

# **NEPTUBE**

## **Mid-Term Report on Video Streaming Web App with ML-Powered Recommendation and AI Integration**



### **Submitted to**

Department of Computer Science and Engineering  
Nepal Engineering College

in Partial Fulfillment of the  
Requirements for the Degree of **B.E. in Computer Engineering**

### **Submitted By**

Aayush Koirala	(021-304)
Prabesh Basnet	(021-355)
Kishor Upadhyaya	(021-332)

### **Supervised By**

Asst.Prof.Bishwadeep Mainaly

**December 2025**

## Abstract

Our project focuses on developing a web app video streaming application that utilizes machine learning to enhance user experience through personalized content recommendations. The system leverages both collaborative filtering and content-based filtering to analyze user preferences, watch history, and interaction patterns, enabling intelligent video recommendations tailored to individual viewing habits.

One of the major features of the application is its differentiated access model, where a user can both create and upload the video as well as view the videos uploaded by other users. This allows user to be the content creator as well as a viewer.

With Artificial intelligence integration for the creator, it helps to generate the thumbnails, video description as well as eye catching titles.

Also, with the help of Machine Learning the recommendation system continuously adapts to user behavior, ensuring the viewer tailored experience and adapts to the user liking and history which brings viewer friendliness.

**Keywords:** Web App, Video Platform, Machine Learning, Smart Recommendations, User Behavior, AI Features, Creator Tools, Viewer Experience.

## Acknowledgement

We'd like to take a moment to thank everyone who's supported us as we get ready to start this project. Even though we haven't begun the actual work yet, the encouragement and guidance we've received so far mean a lot and have helped us shape our ideas and goals.

A big thank you to the **Department of Computer Science and Engineering** at **Nepal Engineering College** for creating an environment that motivates us to think creatively and take on new challenges. Your support gives us the confidence to move forward.

We're also really grateful to our **teachers**, whose advice and feedback during the early stages have helped us figure out the right direction for our project.

And of course, thank you to our **families and friends** for always being there, cheering us on, and believing in us. Your support makes a big difference as we prepare to take this next step.

*Yours sincerely,*

**Aayush Koirala** (021-304)

**Prabesh Basnet** (021-355)

**Kishor Upadhyaya** (021-332)

*December 2025*

# Contents

<b>Abstract</b> . . . . .	<b>i</b>
<b>Acknowledgement</b> . . . . .	<b>ii</b>
<b>List of Figures</b> . . . . .	<b>iii</b>
<b>List of Tables</b> . . . . .	<b>iv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Introduction of the Project . . . . .	1
1.2 Problem Statement . . . . .	1
1.3 Objective . . . . .	2
1.4 Aim . . . . .	3
1.5 Motivation . . . . .	3
1.6 Scope . . . . .	4
1.7 Application . . . . .	5
1.8 Feasibility Study . . . . .	5
1.8.1 Technical Feasibility . . . . .	5
1.8.2 Economic Feasibility . . . . .	6
1.9 Evaluation of Existing Systems . . . . .	6
1.9.1 YouTube . . . . .	6
1.9.2 CineHub Nepal . . . . .	6
1.9.3 TikTok . . . . .	7
<b>2 Literature Review</b> . . . . .	<b>8</b>
2.1 Machine Learning in Video Recommendations . . . . .	8
2.2 Artificial Intelligence in Content Creation . . . . .	8
2.3 Community Engagement and Personalization . . . . .	9

