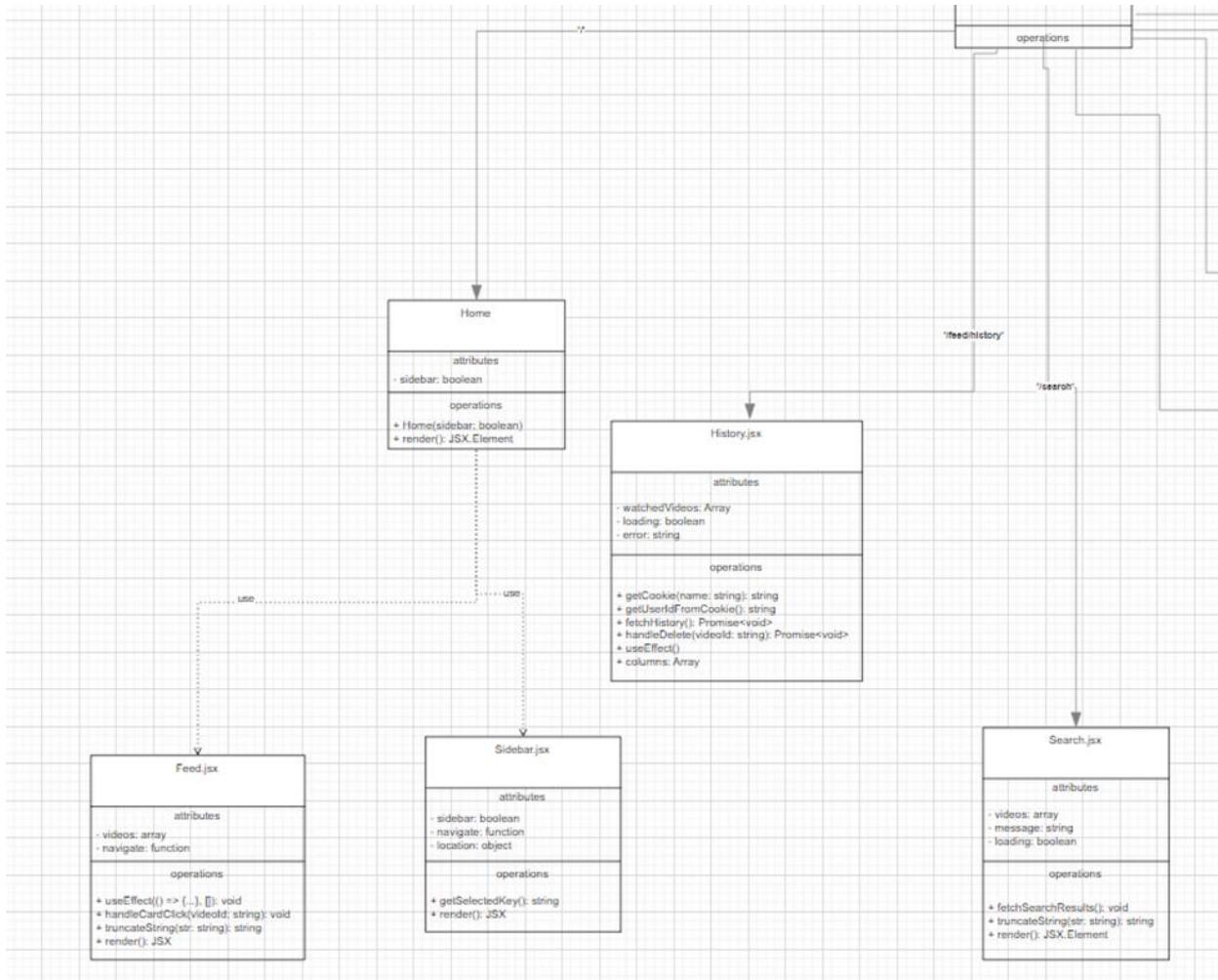
Figure 2.B

C. Class-Diagram:

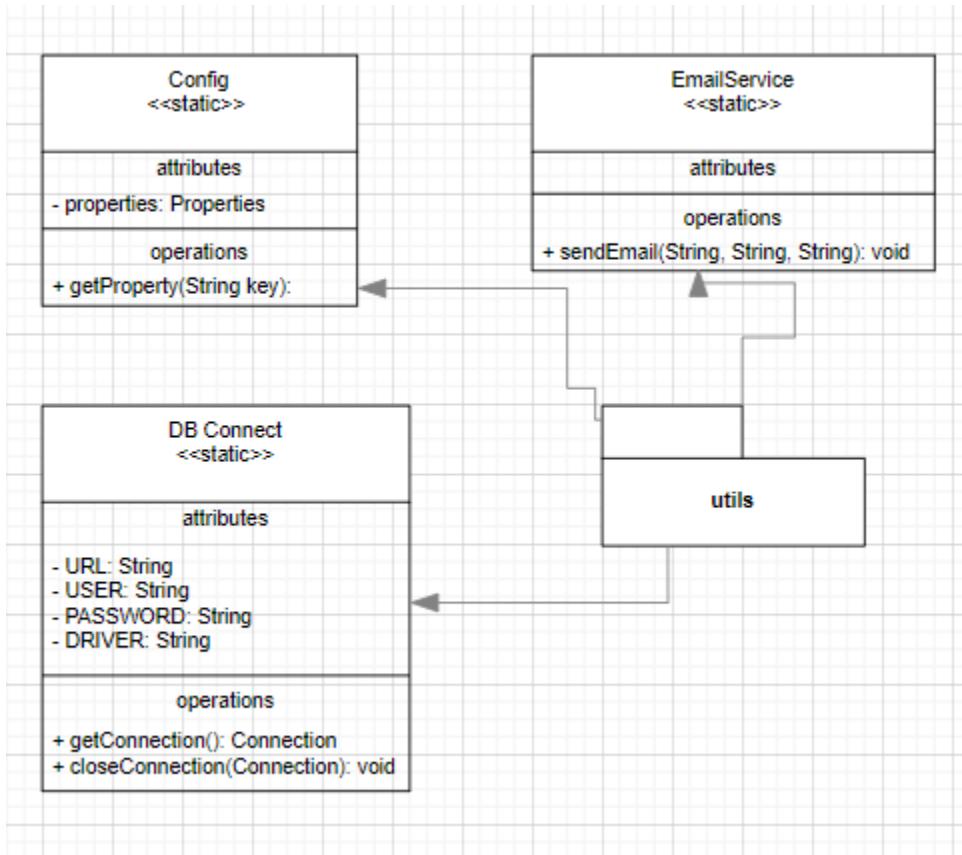
a. Front-End:



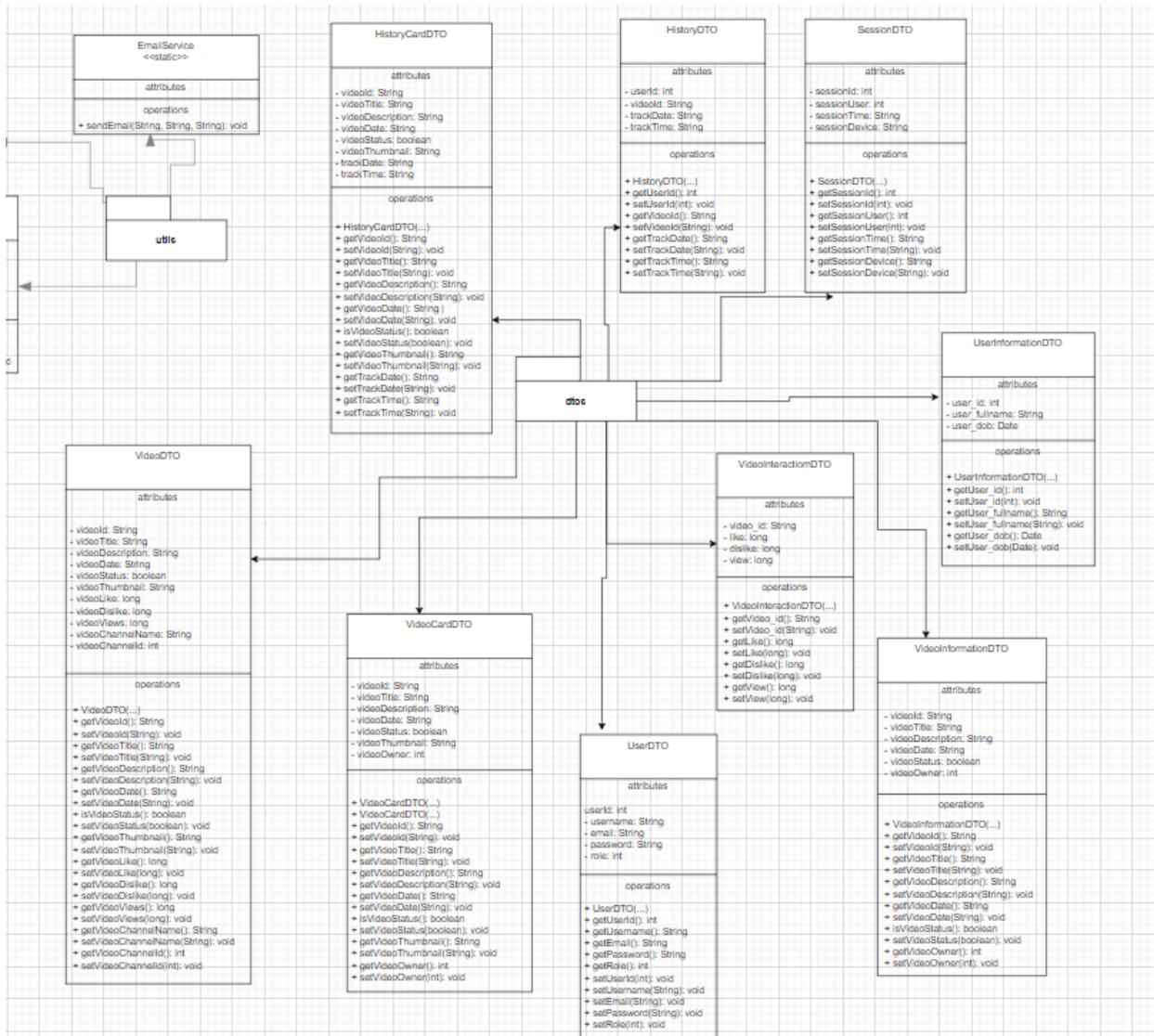




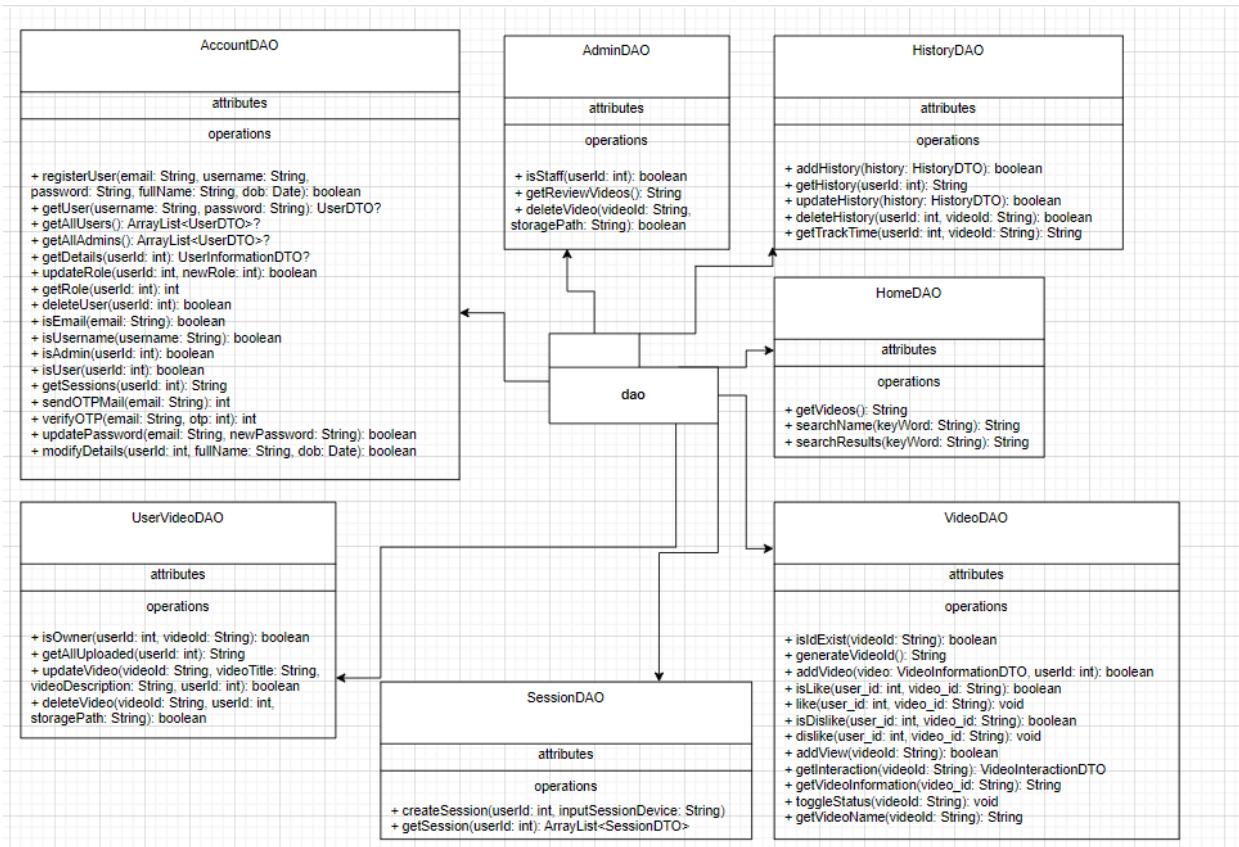
b. Back-End:



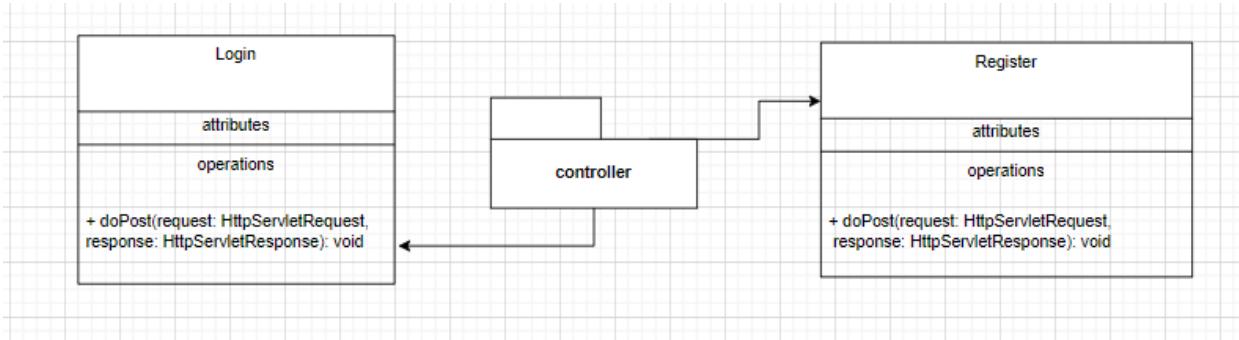
utils



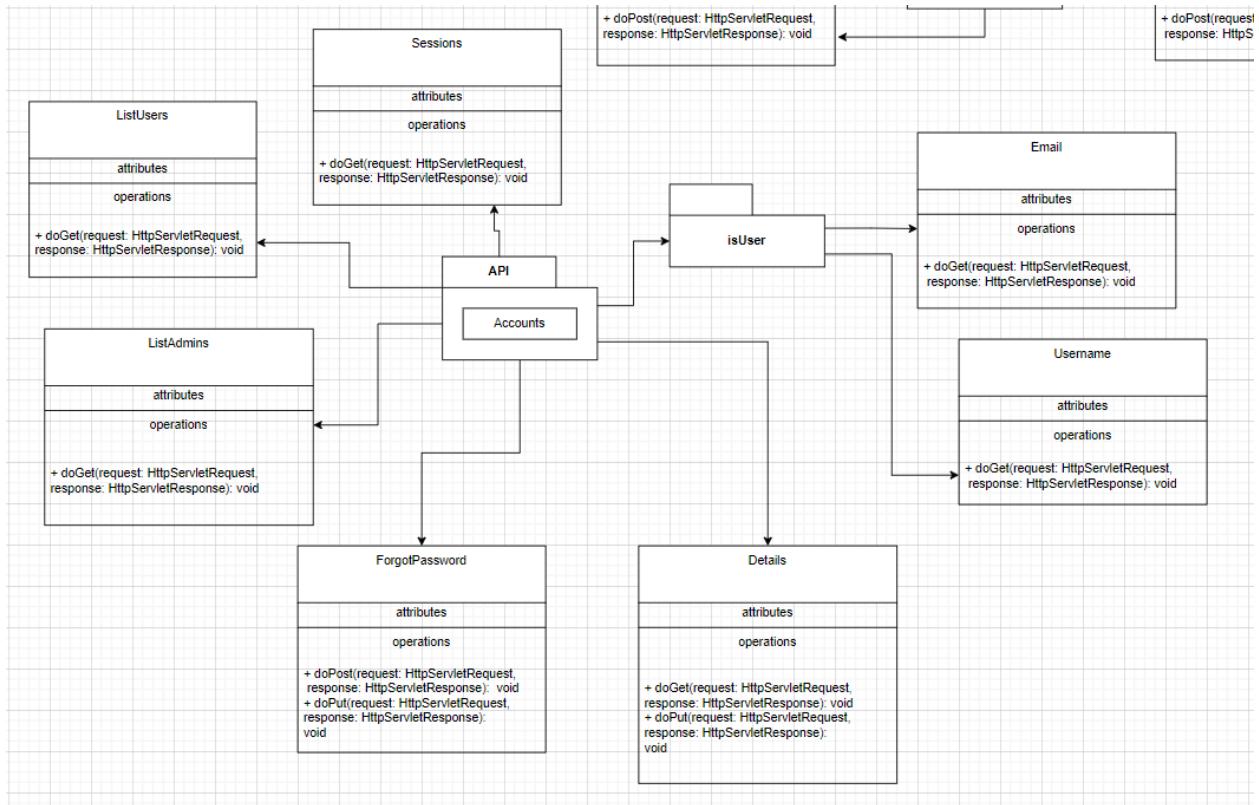
dtos



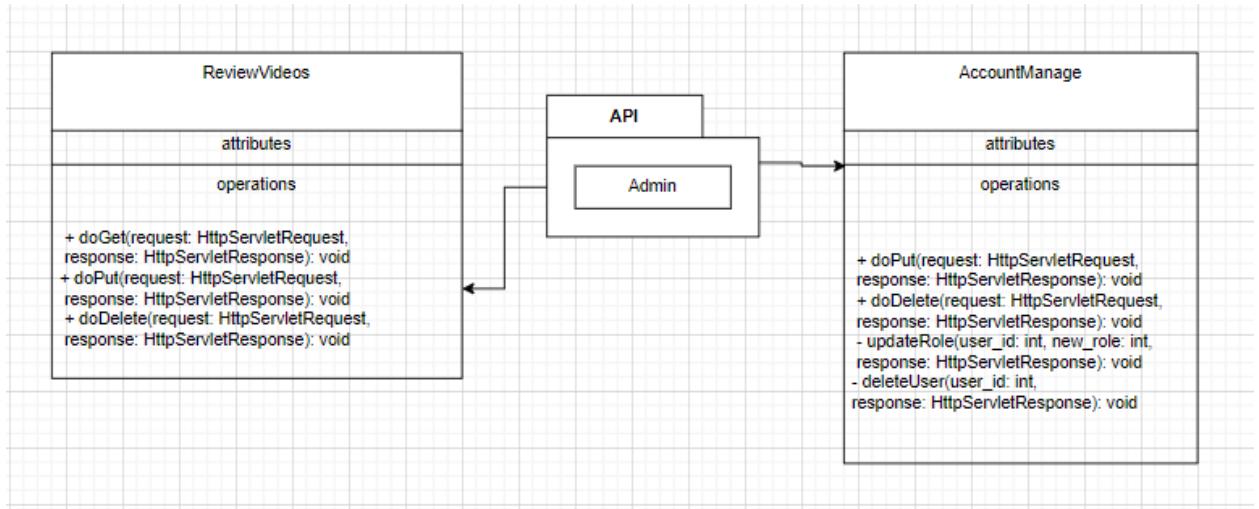
dao



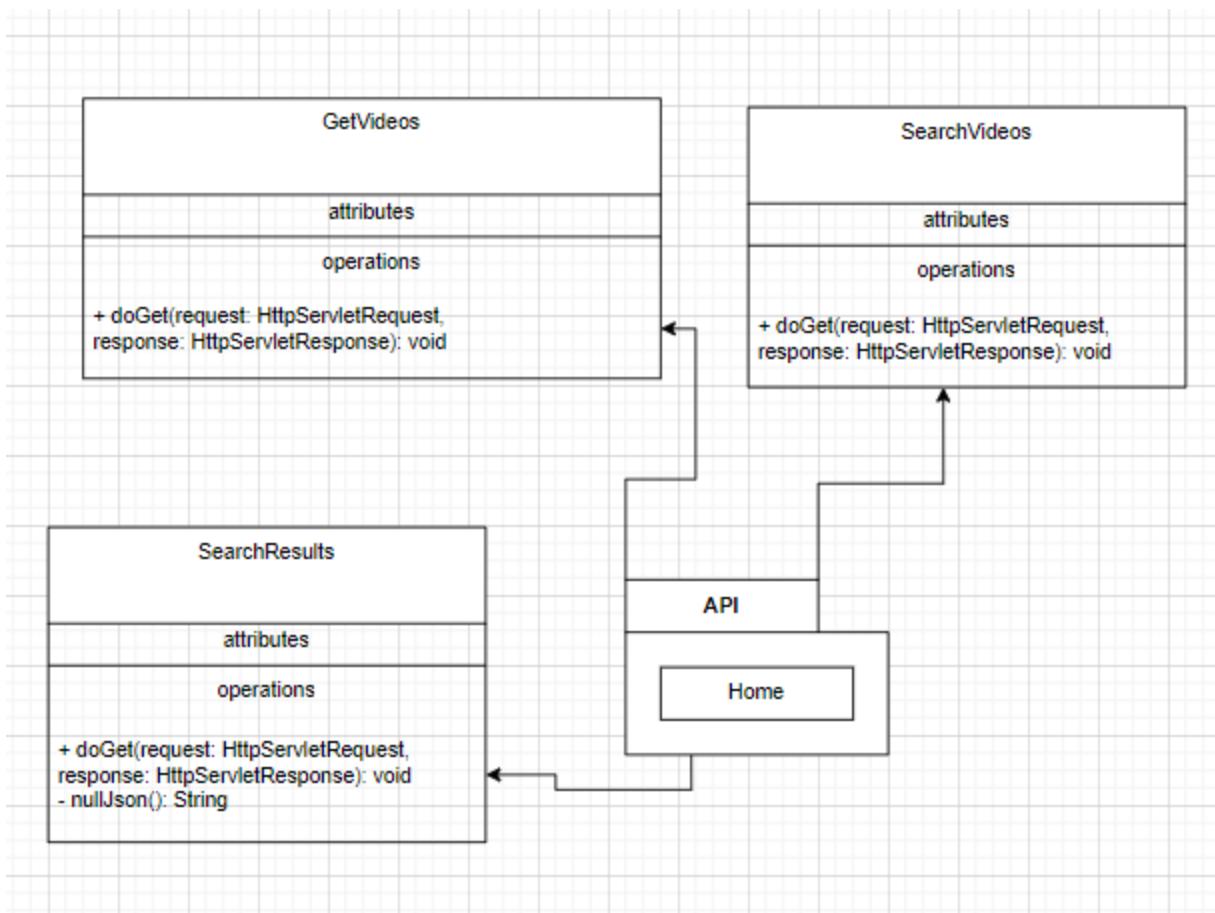
controller



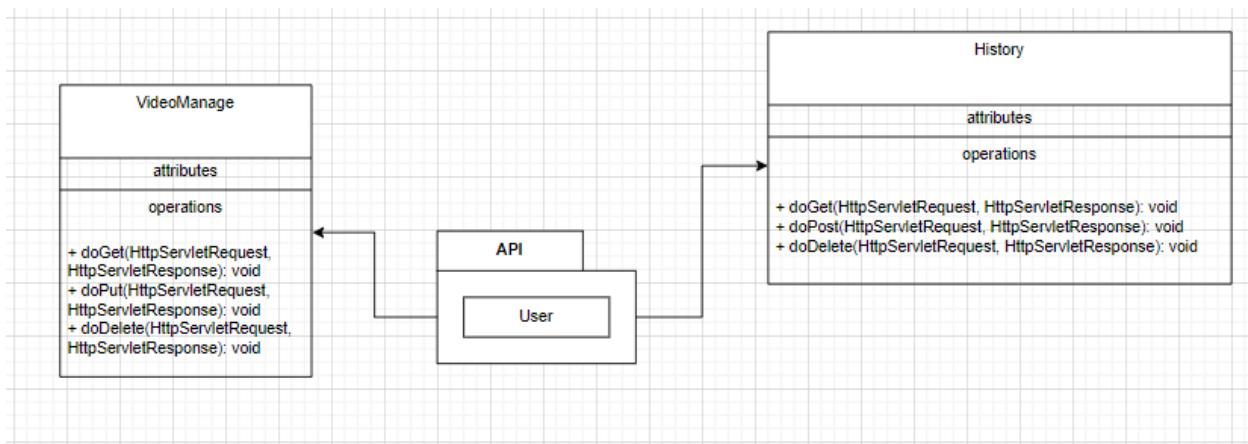
API-Accounts



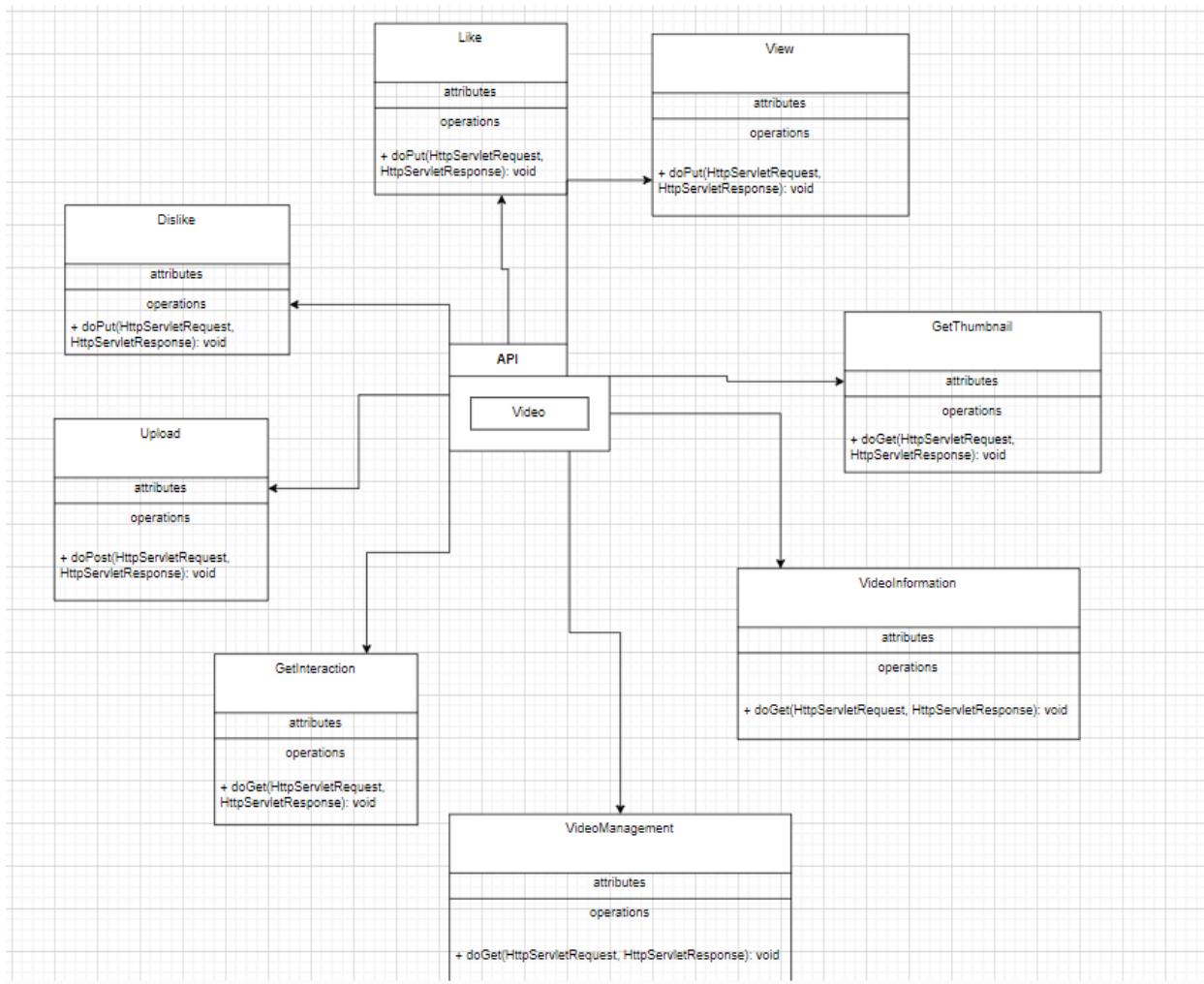
API-Admin



API-Home

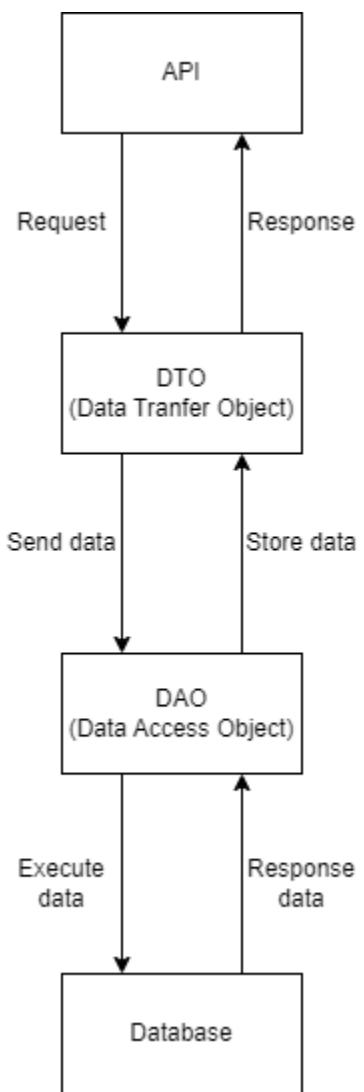


API-User

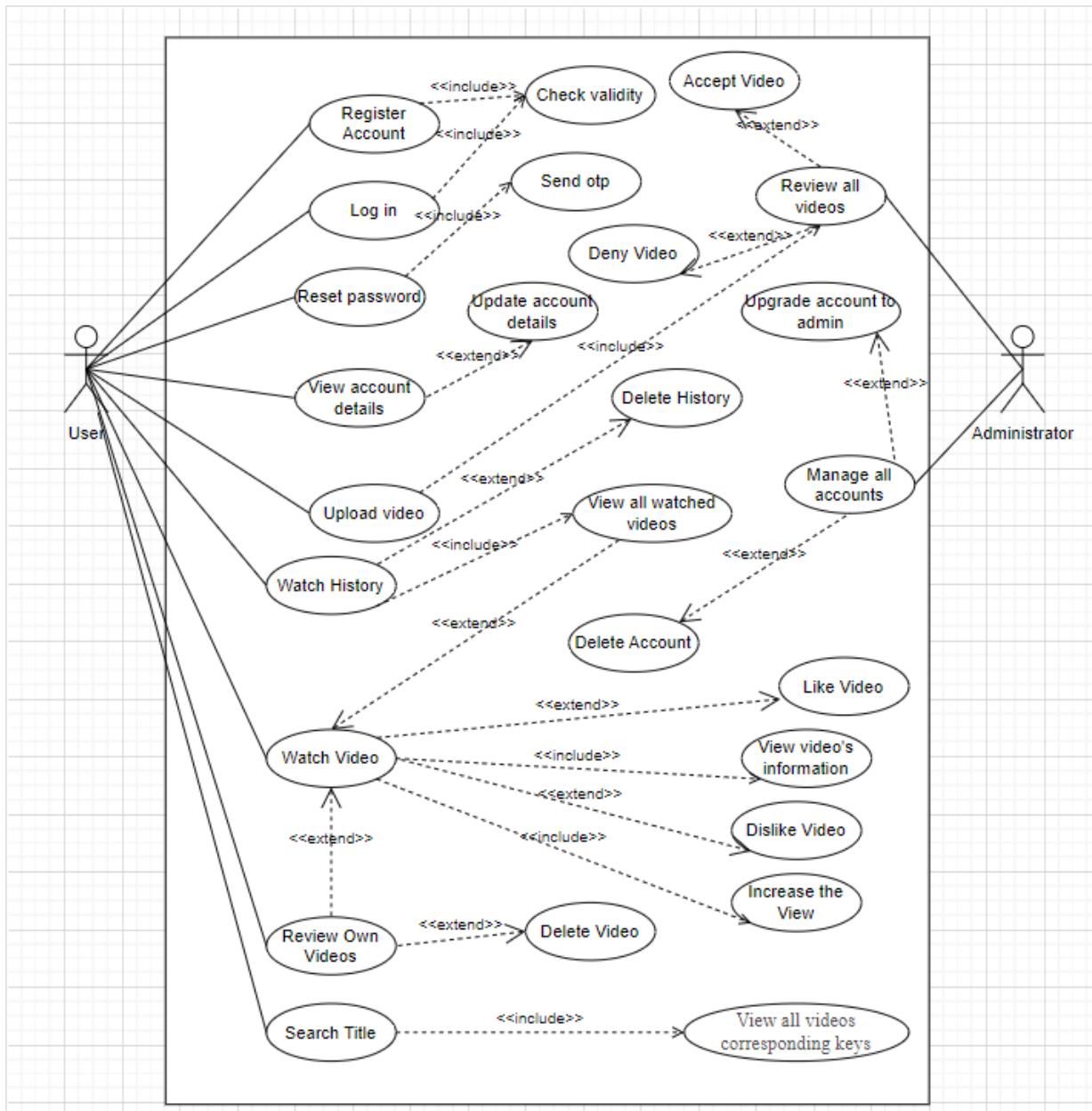


API-Video

c. *Logic flow of backend API:*



D. User Case Diagram:



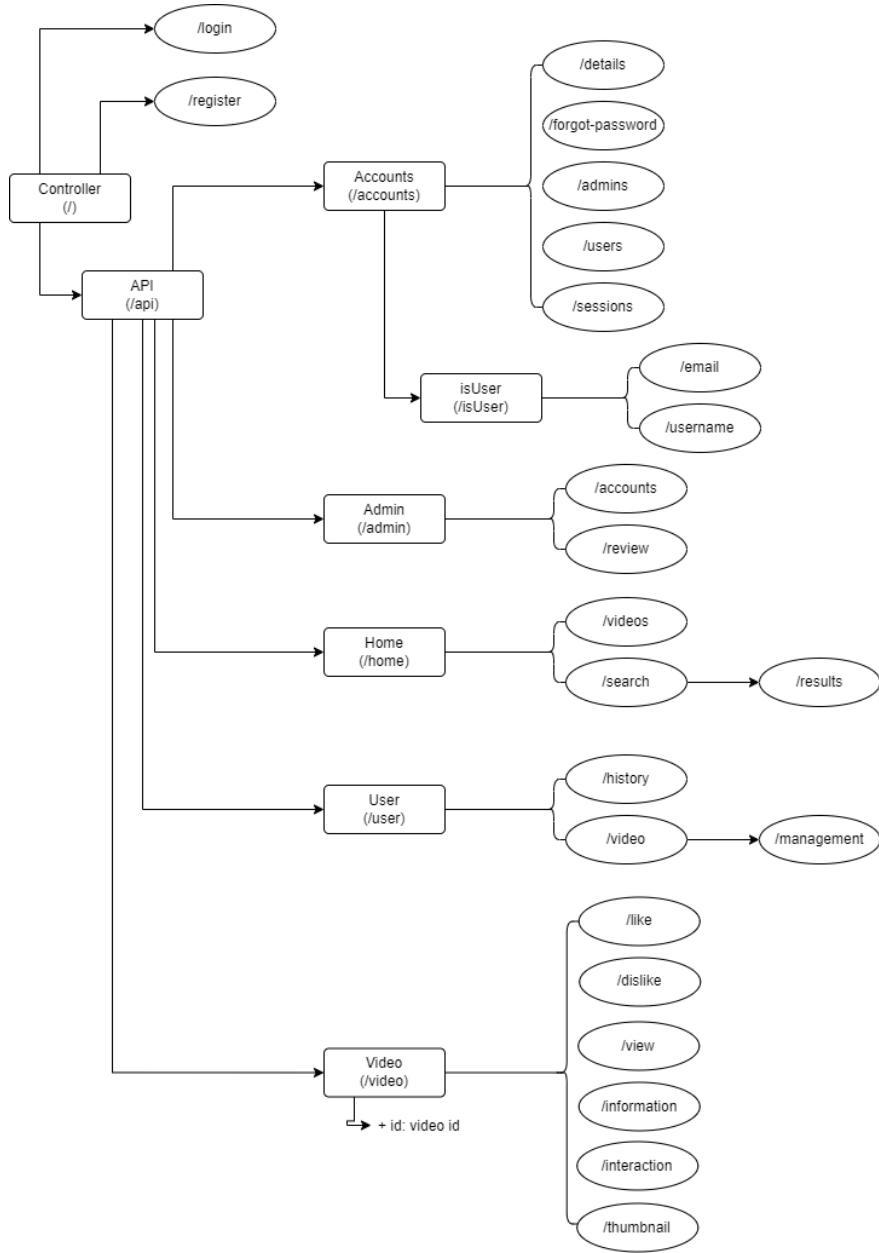
E. API:

a. Front End:

- **'/login'**: Login page for users to enter credentials.
- **'/register'**: Registration page for creating a new account.
- **'/forgot-password'**: Page to reset the password via email.
- **'/'**: Homepage showing video content.