2.4	Streaming and Content Platforms in Nepal . . . . .	9
<b>3</b>	<b>System Design . . . . .</b>	<b>10</b>
3.1	System Description . . . . .	10
3.2	Flowchart . . . . .	10
3.2.1	User's Flowchart . . . . .	10
3.2.2	Creator's Flowchart . . . . .	13
3.3	Use Case Diagram . . . . .	15
3.3.1	User's Use Case Diagram . . . . .	15
3.3.2	Creator's Use Case Diagram . . . . .	16
3.3.3	Admin's Use Case Diagram . . . . .	18
3.4	Sequence Diagram . . . . .	18
3.4.1	User's Sequence Diagram . . . . .	19
3.4.2	Admin's Sequence Diagram . . . . .	20
3.4.3	Creator's Sequence Diagram . . . . .	21
<b>4</b>	<b>Implementation and Discussion . . . . .</b>	<b>23</b>
4.1	Tasks Completed . . . . .	23
4.2	Tasks Remaining . . . . .	23
4.3	Test Cases . . . . .	23
4.4	Gantt Chart . . . . .	25
<b>5</b>	<b>Conclusion . . . . .</b>	<b>26</b>
	<b>References . . . . .</b>	<b>27</b>
	<b>Appendices . . . . .</b>	<b>29</b>
<b>A</b>	<b>Technology Stack . . . . .</b>	<b>30</b>
A.1	Frontend Technologies . . . . .	30
A.2	Backend Technologies . . . . .	31
A.3	AI/ML Technologies . . . . .	31

A.4	DevOps & Infrastructure . . . . .	32
<b>B</b>	<b>Database Schema . . . . .</b>	<b>33</b>
B.1	Core Tables . . . . .	33
B.2	Key Relationships . . . . .	33

## List of Figures

3.1	User's Flowchart . . . . .	12
3.2	Creator's Flowchart . . . . .	14
3.3	User's Use Case Diagram . . . . .	16
3.4	Creator's Use Case Diagram . . . . .	17
3.5	Admin's Use Case Diagram . . . . .	18
3.6	User's Sequence Diagram . . . . .	20
3.7	Admin's Sequence Diagram . . . . .	21
3.8	Creator's Sequence Diagram . . . . .	22

## List of Tables

4.1	Test Cases for Core Functionalities . . . . .	24
4.2	Project Development Timeline (Gantt Chart) . . . . .	25
A.1	Frontend Technology Stack . . . . .	30
A.2	Backend Technology Stack . . . . .	31
A.3	AI/ML Technology Stack . . . . .	31
A.4	DevOps and Infrastructure Stack . . . . .	32
B.1	Database Tables Overview . . . . .	33



# Chapter 1

## Introduction

### 1.1 Introduction of the Project

In today's content-rich digital world, discovering videos that match user interests can often feel overwhelming. To tackle this challenge, we introduce **Neptune**, a web-based video streaming platform designed to deliver a personalized and engaging viewing experience through the power of artificial intelligence and machine learning.

At the heart of Neptune lies an intelligent recommendation system powered by machine learning algorithms, including collaborative filtering and content-based filtering. These systems analyze user preferences, watch history, and interaction patterns to generate highly relevant video suggestions that adapt over time.

Going beyond recommendations, Neptune also integrates AI-powered tools for content creators, including automatic thumbnail generation, video title suggestions, and description drafting [1], helping creators make their content more appealing and discoverable with minimal effort.

While the platform draws inspiration from larger services like YouTube, Neptune focuses on a simpler, more focused user experience. Core features include the ability to create playlists, like or dislike videos, comment, and subscribe to favorite creators—fostering community and interaction.

By combining adaptive machine learning with creator-focused AI tools and essential social features, Neptune aims to create a user-friendly streaming environment that supports both content discovery and creation.

### 1.2 Problem Statement

In Nepal, the growing demand for digital video content faces several challenges that limit the overall user experience and content creator opportunities on streaming platforms. Despite improvements in internet access and usage, the following issues remain prevalent:

- **Heavy Reliance on Foreign Platforms:** Most Nepali users depend on international streaming services, which often do not prioritize local languages, culture, or content preferences.
- **Lack of Localized Video Streaming Platforms:** There is a shortage of streaming services designed specifically for the Nepali market, which limits access to culturally relevant content and fails to address local user needs.
- **Content Discovery Overload:** Users in Nepal often feel overwhelmed by the vast number of available videos, making it difficult to find content that truly matches their interests.
- **Monetization Challenges:** Content creators face difficulties in monetizing their work effectively, impacting the sustainability of local content production.

This project aims to develop **Neptune**, a web-based video streaming platform designed specifically for the Nepali audience [2]. Leveraging machine learning, the platform will provide personalized video recommendations based on users' viewing habits and preferences. Alongside an intuitive user experience, Neptune will offer essential social features such as playlists, comments, likes/dislikes, and subscriptions to foster community engagement. Additionally, AI-powered tools [3] will assist content creators by automatically generating video thumbnails, titles, and descriptions to enhance content visibility. By focusing on localization, scalability, and creator support, Neptune seeks to improve content discovery, promote Nepali creators, and contribute to the growth of Nepal's digital entertainment landscape.

### 1.3 Objective

Following are the main objectives of our project:

- Create a full-fledged online streaming platform where users will be able to smoothly watch videos, choose between different quality levels, and enjoy stable playback even on slower networks.
- Build a system that lets creators upload their content easily, manage their profiles, and customize things like thumbnails, titles, and video details with AI helpers [4].
- Add features that bring life to the platform, like playlists, comments, likes, and channel subscriptions.

- Research popular recommendation techniques [5] including collaborative filtering and content-based filtering, drawing comparisons between how the two work before implementing which fits best with our platform.
- Integrate ad content naturally inside the video player.
- It should be designed for security and scalability to accommodate more users, more videos, and higher traffic.
- Let people trust the platform, by keeping their user data safe with good security practices.

## 1.4 Aim

The aim of this project is to develop **Neptune**, a scalable and user-friendly web-based video streaming platform specifically designed for the Nepali audience. The platform will leverage advanced machine learning [1] [2] techniques to provide personalized video recommendations that reflect individual user preferences and local cultural relevance. Additionally, Neptune will integrate AI-powered tools to assist Nepali content creators in enhancing their videos through automatic thumbnail generation, title suggestions, and description drafting. By incorporating interactive features such as playlists, comments, likes/dislikes, and subscriptions, the platform aims to foster a vibrant online community that encourages user engagement and supports local creators. Overall, this project seeks to bridge the gap in the digital entertainment ecosystem in Nepal by offering a localized, intelligent, and engaging video streaming experience tailored to the unique needs of Nepali viewers and creators contributing in the entertainment sector ecosystem.

## 1.5 Motivation

Nepal's digital landscape is evolving rapidly, with increasing internet accessibility playing a pivotal role in transforming content consumption habits. However, most Nepali users currently rely on international video streaming platforms that often overlook local cultural contexts, linguistic diversity, and the unique preferences of Nepal's audience. This gap limits the availability of relevant, engaging, and culturally resonant content tailored specifically for Nepali viewers.

Simultaneously, local content creators face significant challenges in expanding their reach and enhancing their content's appeal due to the lack of advanced, user-friendly tools. The absence of

AI-driven features for content optimization and personalized discovery constrains their growth and limits user engagement.

Driven by these challenges, Neptube aims to develop a web-based, localized video streaming platform that leverages cutting-edge machine learning techniques to deliver personalized recommendations and AI-powered creator tools [6]. This project is motivated by the vision to empower Nepali creators, enhance content discovery, and foster a vibrant, interactive community. Neptube aspires to bridge the gap between advanced technology and local cultural relevance, thereby enriching Nepal's digital entertainment ecosystem and offering a truly tailored viewing experience.

## 1.6 Scope

The scope of this project encompasses the following key areas:

- **Development of a Web-Based Platform:** Build a responsive and user-friendly video streaming web application accessible via standard web browsers.
- **User Interaction Features:** Implement core functionalities such as video upload, viewing, commenting, liking/disliking, playlist creation, and channel subscription.
- **Personalized Recommendation System:** Integrate machine learning algorithms (collaborative and content-based filtering) to provide customized video suggestions based on user activity and preferences.
- **AI-Powered Creator Tools:** Provide content creators with artificial intelligence tools for automatic thumbnail generation, title suggestions, and video descriptions.
- **Support for Local Content:** Focus on promoting Nepali videos and regional content to support local creators and cater to the cultural preferences of Nepali users.
- **User Authentication and Access Control:** Enable secure sign-up, login, and role-based access (viewer/creator) to manage content uploads and interactions.
- **Scalable Architecture:** Design the system to handle increasing numbers of users and growing video content efficiently over time.
- **Community Engagement:** Encourage active participation through comment sections, feedback mechanisms, and content subscriptions.