- ‘/upload’: Page to upload new videos.
- ‘/account-info’: Page displaying user account information.
- ‘/admin-page’: Admin page for censoring videos.
- ‘/manage-account’: Admin page for managing user accounts.
- ‘/manage-channel’: Page for users to manage their uploaded videos.
- ‘/watch/:videoid’: Page to watch a specific video.
- ‘/search’: Search page for finding video.
- ‘feed/history’: Page displaying user's watch history.

b. Back End:



Controller (/):

- **/login:** Check if the username and password are correct.
- **/register:** Register a new account and add it to the database.

API (/api):

- **Account (/accounts):**

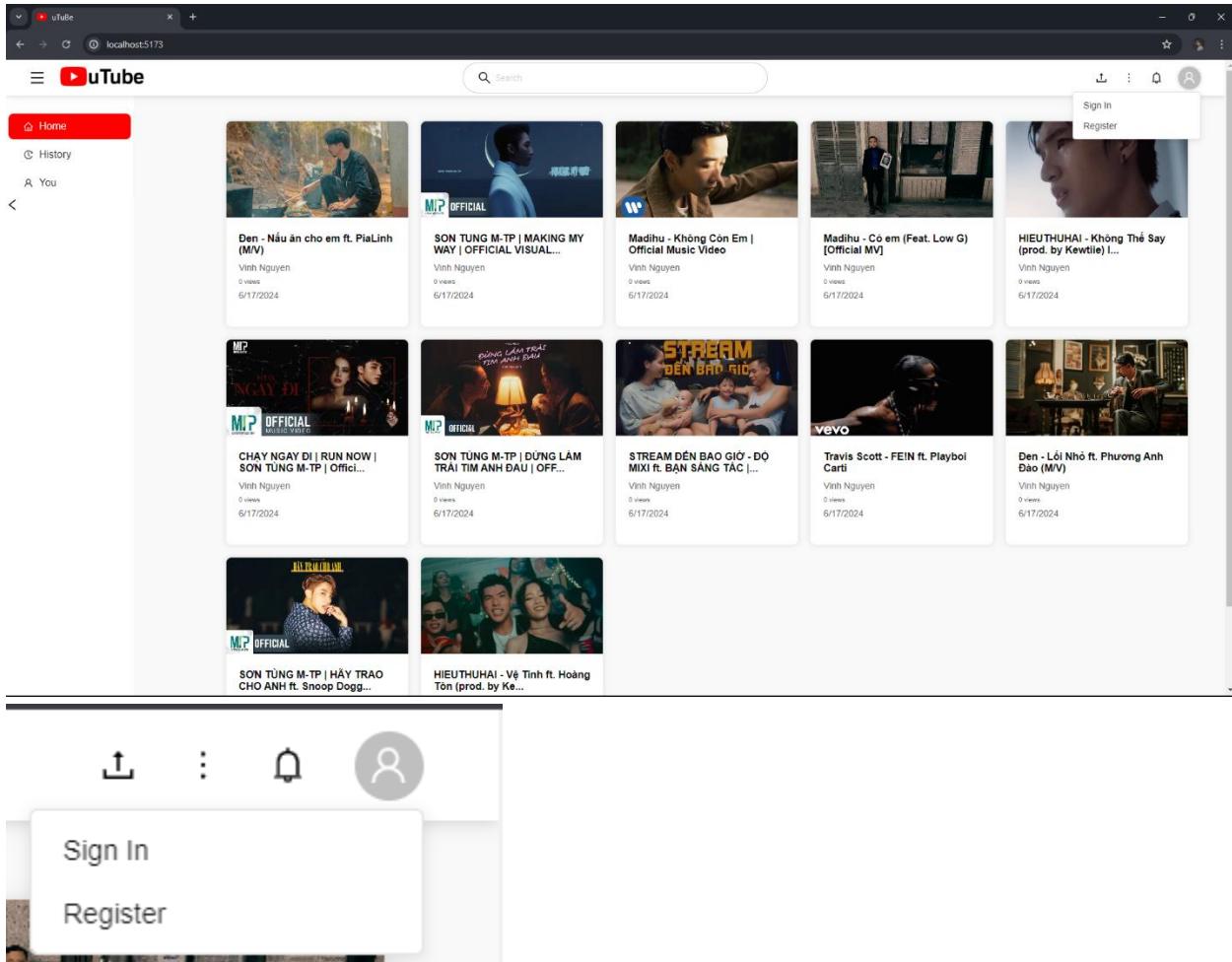
- **/detail:** Retrieve information about this account and support editing this information.
 - **/forgot-password:** Send an email with a parameter, then send an OTP. Check if the OTP is valid.
 - **/admins:** Display all admins in the system.
 - **/users:** Display all users in the system.
 - **/sessions:** Retrieve all sessions of this user.
- **Account -> isUser (/isUser):**
 - **/email:** Check if this email exists on the server.
 - **/username:** Check if this username exists on the server.
- **Admin (/admin):**
 - **/accounts:** Update the role for other users and delete users.
 - **/review:** Retrieve all videos to be reviewed, update their status, or delete them.
- **Home (/home):**
 - **/videos:** Retrieve all videos for the home page.
 - **/search:** Get the keyword and respond with the corresponding video titles.
 - **/search/results:** Retrieve the exact title name and respond with all videos that have this title.
- **User (/user):**
 - **/history:** Retrieve all watched video history of this user. Add new history if this is a new watch review, or delete this history.
 - **/video/management:** Retrieve all videos uploaded by this user, with options to edit or delete these videos.
- **Video (/video):**
 - **/like:** User likes this video.
 - **/dislike:** User dislikes this video.
 - **/view:** Update video views.
 - **/interaction:** Retrieve this video's interactions (likes, dislikes, views).
 - **/information:** Retrieve this video's information (title, channel name, etc.).
 - **/upload:** Feature to help users upload this video.
 - **:= id:** Get video ID and stream this video.

III. IMPLEMENTATION:

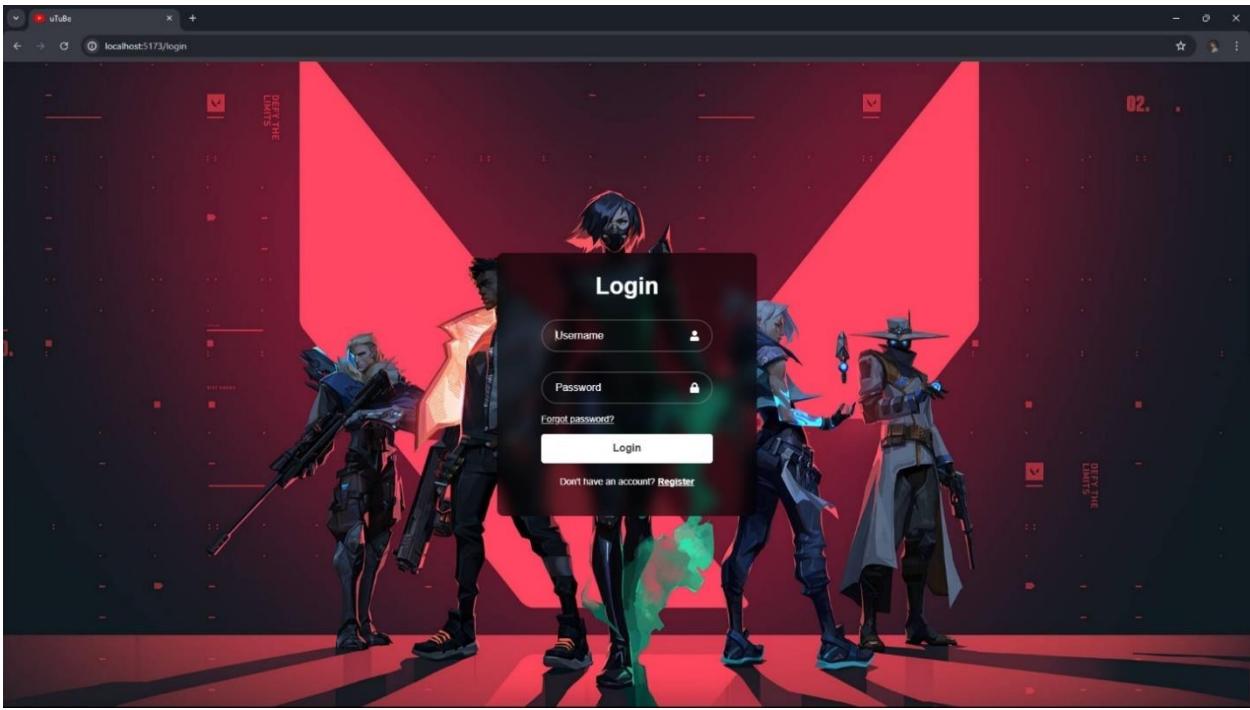
1. Account management:

1.1. Login to the application:

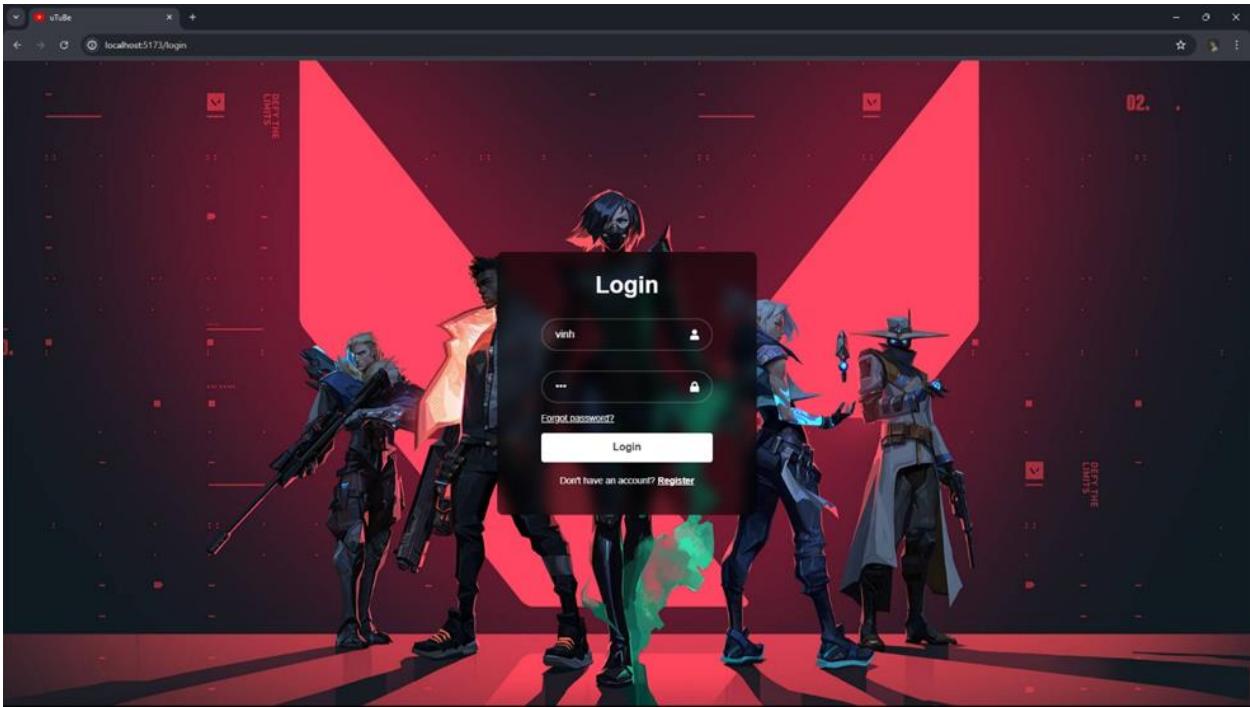
Step 1: Click the account icon:



Step 2: Click “Sign in” the login page will appear:



Step 3: Enter the user and password of your account:



Step 4: Then click “Login”, then the system will check if username and password is true it redirects to Home page. Otherwise, it will return the error message.

- True username and password:

The image consists of two screenshots of a custom YouTube clone application, labeled uTube, running on a local host.

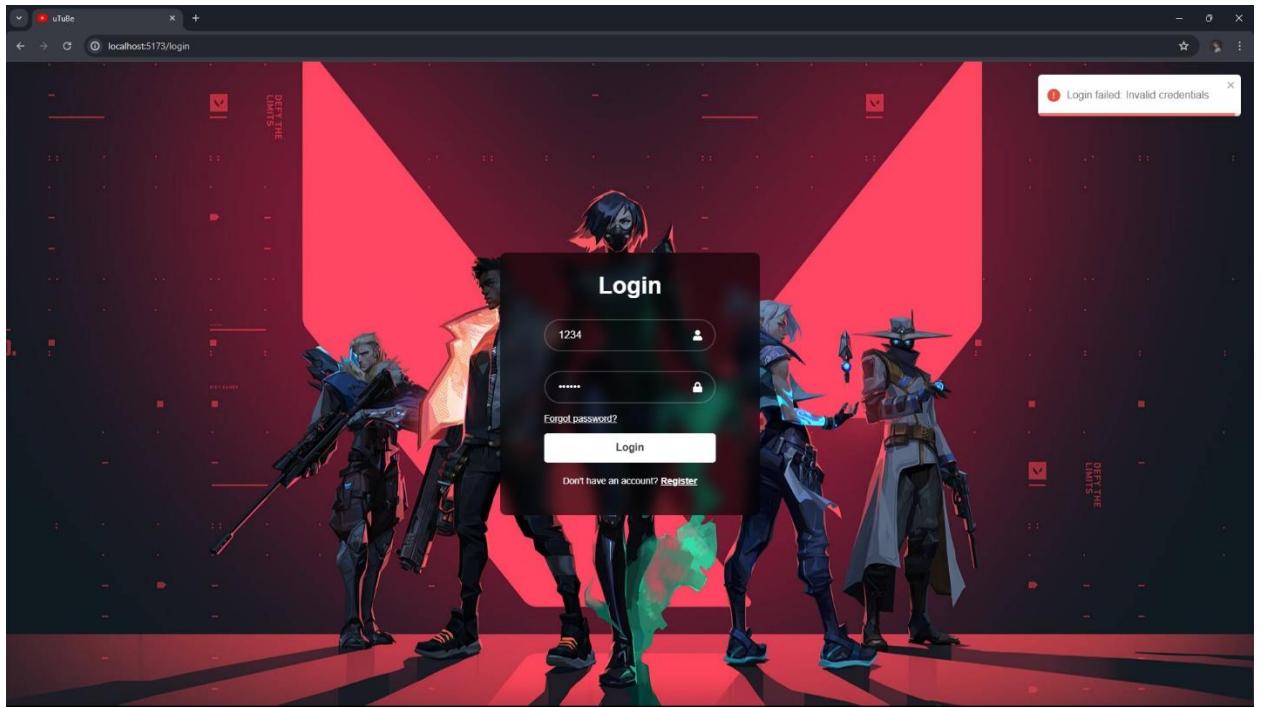
Top Screenshot (Login Screen):

- The background features a dark red and black futuristic theme with four armored characters standing in a row.
- A central modal window titled "Login" contains fields for "username" (with "vinh" entered) and "password".
- Below the fields are links for "Forgot password?" and "Don't have an account? Register".
- A green success message "Login successful" is displayed in the top right corner.

Bottom Screenshot (Home Screen):

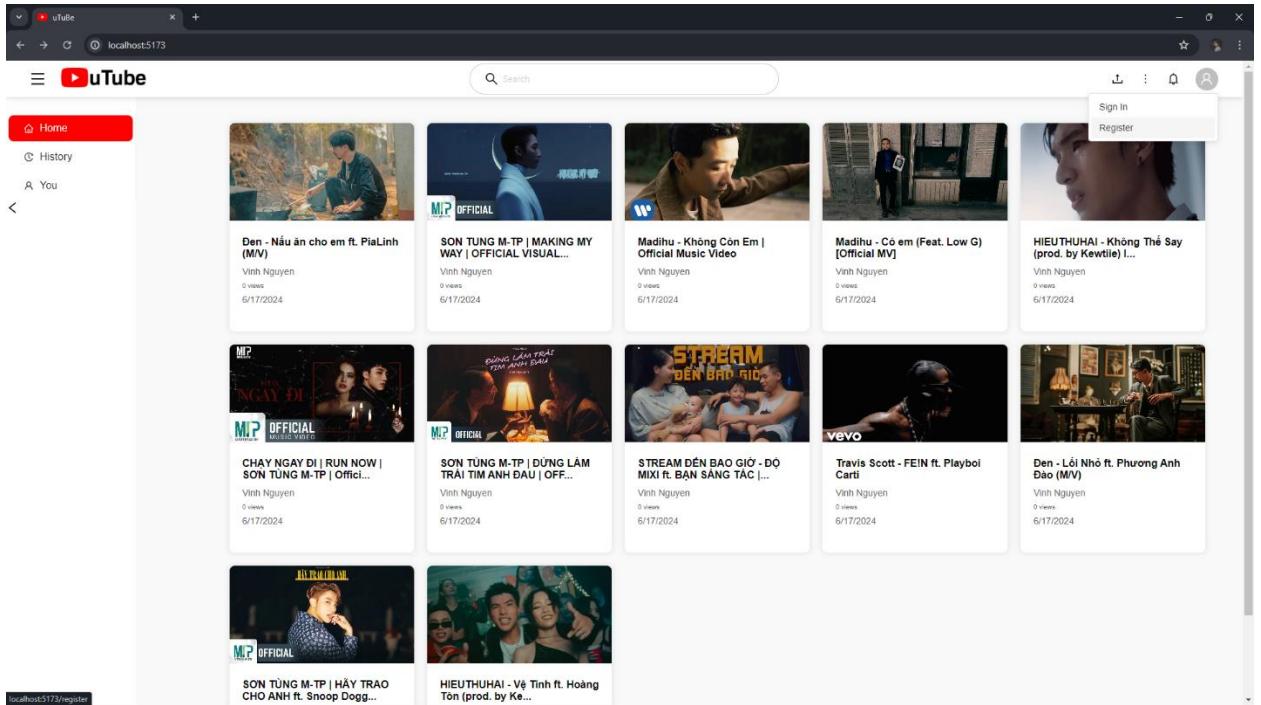
- The interface includes a sidebar with "Home", "History", and "You" options.
- The main area displays a grid of video thumbnails, each with a title, views, and upload date.
- Sample video titles include:
 - Đen - Nẫu ăn cho em ft. PiaLinh (MV)
 - SƠN TÙNG M-TP | MAKING MY WAY | OFFICIAL VISUAL....
 - Madhu - Không Còn Em | Official Music Video
 - Madhu - Cố em (Feat. Low G) [Official MV]
 - HIEUTHUHAI - Không Thể Say (prod. by Kewtlike) [...]
 - CHẠY NGAY ĐI | RUN NOW | SƠN TÙNG M-TP | Official MV
 - SƠN TÙNG M-TP | DÙNG LÀM TRÁI TIM ANH ĐAU | OFF...
 - STREAM ĐÊN BÃO GIỜ - DỘ MIXI ft. BAN SÁNG TÁC [...]
 - Travis Scott - FEIN ft. Playboy Carti
 - Đen - Lời Nhớ ft. Phương Anh Dao (MV)
 - SƠN TÙNG M-TP | HAY TRAO CHO ANH ft. Snoop Dogg...
 - HIEUTHUHAI - Vệ Tinh ft. Hoàng Tôn (prod. by Ke...)