- **Cross-Platform Accessibility:** Ensure the platform is optimized for use across desktops, laptops, and other devices with internet browsers, without reliance on mobile-specific apps.

## 1.7 Application

The applications for our project are:

- **Video Streaming Platform:** Allows users to watch, upload, and share video content through a centralized web application.
- **Content Recommendation:** Provides personalized video suggestions using machine learning to enhance user engagement and retention.
- **Digital Content Creation Support:** Offers AI-based tools to assist creators in generating thumbnails, titles, and descriptions for their videos.
- **Educational Use:** Can be adapted for sharing educational videos, tutorials, and online learning content for students and teachers.
- **Community Engagement:** Enables user interaction through comments, likes, and subscriptions, fostering a sense of online community.
- **Marketing and Branding:** Businesses and creators can use the platform to reach target audiences by uploading promotional and informational videos.
- **Entertainment Hub:** Serves as a go-to destination for users seeking diverse video content, from music and vlogs to documentaries and entertainment clips.

## 1.8 Feasibility Study

### 1.8.1 Technical Feasibility

The development of Neptune is technically feasible using currently available and accessible web technologies. The platform will be built using common web development tools such as HTML, CSS, JavaScript, and backend technologies like Node.js or Python [7] [8]. Machine learning models for video recommendation can be implemented using popular open-source libraries such as Scikit-learn or TensorFlow. AI features for generating thumbnails, titles, and descriptions can be added using pre-trained models or APIs. Since the project is web-based, it can be accessed from any device with a browser, eliminating the need for mobile app development. The technical

requirements are within the skill set of the development team, making implementation realistic and manageable.

### **1.8.2 Economic Feasibility**

The development of Neptube is economically practical, especially within the scope of a student or academic project. Most of the tools and technologies required—such as web development frameworks, machine learning libraries, and database systems—are freely available as open-source resources. Initial costs such as domain registration and basic cloud hosting can be managed using free or low-cost service tiers. Since the project does not require expensive hardware, paid APIs, or large-scale infrastructure during its initial phase, the overall financial requirements are minimal. This makes the project viable within limited budgets, while still allowing for future expansion if the platform gains traction.

## **1.9 Evaluation of Existing Systems**

### **1.9.1 YouTube**

#### **Strengths:**

- Free access with ads, making it widely accessible.
- Supports a broad range of content from all over the world, including user-generated content.
- Low-bandwidth streaming options (from 144p to 1080p), ensuring accessibility in regions with internet speed challenges [9] [21].

#### **Weaknesses:**

- The ad-supported model can disrupt the viewing experience, especially on free-tier content.
- Difficult for growing for creators due to large and international scale of the website.
- Payment to creators is difficult in the context of Nepal.

### **1.9.2 CineHub Nepal**

#### **Strengths:**

- Focuses on Nepali indie films and content from local filmmakers.

**Weaknesses:**

- No advanced recommendation system or personalized content discovery features.
- Only for movies, Creators can't make their own content and upload it.

**1.9.3 TikTok****Strengths:**

- Focuses on promoting small business and small creators.
- No ads so great viewing experience.
- Good recommendation system.

**Weaknesses:**

- No long form video upload feature.
- Videos are always on portrait format.

Existing platforms like TikTok and YouTube have undoubtedly set the stage for streaming services, but there is room for innovation in areas such as local content, personalized recommendations. Neptune aims to address these opportunities by integrating machine learning for personalized content delivery and offering both content creation and viewing experience. This approach ensures a more inclusive and user-centric streaming experience.

## **Chapter 2**

### **Literature Review**

The evolution of video streaming platforms has been profoundly shaped by advancements in machine learning and artificial intelligence. These technologies enable personalized content recommendations and intelligent content creation, enhancing both user engagement and creator efficiency. As global platforms continue to innovate, there is a growing opportunity to localize these technologies for underserved markets like Nepal. This literature review explores key developments in recommendation systems, AI-assisted content tools, user engagement strategies, and the current state of digital streaming services relevant to the Nepali context.

#### **2.1 Machine Learning in Video Recommendations**

- Netflix reports 80% of its content views are driven by its recommendation system [10] [11].
- Collaborative filtering and content-based filtering are widely used for tailoring content suggestions [12].
- Hybrid models and session-based neural networks improve personalization in real-time.
- Dynamic representation learning adapts to changing viewer preferences.
- Such techniques are essential for solving cold-start and over-saturation problems in localized platforms.

#### **2.2 Artificial Intelligence in Content Creation**

- AI tools are used to generate thumbnails, video titles, and descriptions [13].
- YouTube and other platforms employ AI to assist creators with trend-based suggestions [14].
- AI improves accessibility for creators with fewer technical skills, which is vital in developing regions like Nepal.



## **2.3 Community Engagement and Personalization**

- User interactions (likes, comments, playlists) enhance platform stickiness.
- Personalized UI/UX improves satisfaction and drives retention [9].

## **2.4 Streaming and Content Platforms in Nepal**

- Most Nepali streaming services (e.g., NET TV, NepalFlix) lack ML-based recommendations [15] [16].
- Over-reliance on mobile apps leads to exclusion of desktop/laptop users.
- Limited monetization tools for Nepali creators restrict local content growth.
- There is a strong demand for culturally relevant, personalized content in the Nepali language.

## **Chapter 3**

### **System Design**

#### **3.1 System Description**

Neptune is a web-based video streaming application designed to provide an engaging platform for both content creators and viewers. Users can upload videos, create playlists, and manage their channels, while viewers can watch videos, comment, like or dislike, and subscribe to channels they enjoy. The platform integrates machine learning techniques, such as collaborative and content-based filtering, to deliver personalized video recommendations based on individual user preferences and viewing history.

Additionally, Neptune includes AI-powered tools that assist content creators by automatically generating video thumbnails, titles, and descriptions, helping improve content visibility and user engagement. The system is built with scalability and performance in mind, ensuring it can support a growing user base and content library. Focused on promoting Nepali and regional content, Neptune aims to offer a tailored and user-friendly experience for its audience.

#### **3.2 Flowchart**

An illustration of the logical flow and order of actions or steps in a system or process is called a flowchart. It represents numerous kinds of decisions, actions, inputs, outputs, and connectors using a variety of shapes and symbols. Workflows may be analyzed, communicated, and improved more easily when complicated processes are illustrated visually and simply via flowcharts. They are extensively used to visually map out processes and identify bottlenecks or areas for improvement in a variety of sectors, including software development, project management, business analysis, and problem-solving.

##### **3.2.1 User's Flowchart**

The user's flowchart provides a clear and professional overview of how a user interacts with a video-sharing platform. It begins with the user registering or logging into the system, followed by browsing and searching for videos. Once a video is found, the user decides whether to add

it to a playlist. If not, they evaluate whether they prefer the video. Based on their preference, the user may like or dislike the video. If they dislike it, they may choose to watch it later. If they like the video, they can choose to leave a comment. After commenting or not, the user is prompted to subscribe to the creator. Regardless of whether they subscribe, the user continues watching and receives recommendations for similar videos before the flow ends. This flowchart represents a typical interactive experience of a user engaging with content on the platform.