- If wrong username or password. Appearing error message:

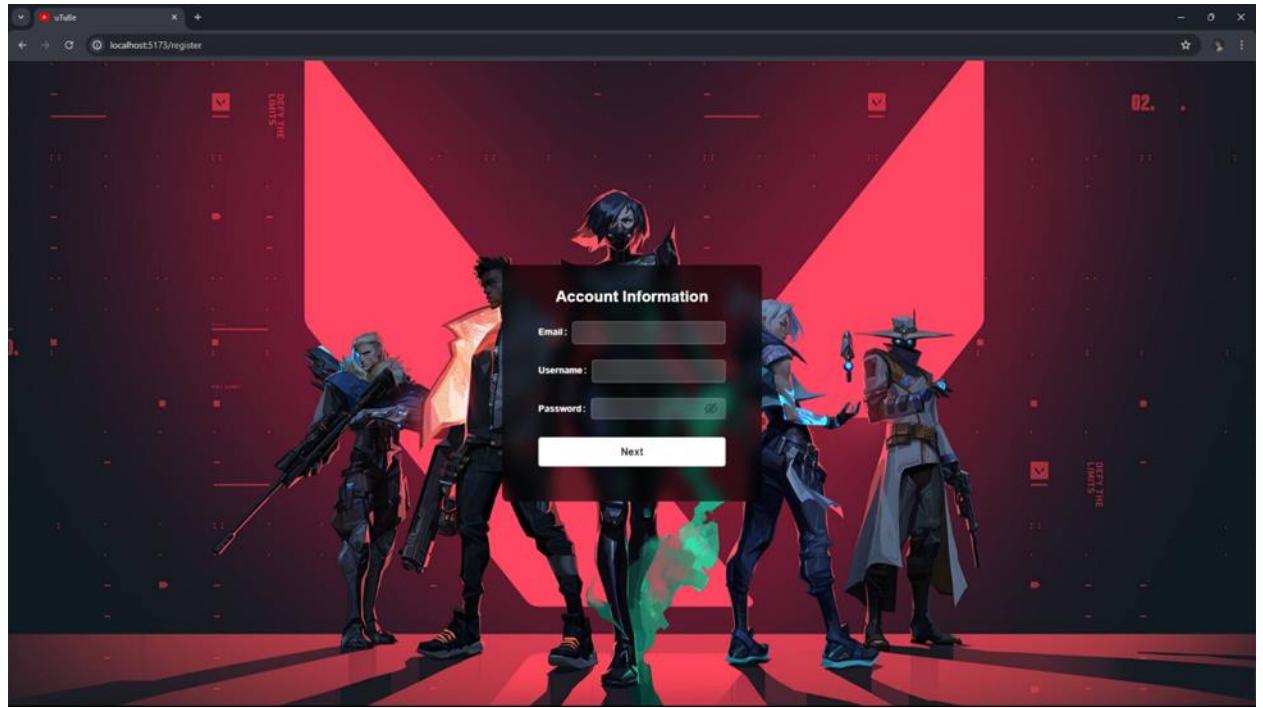


1.2. Register:

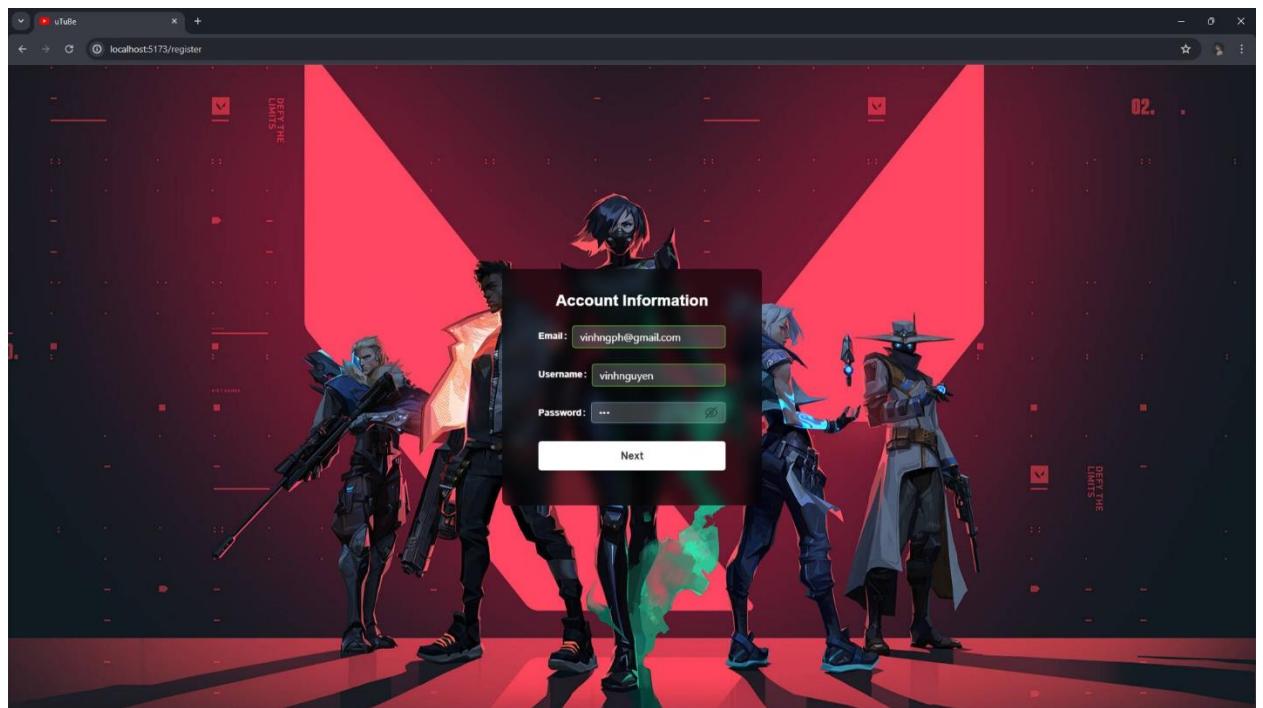
- At home page, click “Account icon” then click “Register”:



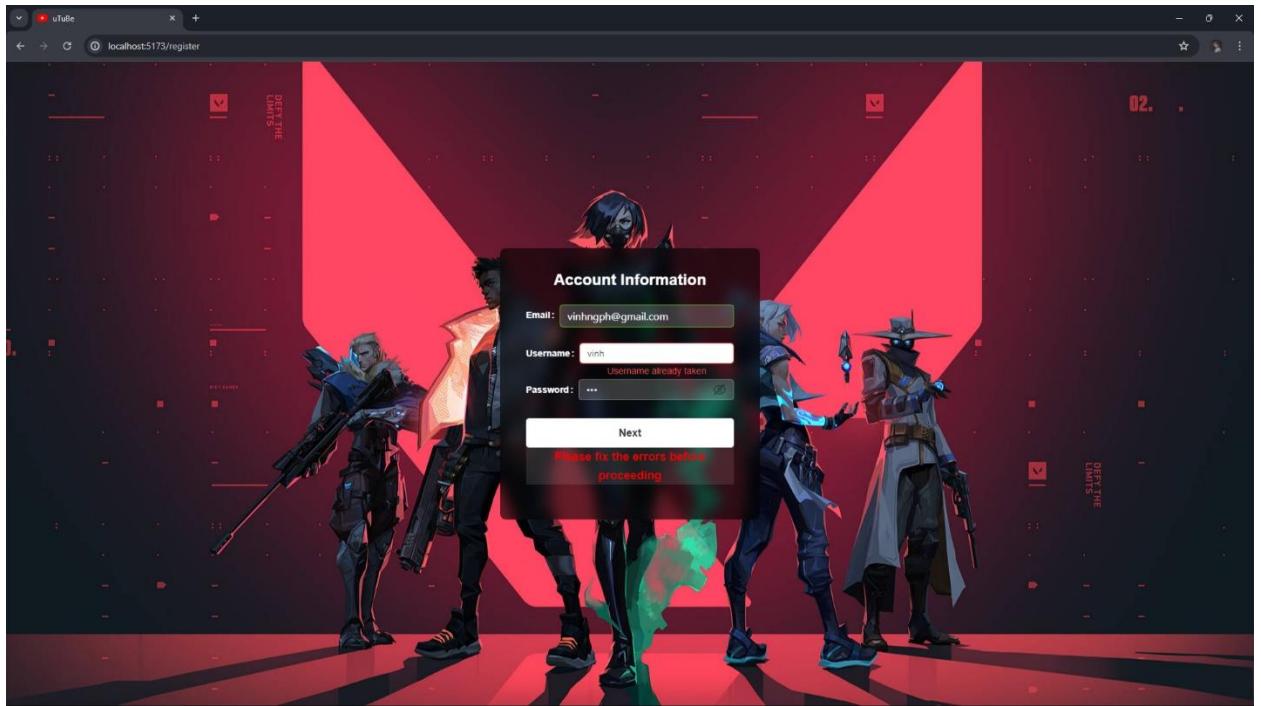
- Then, the Register page will appear, enter the required information:



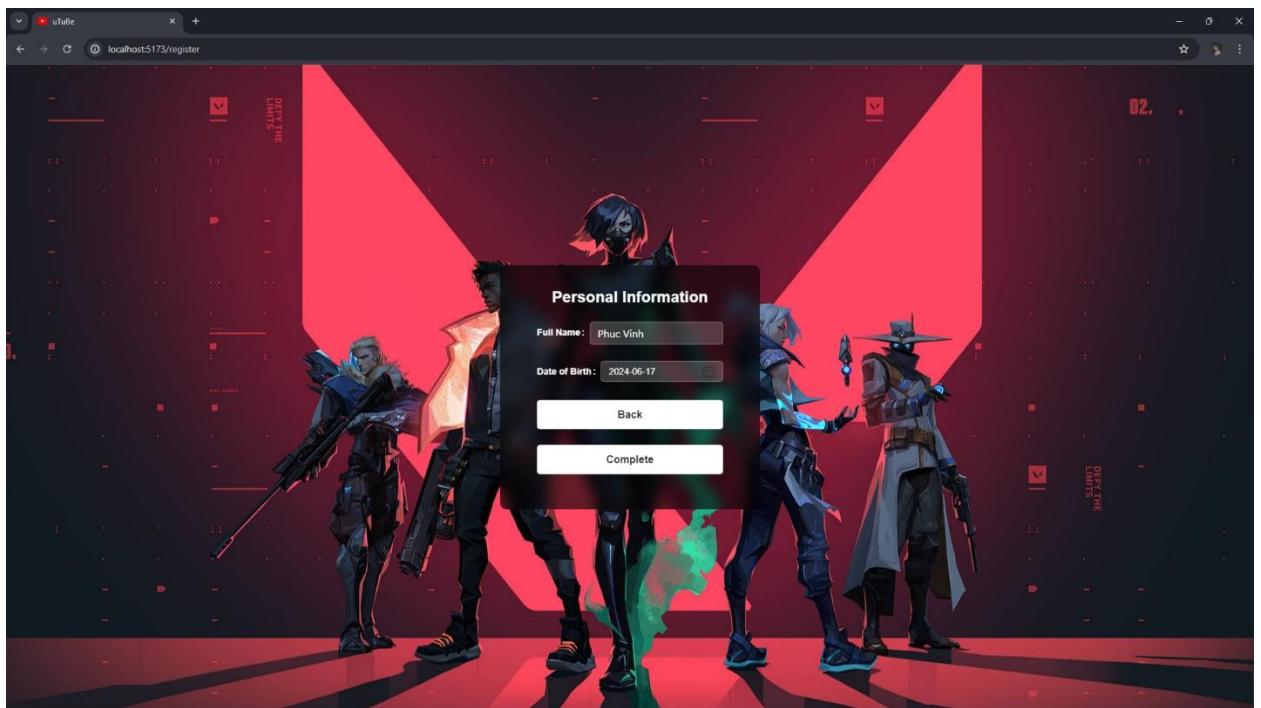
- If username or password input do not exist, the border will be green color.



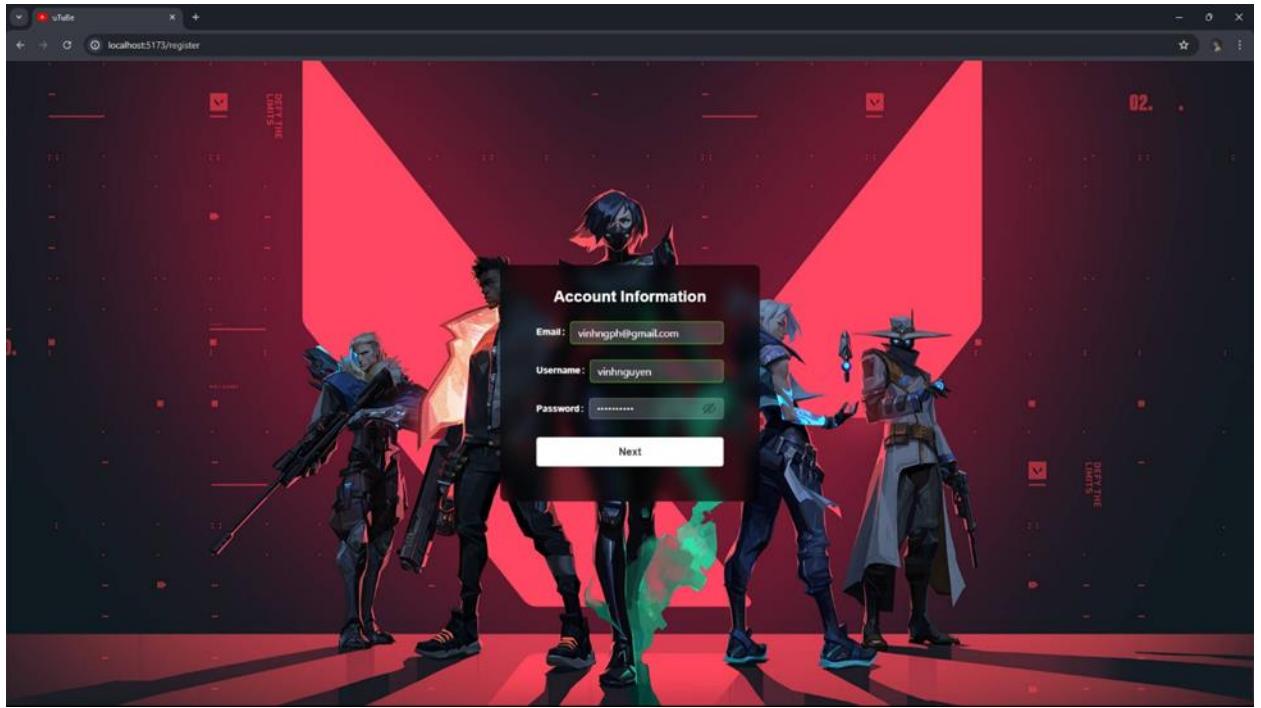
- If username or password input exist, the border will be red color within error message, and user cannot move to next step.



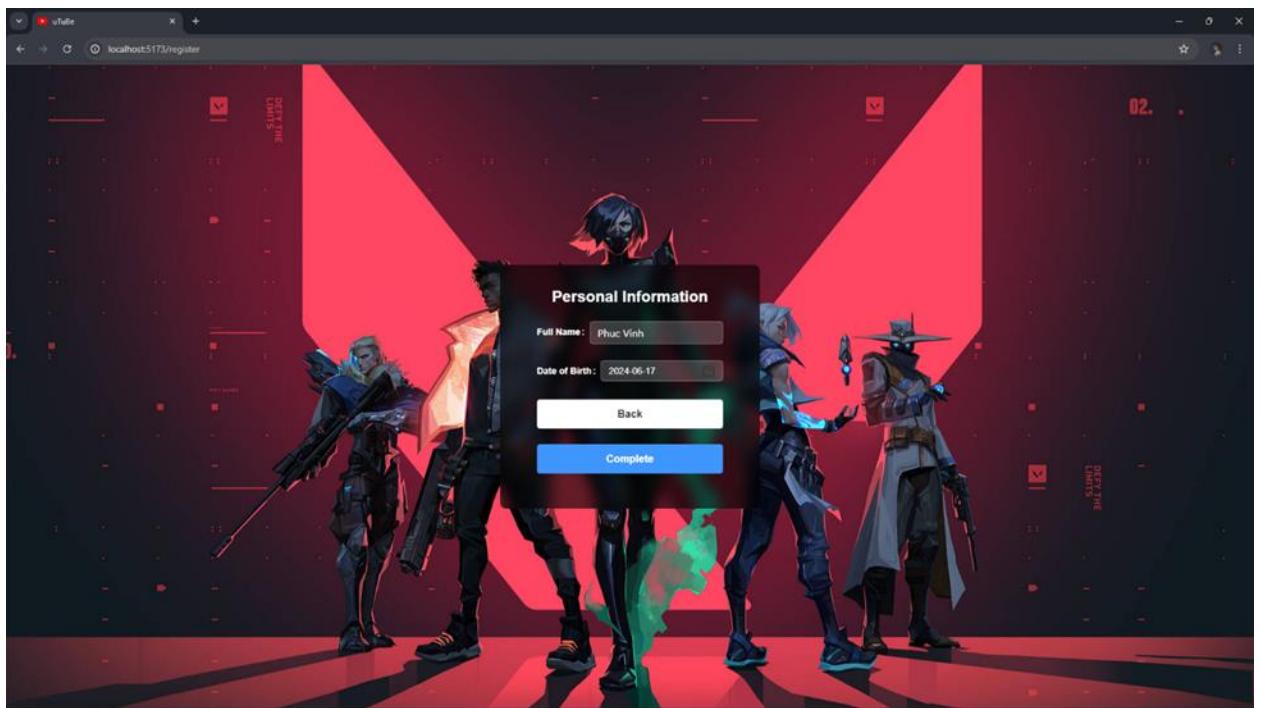
- Next step, input the required information:

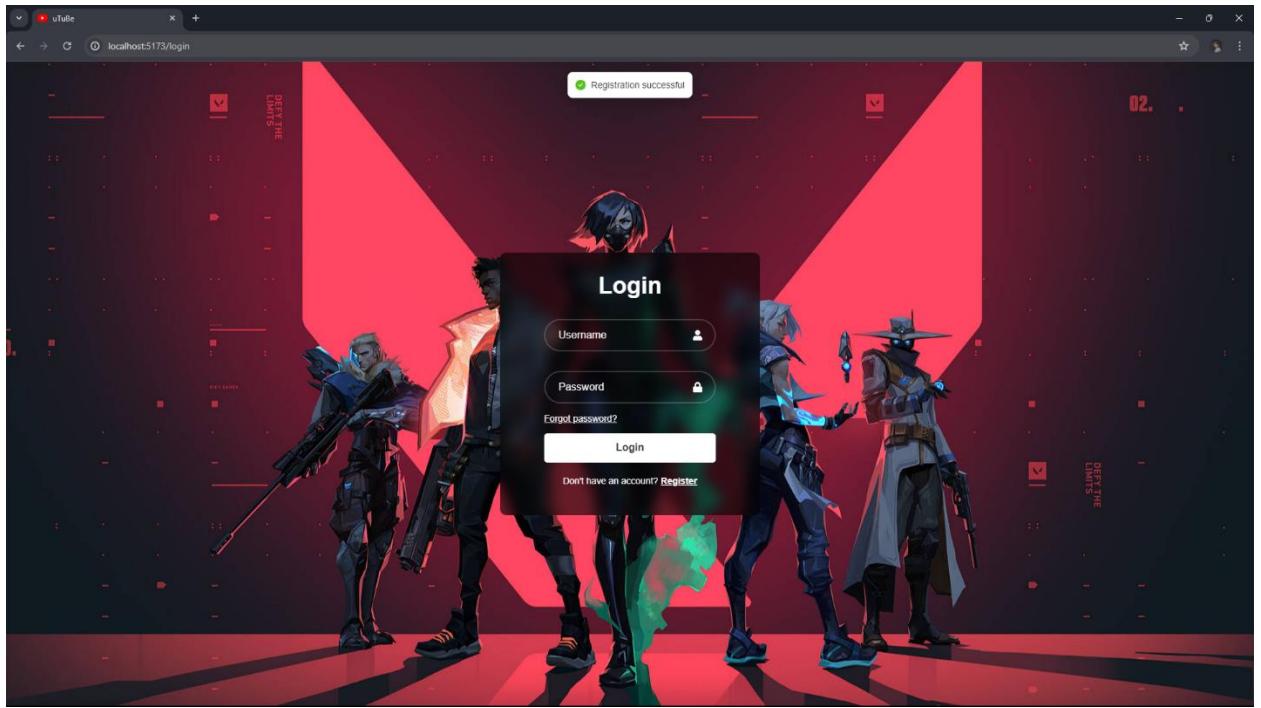


- Notice, the user can back to the previous step:

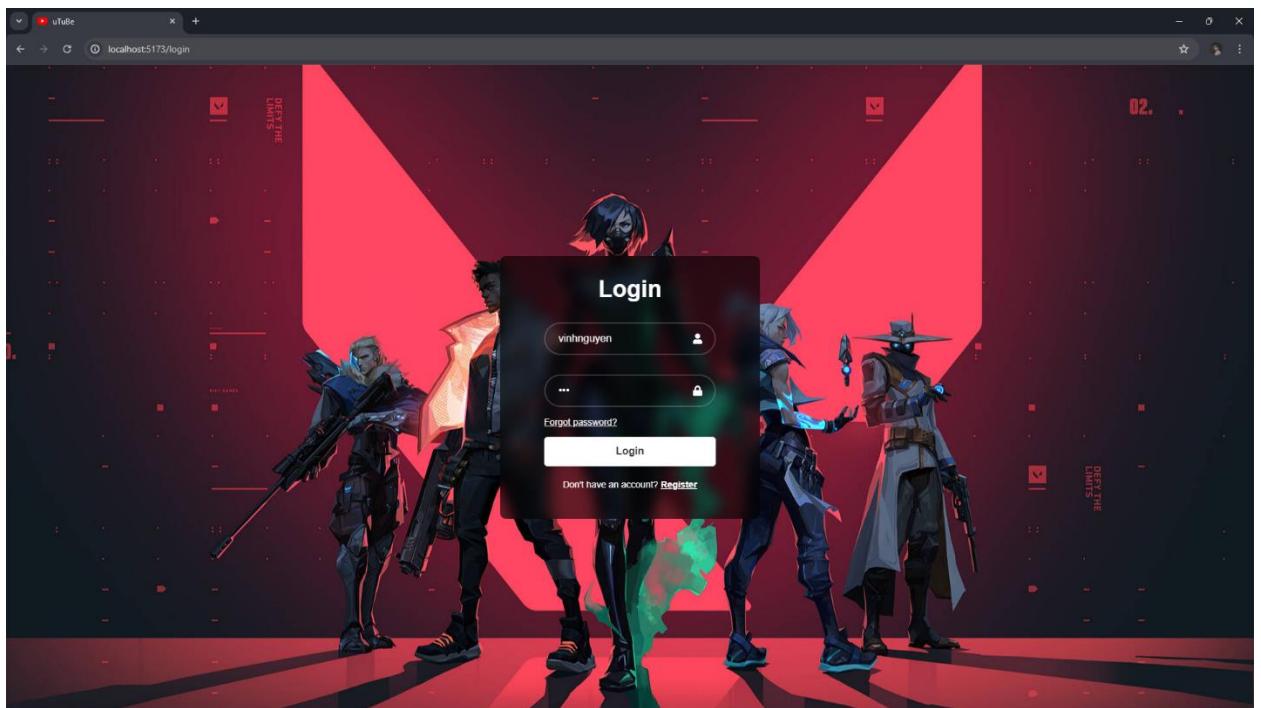


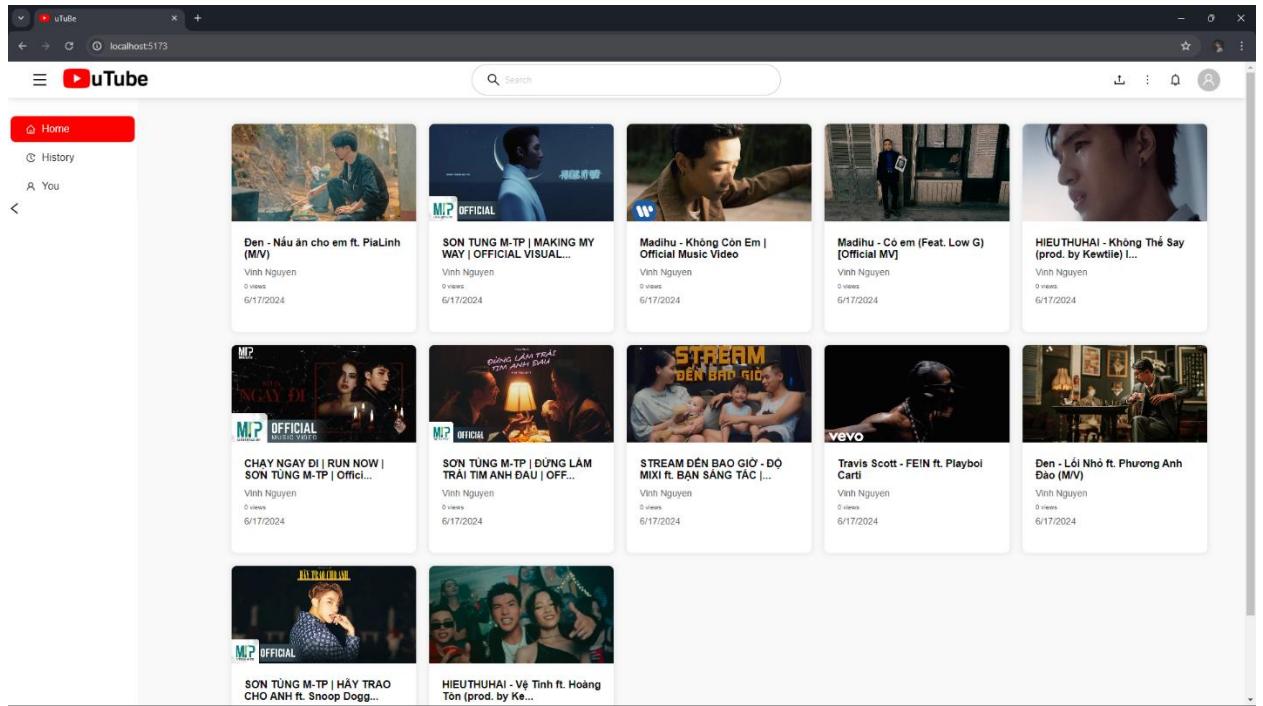
- After completed, turn to login page:





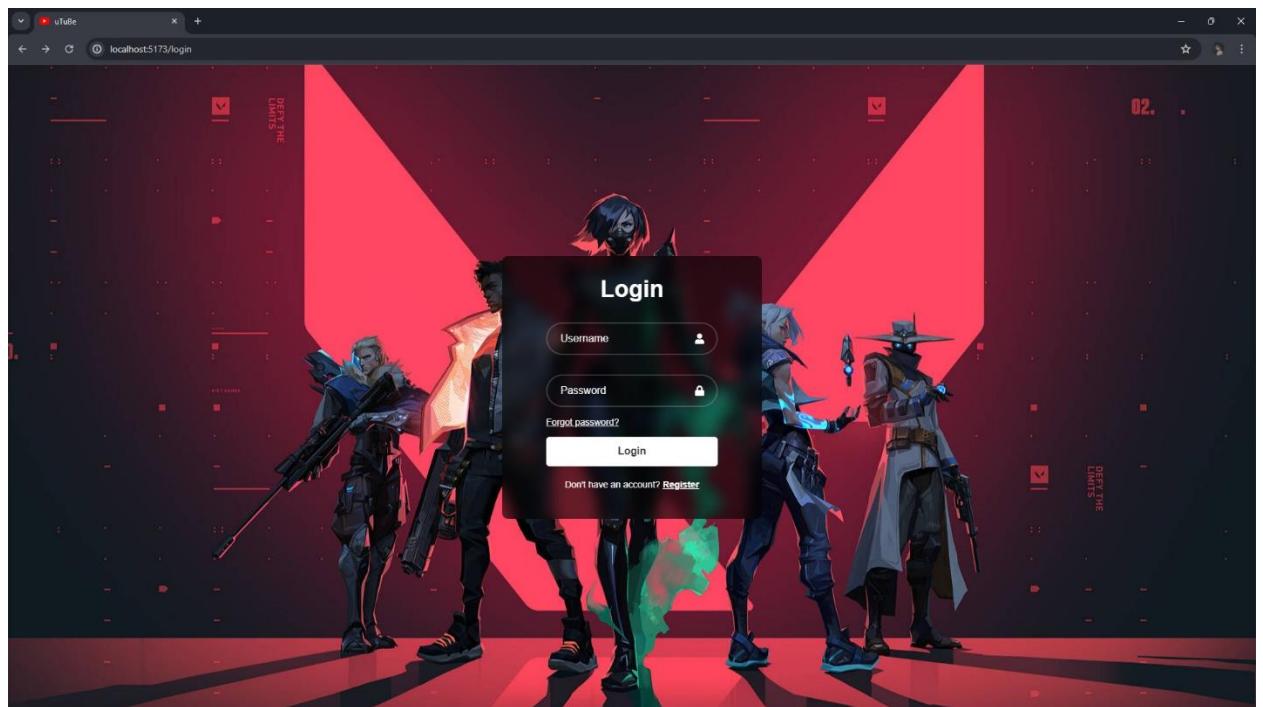
- Enter the previous registered account:





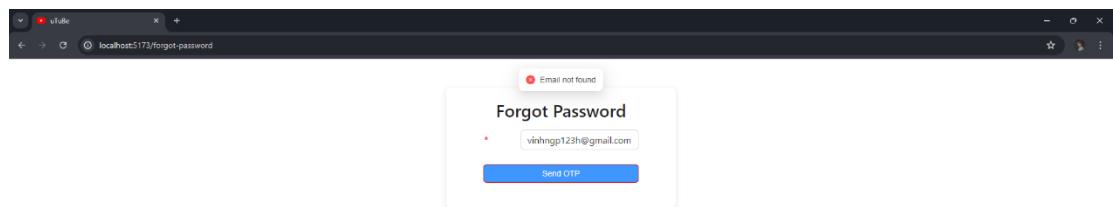
1.3. Forgot password:

- Step 1, click “Forgot password”:



- Step 2, enter the email of registered account (if this email does not use, cannot send OTP mail):

- Wrong email:



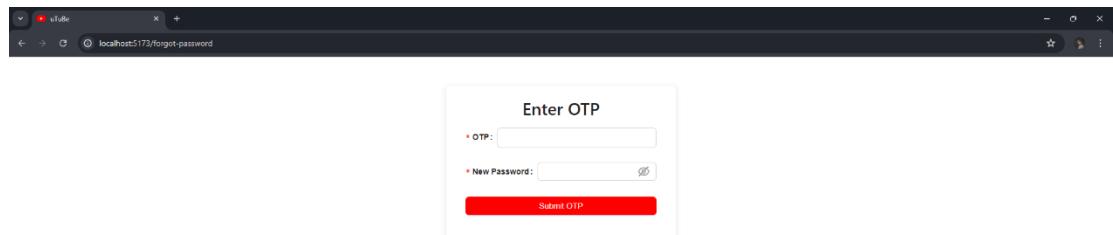
Email not found

Forgot Password

vnhngh123@gmail.com

Send OTP

-
- Accepted email, move to next step:



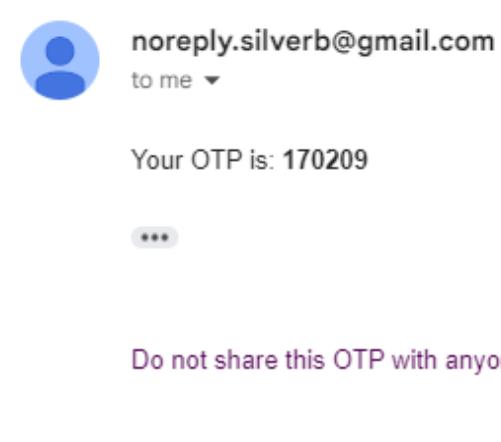
Enter OTP

OTP:

New Password: 

Submit OTP

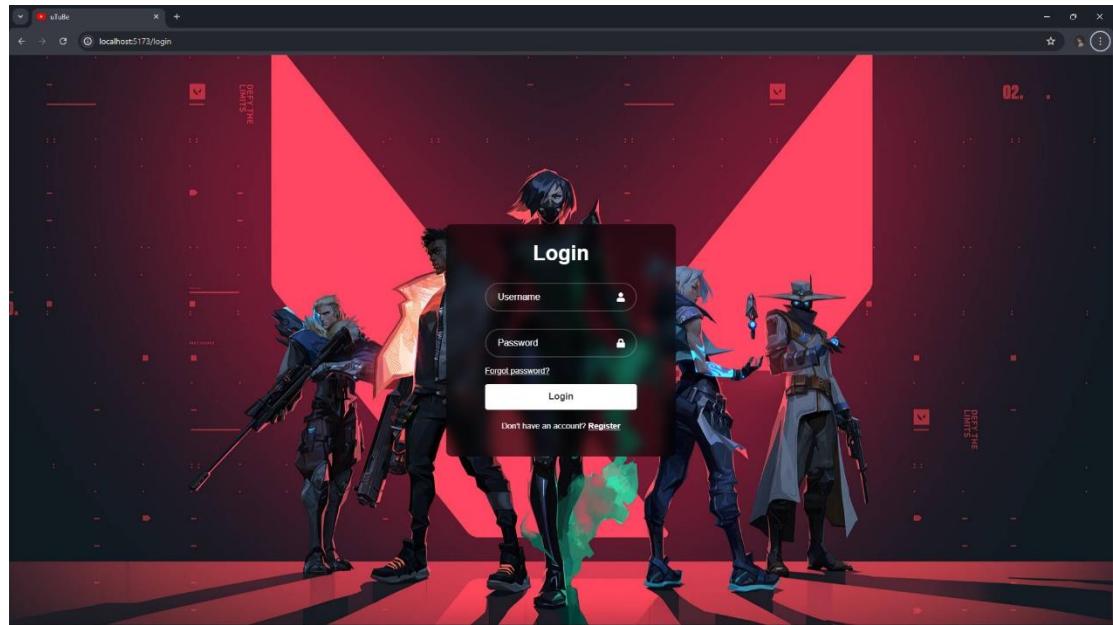
- OTP mail:



- Then, enter the required information:

A screenshot of a web browser window titled 'YouTube' with the URL 'localhost:5173/forgot-password'. The page displays an 'Enter OTP' form with two input fields: 'OTP' containing '170209' and 'New Password' containing '123456'. A red 'Submit OTP' button is at the bottom.

- If true OTP, redirect to login page, then enter new password:



1.4. Account information:

Step 1, click the “Account information” below the account icon:

Step 2, the information page:

YouTube

localhost:5173/account-info

Home History You <

Account Details

User ID
1002

Email
vinhngph@gmail.com

Username
vinhnguyen

Full Name
Phuc Vinh

Date of Birth
06/17/2024

Edit

Step 3, the user can edit their information by clicking “Edit” button. They can change Full name and Date of birth:

YouTube

localhost:5173/account-info

Home History You <

Account Details

User ID
1002

Email
vinhngph@gmail.com

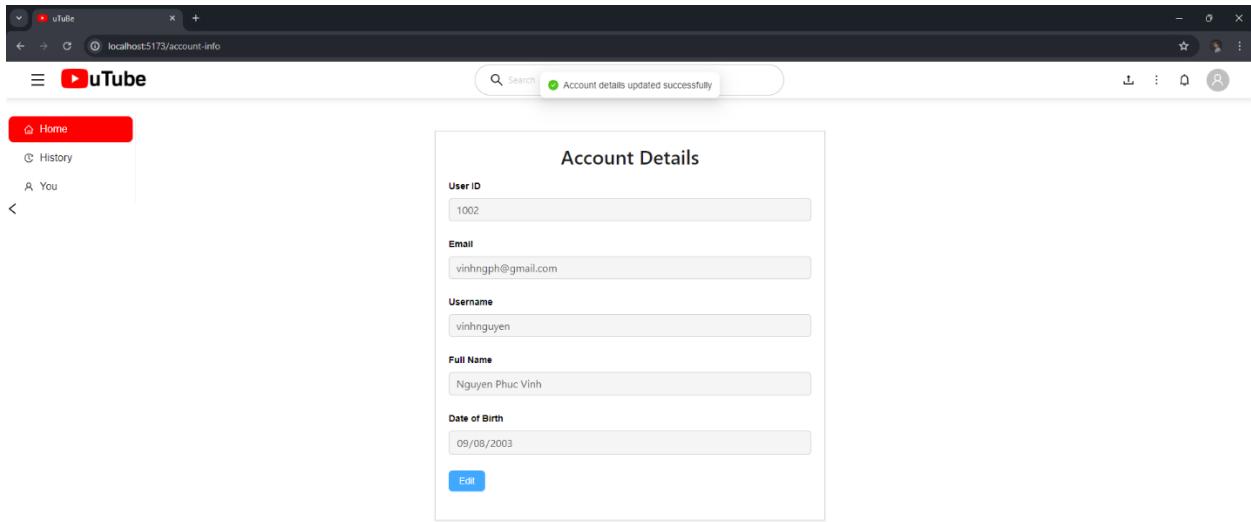
Username
vinhnguyen

Full Name
Nguyen Phuc Vinh

Date of Birth
09/08/2003

Save

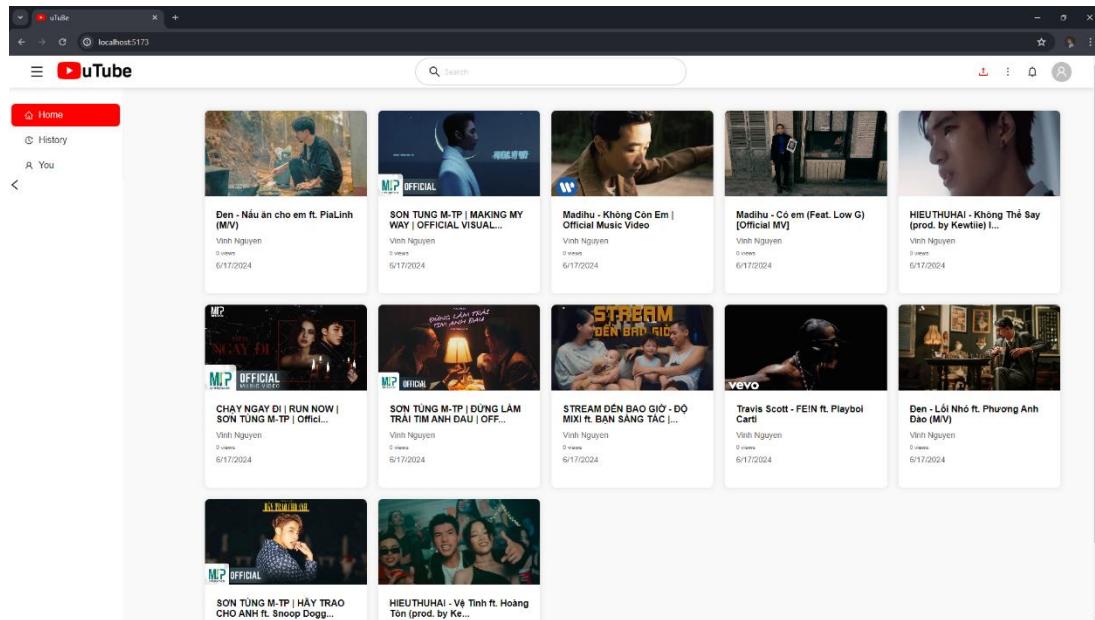
Step 4, click “Save” button:



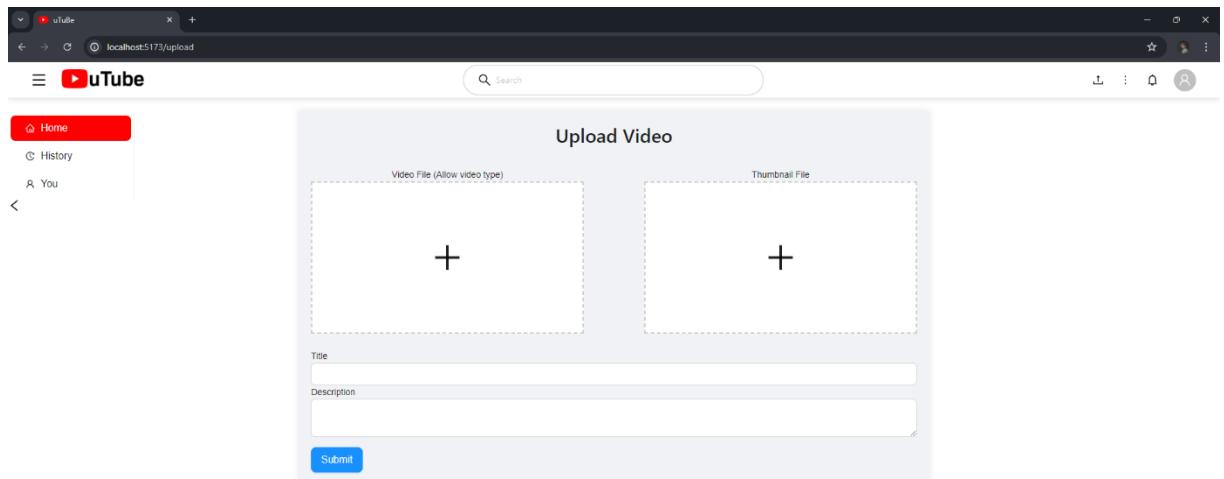
2. Video management:

2.1. Upload video:

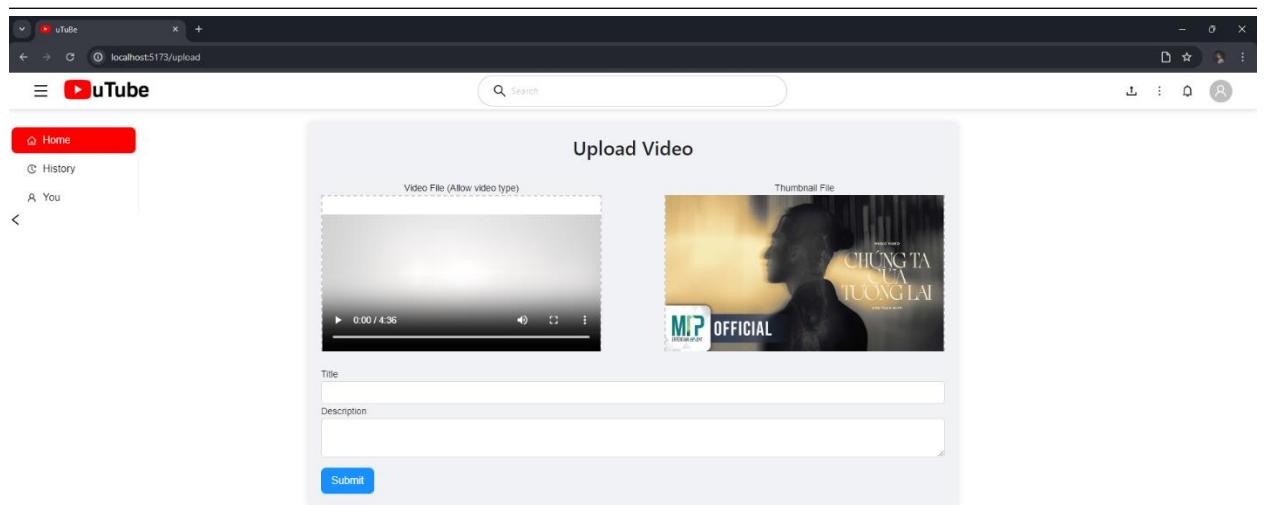
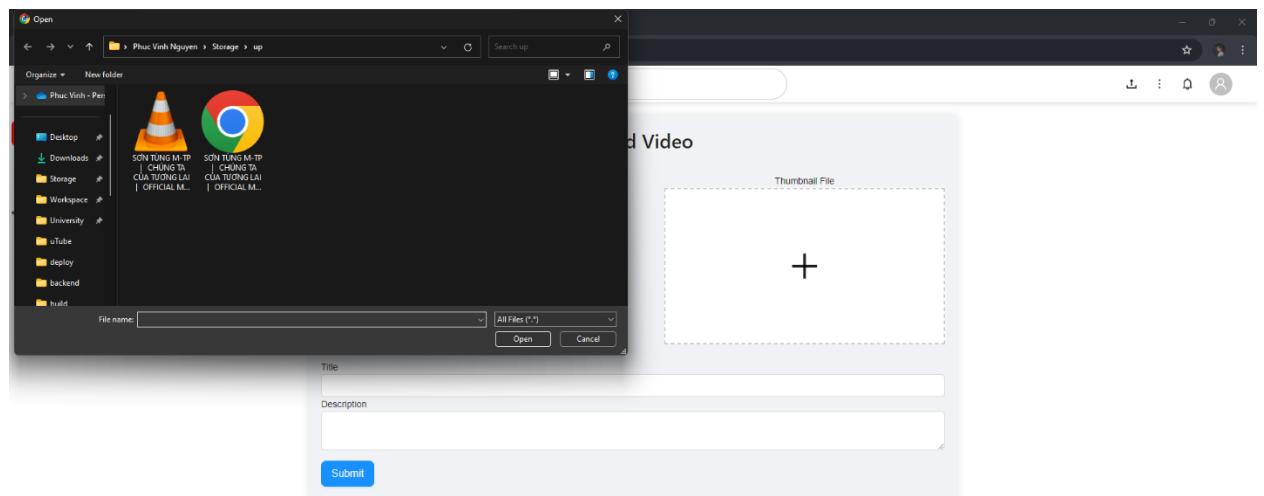
- Click the  button in home page:



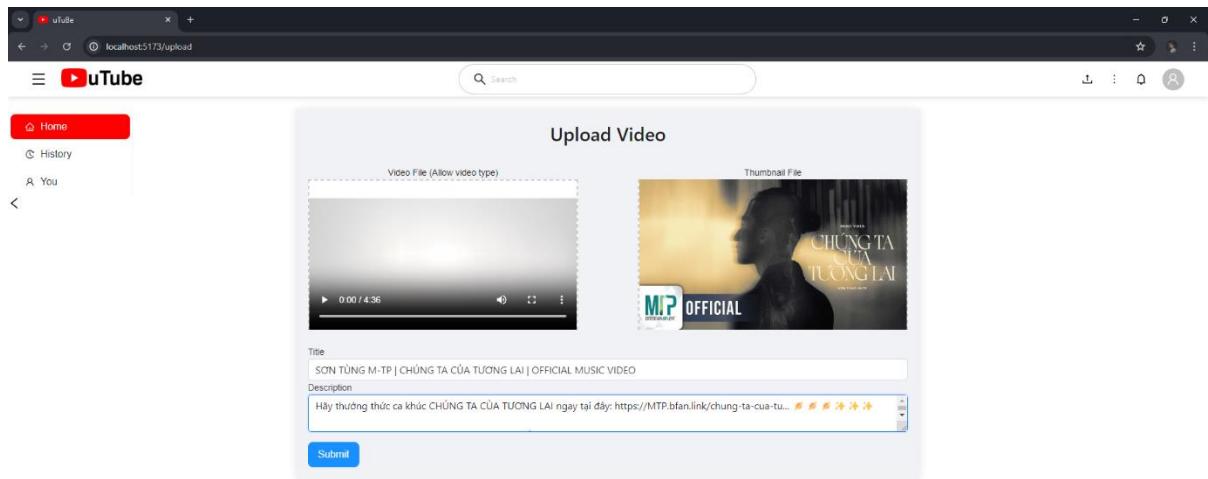
- Prepare the video file and thumbnail file to upload in this upload page:



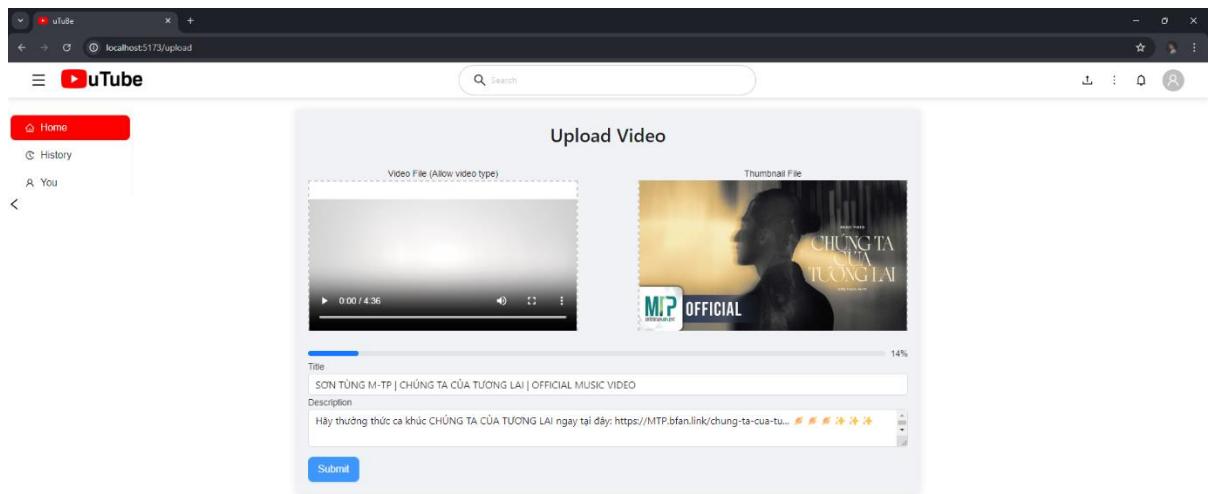
- Choosing the video and thumbnail file:



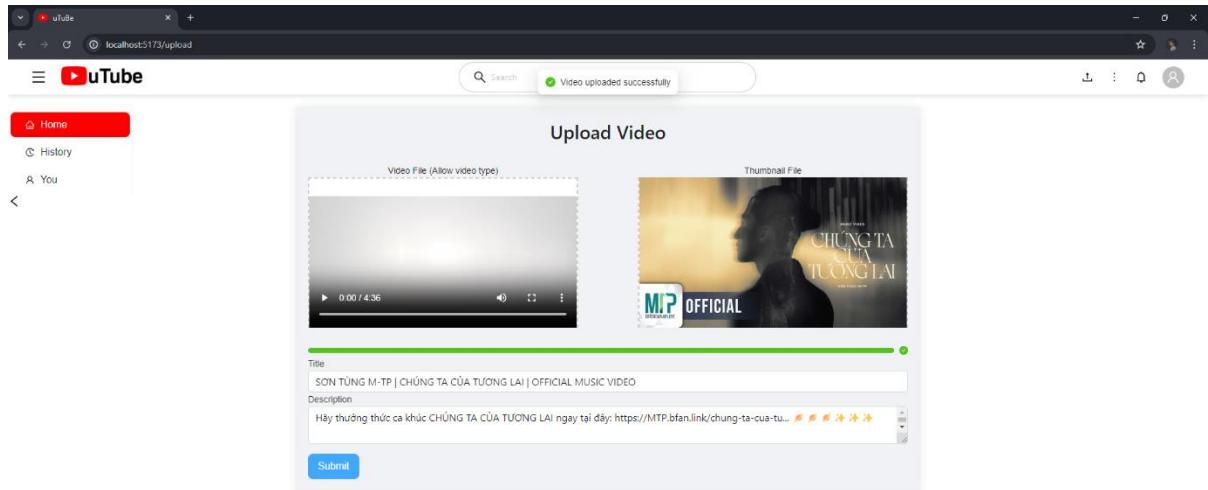
- Then adding the Tile and Description and click “Submit” button to upload video:



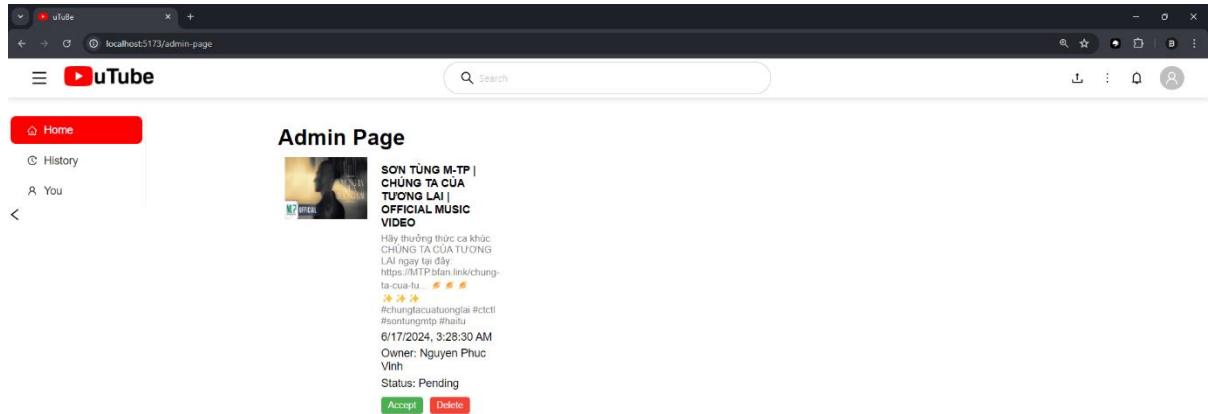
- The upload progress will run:



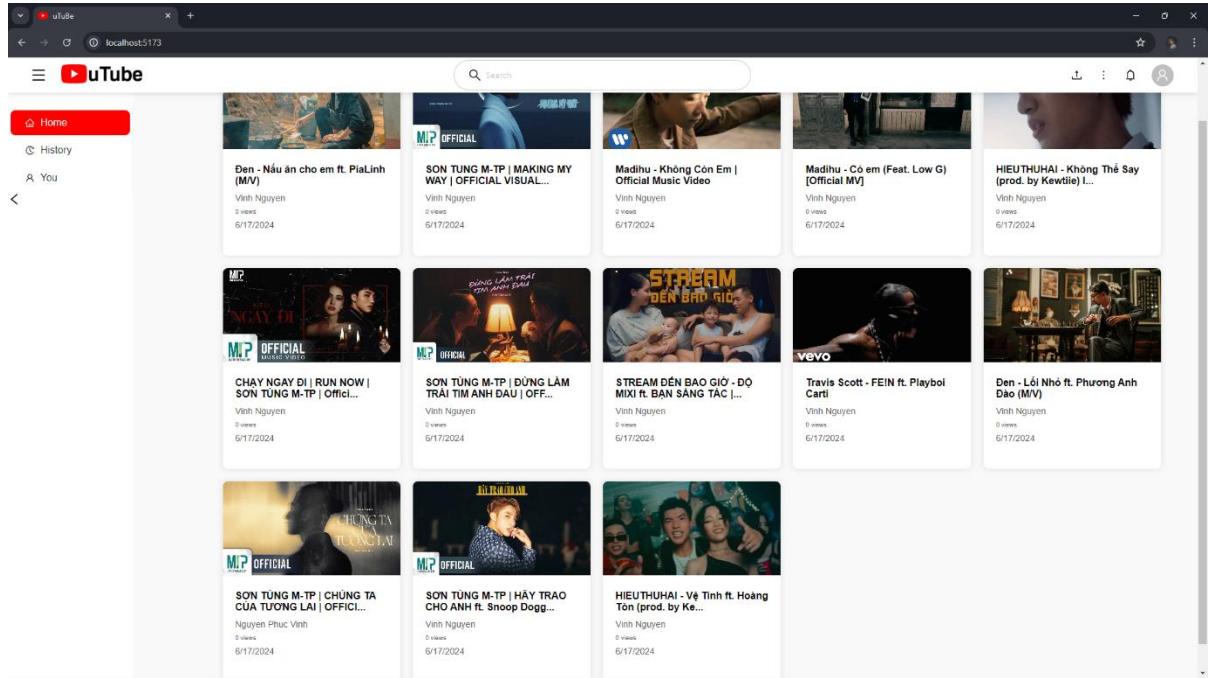
- If done, the progress status will change color to green:



- But, on the home page the video does not appear, because this video needs to be accepted by administrators (this is a function of admin role):

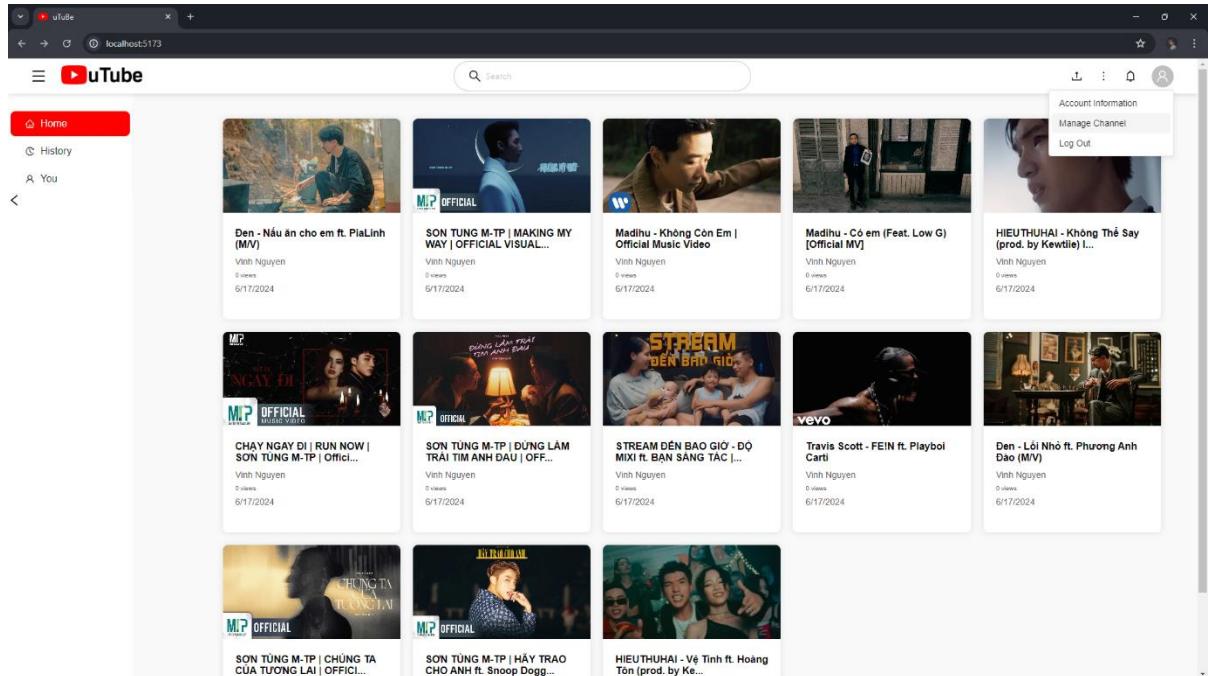


- After accepted, the video appears on home page:

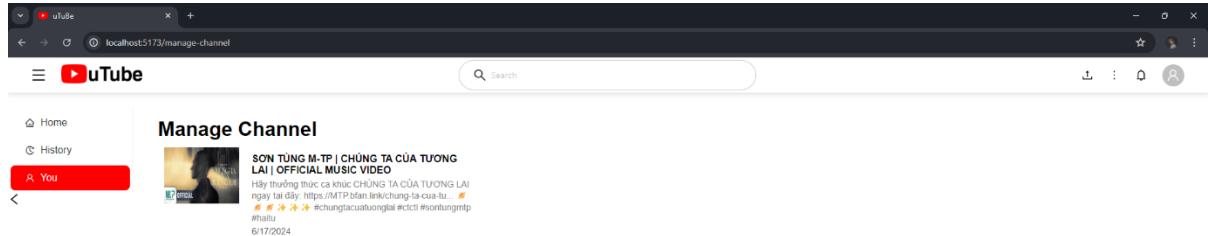


2.2. Manage uploaded video's user:

- Click “Manage Channel” below the account icon:

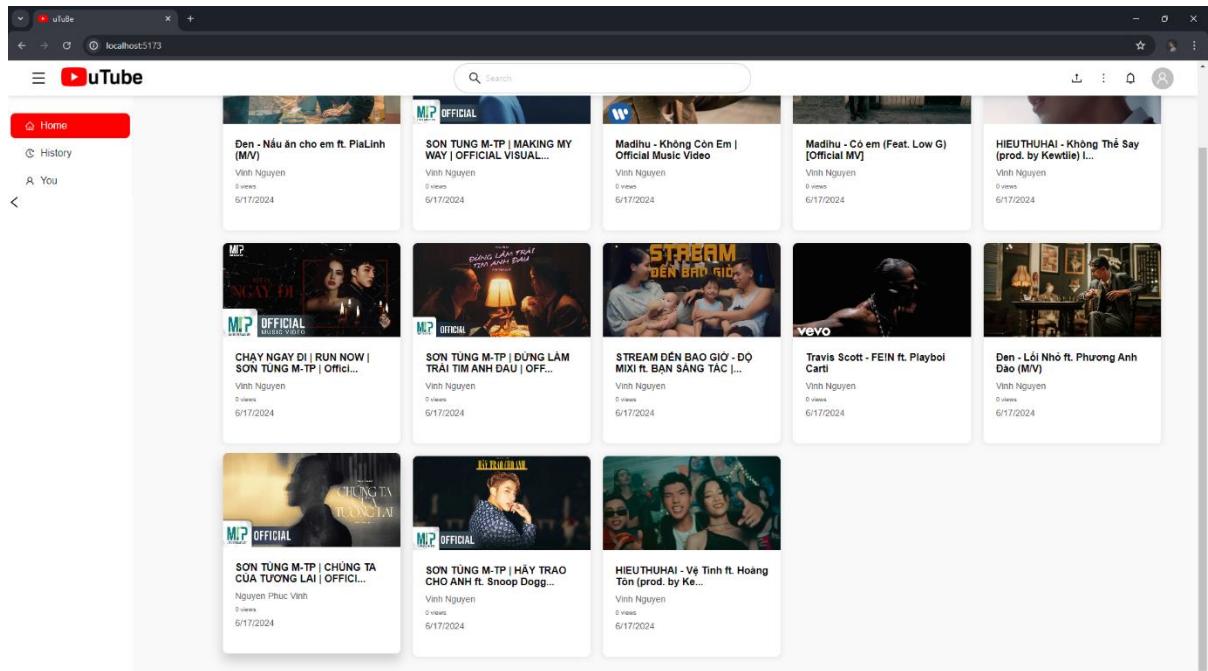


- Then all videos uploaded by this user will be shown:

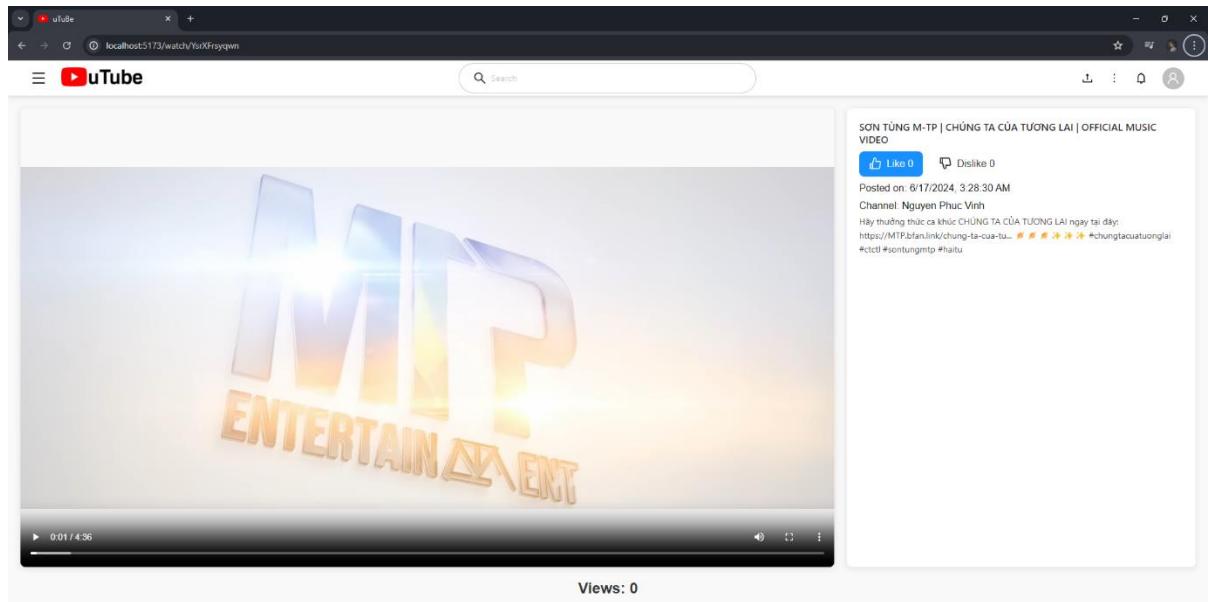


2.3. Watching the video:

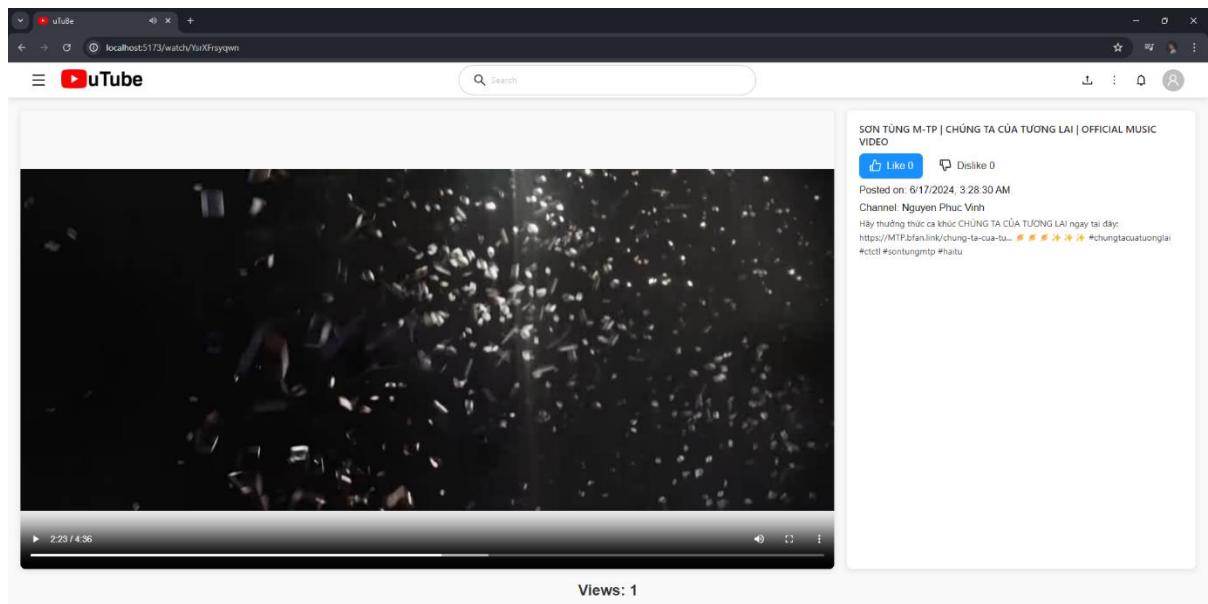
- Choosing the video in home page, I will choose the uploaded previous video:



- This is a viewing page. Users can “Like” or “Dislike”. The view of video will be updated if the total time user spends to watch equal half of video’s duration:



- After half duration of this video, the view increase 1:



- If user like this video, increase like by 1:

SƠN TÙNG M-TP | CHÚNG TA CỦA TƯƠNG LAI | OFFICIAL MUSIC VIDEO

 Like 1  Dislike 0

Posted on: 6/17/2024, 3:28:30 AM

Channel: Nguyen Phuc Vinh

Hãy thưởng thức ca khúc CHÚNG TA CỦA TƯƠNG LAI ngay tại đây:
<https://MTP.bfan.link/chung-ta-cua-tu...> 🎶🎶🎶🎶🎶🎶 #chungtacuatuonglai
#ctctl #sontungmtp #haitu

- If user dislike, remove like before and increase dislike by 1:

SƠN TÙNG M-TP | CHÚNG TA CỦA TƯƠNG LAI | OFFICIAL MUSIC VIDEO

 Like 0  Dislike 1

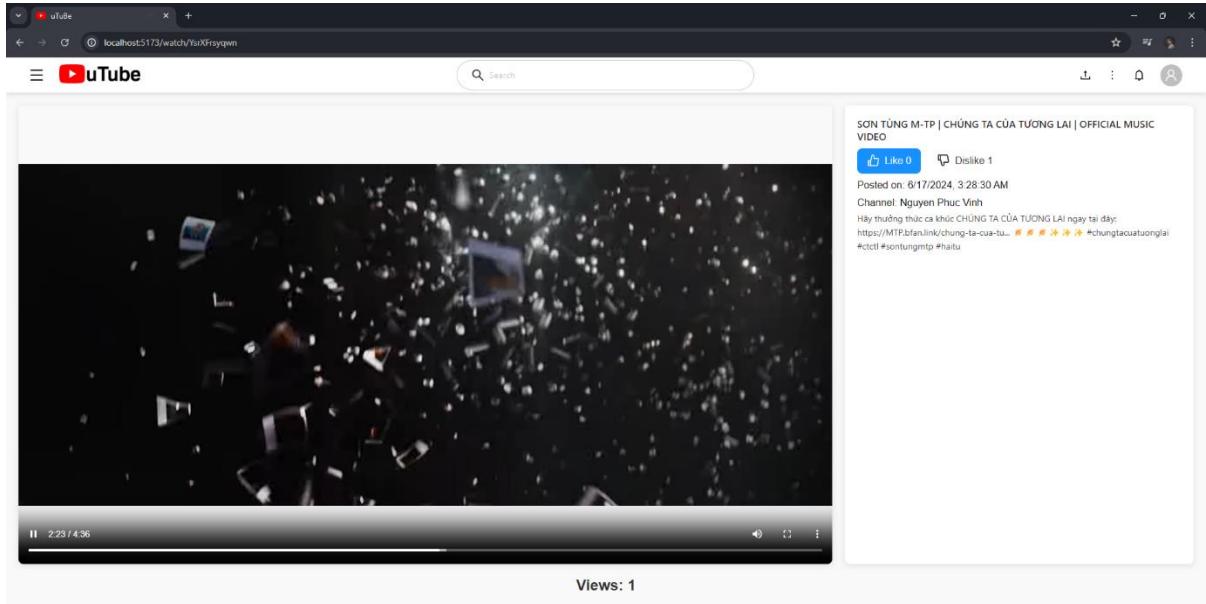
Posted on: 6/17/2024, 3:28:30 AM

Channel: Nguyen Phuc Vinh

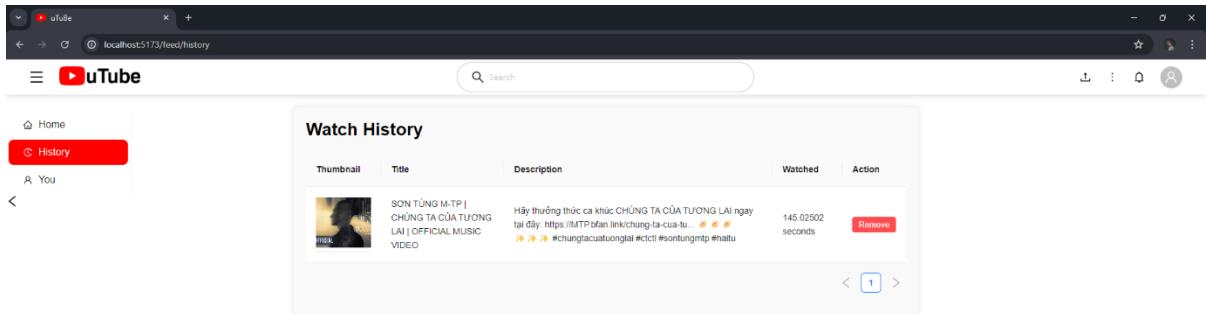
Hãy thưởng thức ca khúc CHÚNG TA CỦA TƯƠNG LAI ngay tại đây:
<https://MTP.bfan.link/chung-ta-cua-tu...> 🎶🎶🎶🎶🎶🎶 #chungtacuatuonglai
#ctctl #sontungmtp #haitu

2.4. User watching history:

- In the previous moment, I have watch “CHÚNG TA CỦA TƯƠNG LAI” and stop at 2:23.
- Now, I come back to this video:

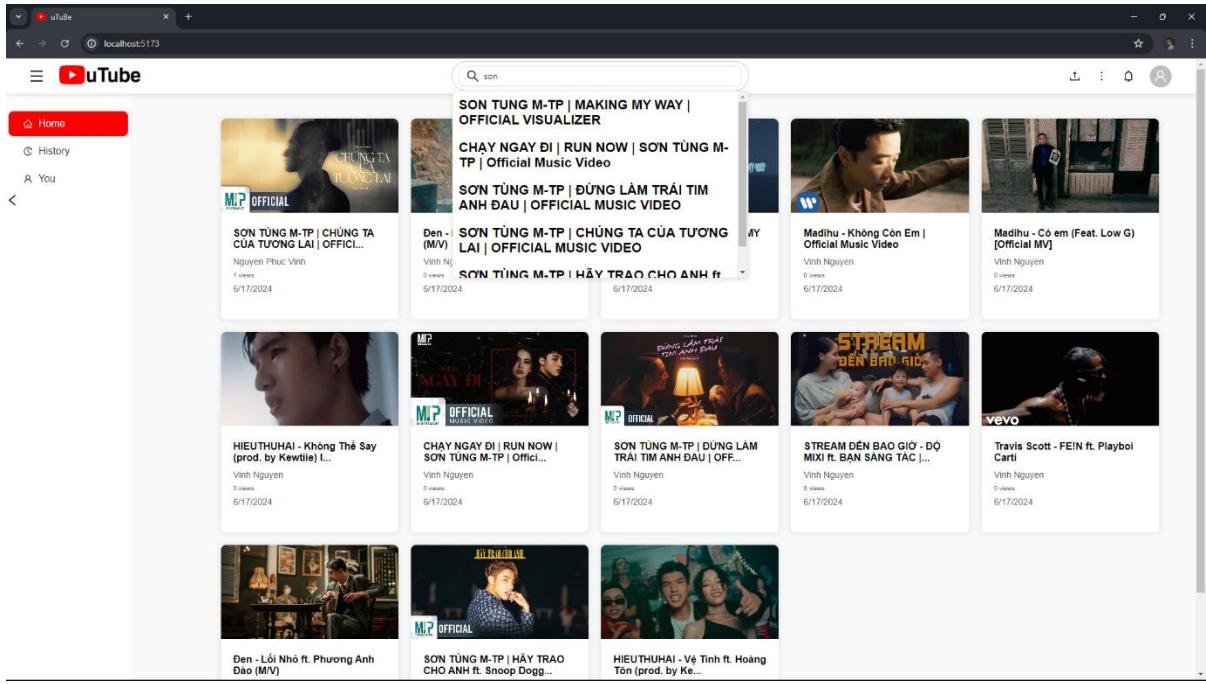


-
- And this video plays continuously at the current stop video's time.
 - It also updates this history in user history page:

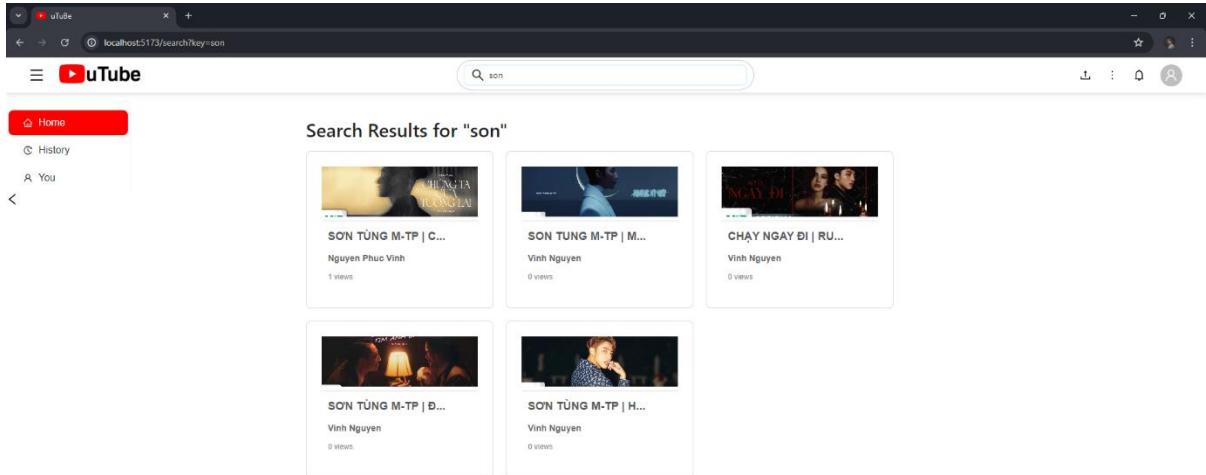


3. Search engine:

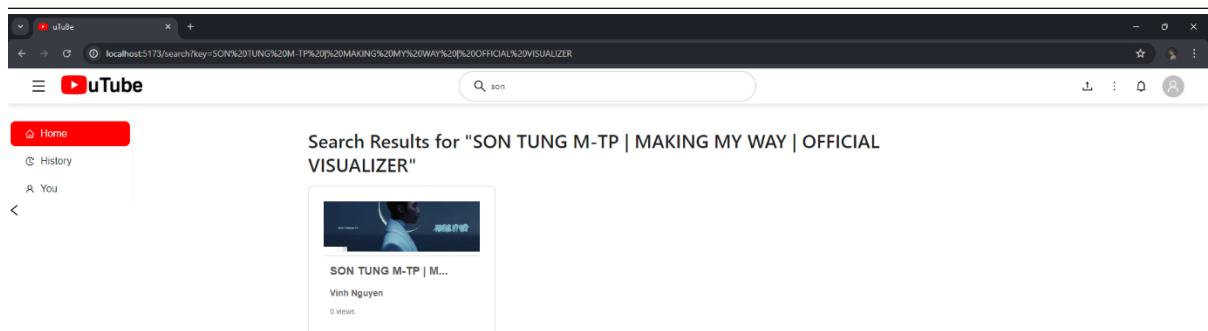
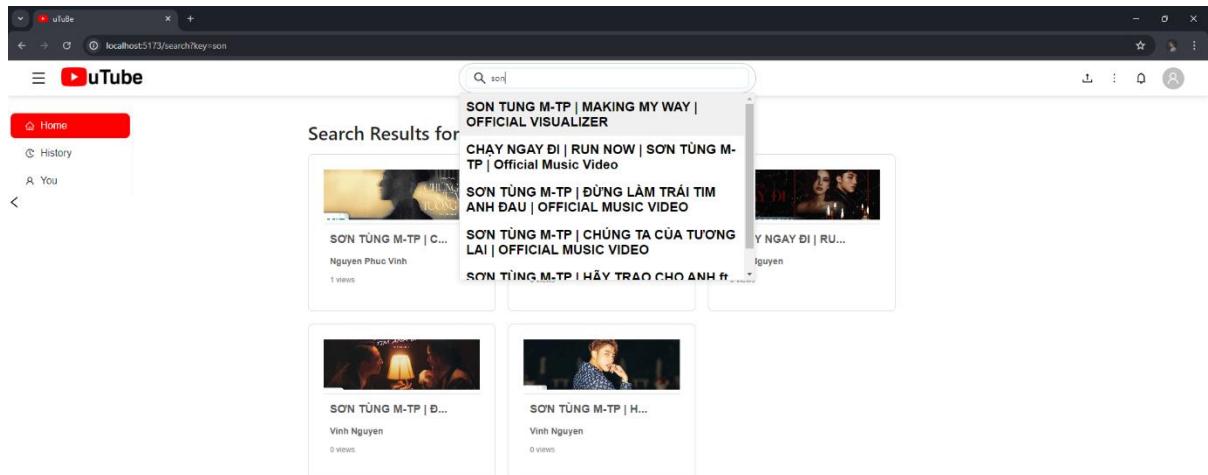
- Input keyword into search's box:



- If I just enter keyword “son” and enter, it will search for this keyword:



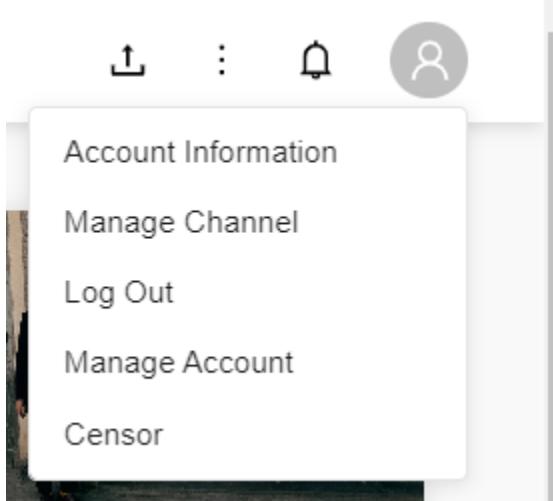
- When I search “son”, then click the search results below, it will search exactly this title name:



4. Admin role

4.1. Manage role for users:

- Click the “Manage Account” field below the account icon:



- The list of users and admins will be shown:

The image shows a web browser window titled "uTube" at the URL "localhost:5173/manage-account". On the left sidebar, there are links for "Home", "History", and "You". The main content area has two sections: "Users" and "Admins".

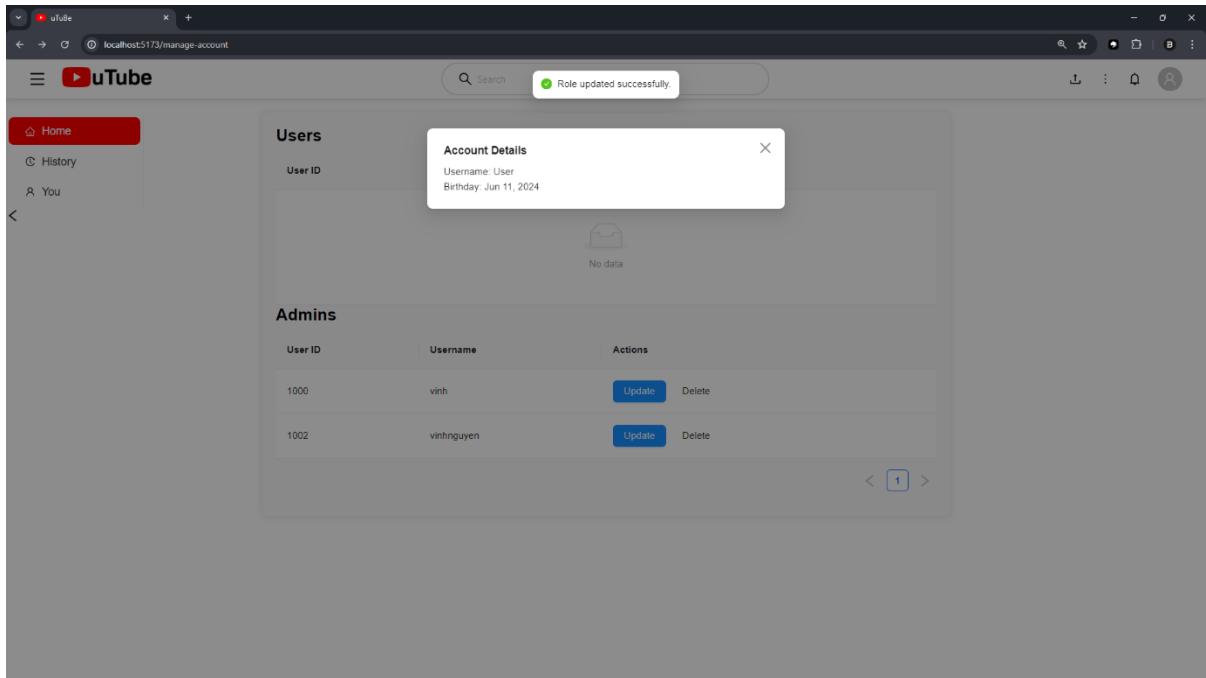
User ID	Username	Actions
1004	user	Update Delete

Below the "Users" table is a page navigation indicator: < [1] >.

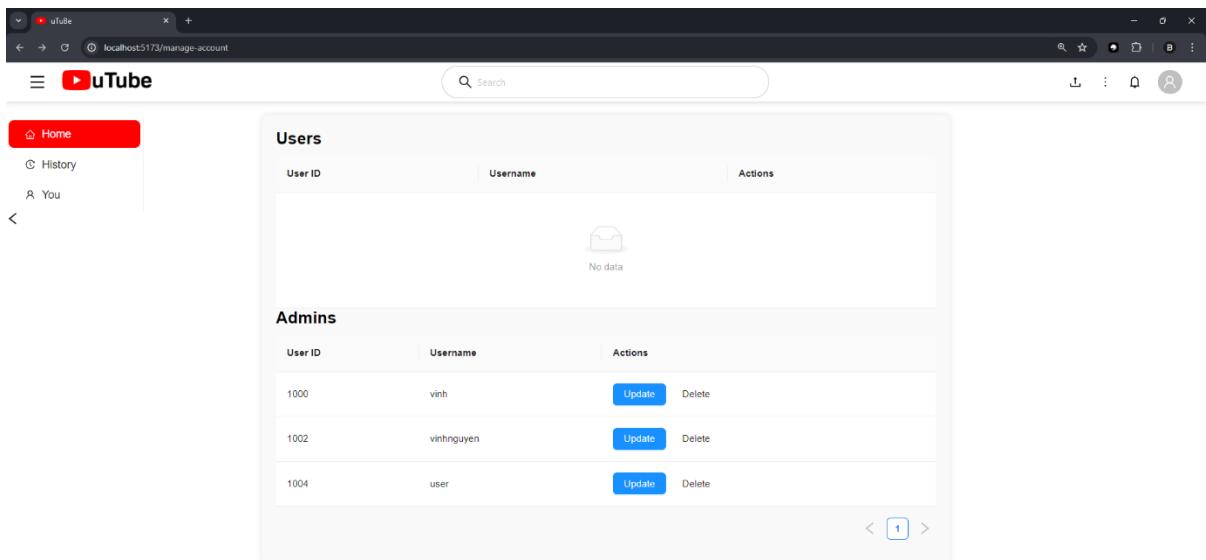
User ID	Username	Actions
1000	vinh	Update Delete
1002	vinhnguyen	Update Delete

Below the "Admins" table is a page navigation indicator: < [1] >.

- Now, I can update users to be an administrator by clicking “Update” button:

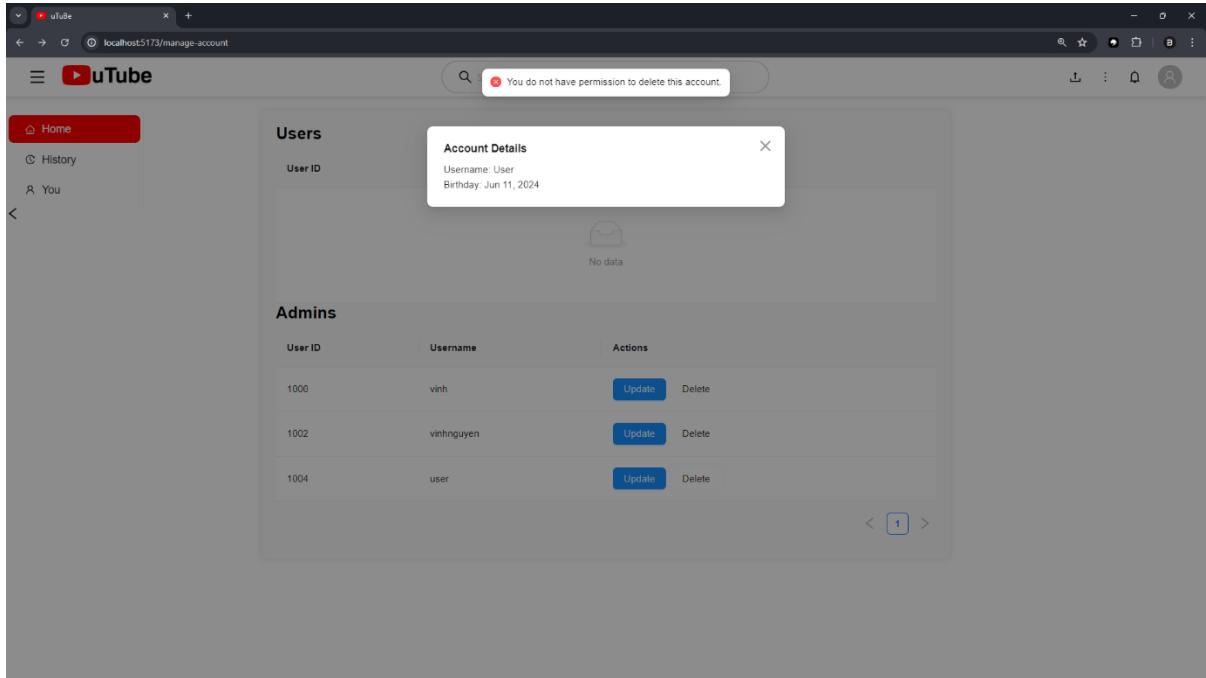


- Then, the list will be updated:



4.2. Delete users:

- Click “Delete” button to delete this administration:



- But the current account is administration, so it just can delete another user, deleting “another”:

User ID	Username	Actions	
1005	another	<button>Update</button>	<button>Delete</button>

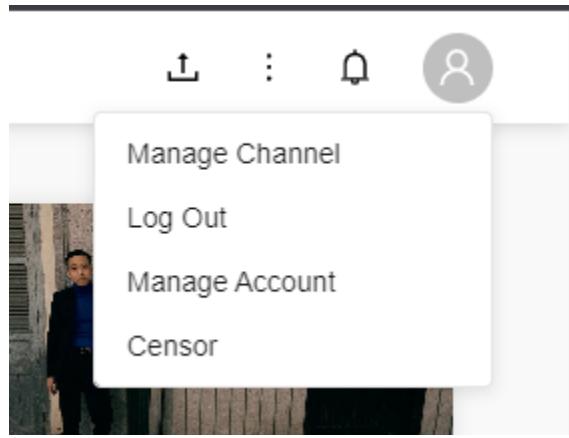
User ID	Username	Actions	
1000	vinh	<button>Update</button>	<button>Delete</button>
1002	vinhnguyen	<button>Update</button>	<button>Delete</button>
1004	user	<button>Update</button>	<button>Delete</button>

Account deleted successfully.

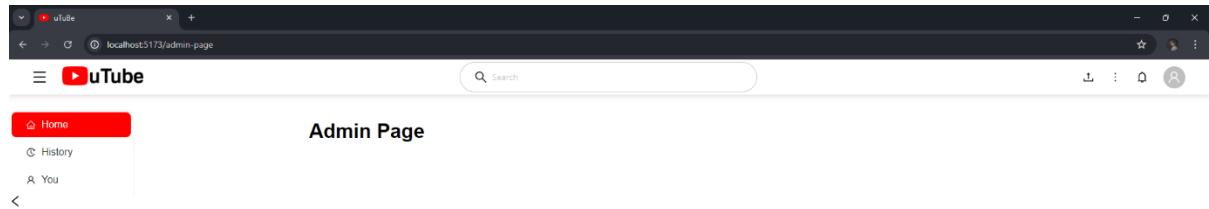
User ID	Username	Actions	
1000	vinh	<button>Update</button>	<button>Delete</button>
1002	vinhnguyen	<button>Update</button>	<button>Delete</button>
1004	user	<button>Update</button>	<button>Delete</button>

4.3. Review upload videos:

- Click “Censor” below the account icon:



- The list of newest upload videos will be shown. But right now, there is not upload videos in this page:



- I have uploaded new video, so the list will be updated:

Admin Page

Dù Con Đã Khôn Lớn - Yunobigboi x Tống Dài | OST Lát Mát 7 (Official Music Video)

Description: #yunobigboi #ostlata7 #tongdai #yunobigboi #lyanckhoa

Music Producer: Tống Dài Production Management: Tống Dài Production

Date: 6/17/2024, 4:34:51 AM

Owner: Vinh Nguyen

Status: Pending

Accept Delete

- I can accept this video by clicking on “Accept” button, otherwise I can deny this video by clicking on “Delete” button.
- After video has been accepted. It will appear on home page:

Home History You

SƠN TÙNG M-TP | CHỨNG TÀ CỦA TƯỞNG LẠI | OFFICIAL MUSIC VIDEO
by Kewtie

Den - Nấu Ăn Cho Em ft. PiaLinh (MV)
Vinh Nguyen
6/17/2024

SON TUNG M-TP | MAKING MY WAY | OFFICIAL VISUAL...
Vinh Nguyen
6/17/2024

Madhu - Không Còn Em | Official Music Video
Vinh Nguyen
6/17/2024

HIEUTHUHAI - Không Thể Say (prod. by Kewtie) I...
Vinh Nguyen
6/17/2024

CHẠY NGAY ĐI | RUN NOW | SON TUNG M-TP | Official Music Video
Vinh Nguyen
6/17/2024

SON TUNG M-TP | DỪNG LÂM TRẠI | TÌM ANH ĐAU | OFFICIAL MUSIC VIDEO
Vinh Nguyen
6/17/2024

STREAM ĐÊN BAO GIỜ - ĐỖ MIXI ft. BẢN SANG TAC |...
Vinh Nguyen
6/17/2024

Travis Scott - FEIN ft. Playboi Carti

Den - Lời Nhớ ft. Phương Anh Dao (MV)

SON TUNG M-TP | HAY TRAO CHO ANH ft. Snoop Dogg...

HIEUTHUHAI - Vé Tình ft. Hoàng Tân (prod. by Ke...)

5. Root role

- This role has similar functions compared to Admin role.
- But it can delete other administrative:

User ID	Username	Actions
1002	vinhnguyen	<button>Update</button> <button>Delete</button>
1006	user	<button>Update</button> <button>Delete</button>

-
- I will delete user id “1006”:

User ID	Username	Actions
1002	vinhnguyen	<button>Update</button> <button>Delete</button>

IV. DISCUSSION AND CONCLUSION

Developing a video platform web application presented significant challenges, particularly in handling video-type data and implementing a time-tracking save history function. One of the primary difficulties was efficiently storing and retrieving large video files. React Vite and the Ant Design (antd) library provided a solid foundation for the frontend, ensuring a responsive and intuitive user interface. Moreover, the backend, built with Servlets, required meticulous optimization to manage video data. This involved configuring Docker containers for scalability and utilizing AWS services for reliable storage and streaming. Implementing the save history function to track users' watch time added another layer of complexity, as it demanded precise synchronization between the frontend and backend to accurately record and update viewing progress.

This project was a comprehensive learning experience, enhancing our understanding of full-stack web development and modern deployment techniques. We gained practical knowledge in managing video data, optimizing server performance, and ensuring seamless user interactions. The use of Docker and AWS not only improved our deployment skills but also taught us how to leverage cloud services for scalable applications. Moreover, tackling the time-tracking save history function reinforced the importance of synchronization and data integrity in web applications. Overall, this project strengthened our technical skills and prepared us for future challenges in web development and cloud computing.

V. References:

1. Front End
 - <https://vitejs.dev/guide/>
 - <https://ant.design/>
 - <https://www.w3schools.com/>
 - <https://github.com/ant-design/ant-design-pro>
2. Back End
 - <https://docs.oracle.com/javaee/7/tutorial/servlets.htm>
 - <https://dev.mysql.com/doc/>
 - <https://www.mysqltutorial.org/>
 - <https://docs.docker.com/>
 - <https://docs.aws.amazon.com/>
 - <https://docs.docker.com/samples/mysql/>
 - <https://www.javatpoint.com/example-to-connect-to-the-mysql-database>