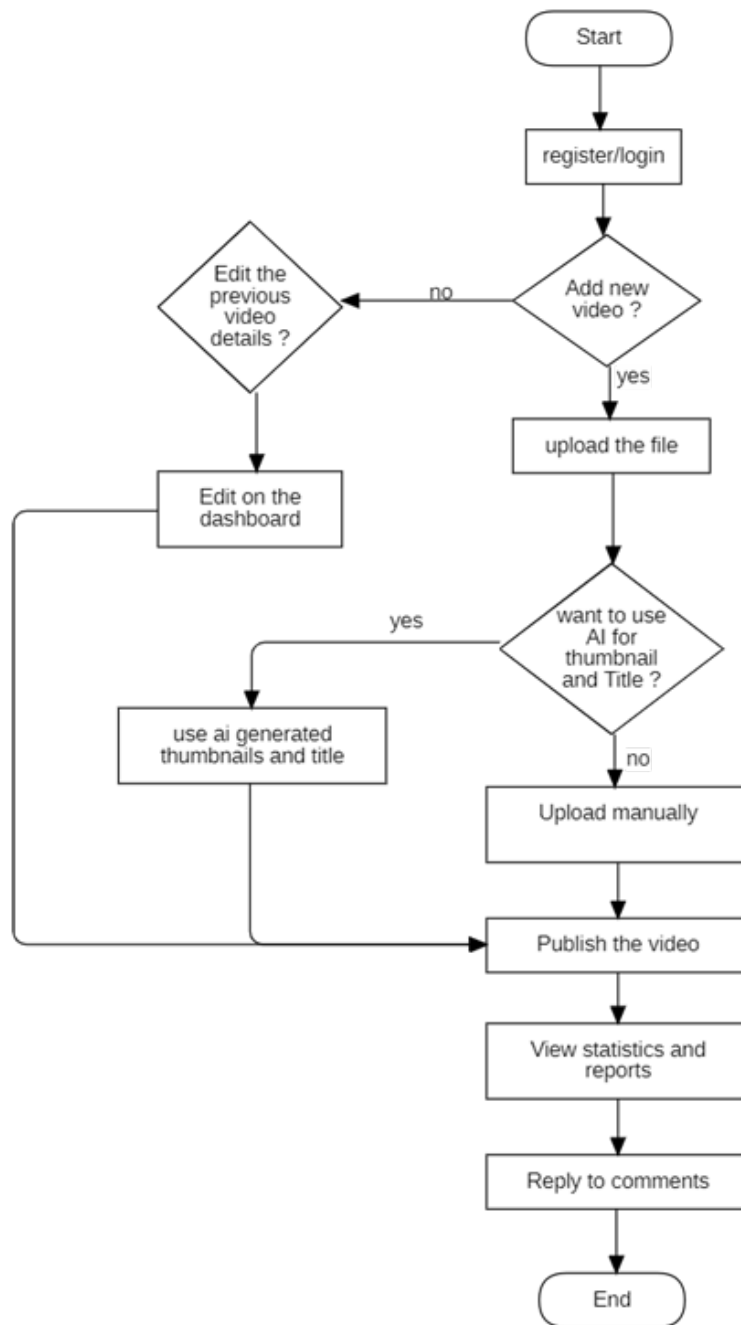


### **3.2.2 Creator's Flowchart**

The creator's flowchart provides a clear and organized overview of the steps a content creator follows on a video-sharing platform. It begins with the creator registering or logging into their account. After logging in, the creator can choose to either edit the details of a previously uploaded video or upload a new one. If they choose to edit, they are directed to the dashboard to make changes to existing content.

If the creator wants to upload a new video, they start by uploading the video file. Then, they are given the option to use AI-generated thumbnails and titles for the video or to enter them manually. This provides flexibility based on the creator's preference or need for convenience. Once the thumbnail and title are set—either through AI or manual input—the video is ready to be published.

After publishing, the creator can access detailed statistics and reports to track the performance of the video, such as views, likes, and comments. Finally, the creator can engage with the audience by replying to comments, which helps build a stronger connection with viewers. This flowchart effectively summarizes the content creation and management process, from uploading and editing to analyzing performance and engaging with the audience.



**Figure 3.2:** Creator's Flowchart

### **3.3 Use Case Diagram**

Use case diagrams are a simple and effective way to show how different people interact with a system. They help us understand what the system is supposed to do by focusing on the users and their actions. Instead of getting into the technical details, these diagrams give a clear picture of the main features and how each user—like a customer, service provider, or admin—connects with them. This makes it easier for everyone involved, from developers to stakeholders, to stay on the same page and ensure the system meets real user needs.

#### **3.3.1 User's Use Case Diagram**

The use case diagram represents the interaction between a Viewer and a video streaming platform, outlining key functionalities for consuming and engaging with content. The Viewer begins by Logging in to access personalized features and can then Search or Browse Videos to discover content. Upon selecting a video, they can Watch it, with extended options to Like or Dislike and Comment, promoting user interaction. Viewers also have the ability to Delete their Comments, Subscribe to Channels for updates, and Add or Remove Videos from Playlists for easier organization. Additionally, the platform offers Recommendations based on viewing behavior. The session concludes with the Logout function. This diagram efficiently captures the essential actions a Viewer can take to interact with and personalize their experience on the platform.



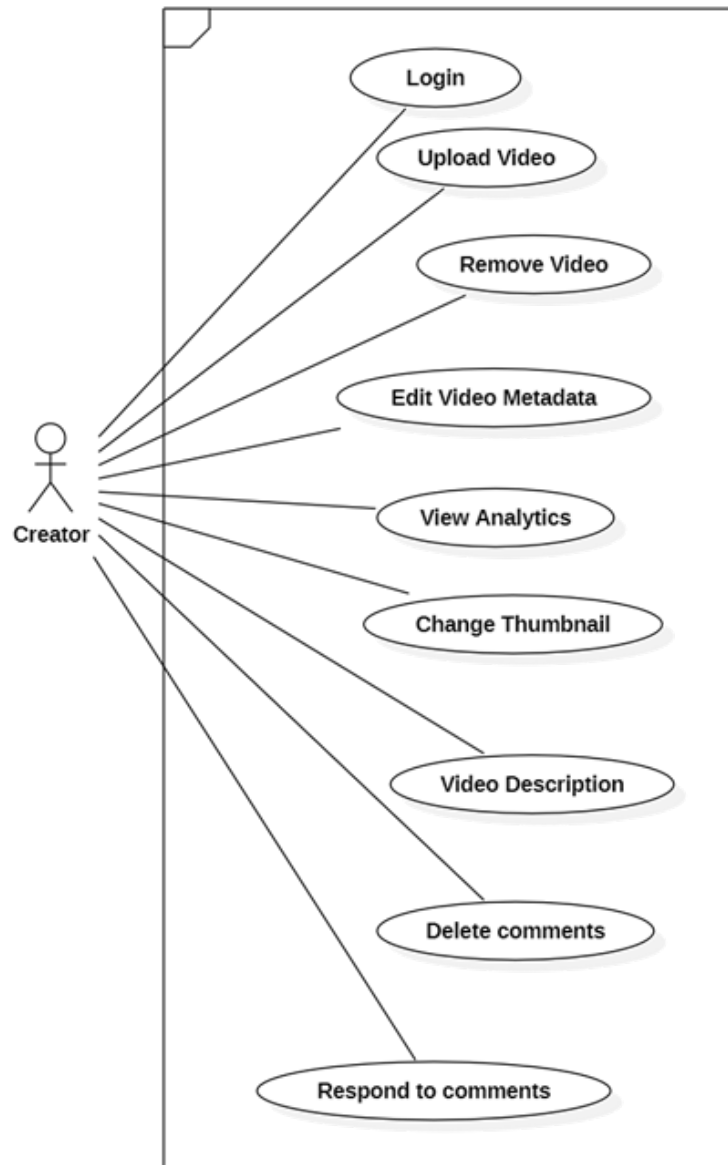
**Figure 3.3:** User's Use Case Diagram

### 3.3.2 Creator's Use Case Diagram

The use case diagram illustrates the interactions between a Creator and a video content platform, outlining the core functionalities available for managing video content. The Creator begins by performing a Login, enabling access to their account and content management tools. They can then Upload Video to share new content with viewers and Remove Video to delete existing content when necessary. The Edit Video Metadata use case allows creators to modify details like titles, tags, and categories, while Change Thumbnail and Video Description support enhancing the visual and textual representation of the video. View Analytics enables creators to monitor video performance and audience engagement through data insights. In terms of community interaction, the Creator can Delete Comments that are inappropriate or irrelevant and Respond to

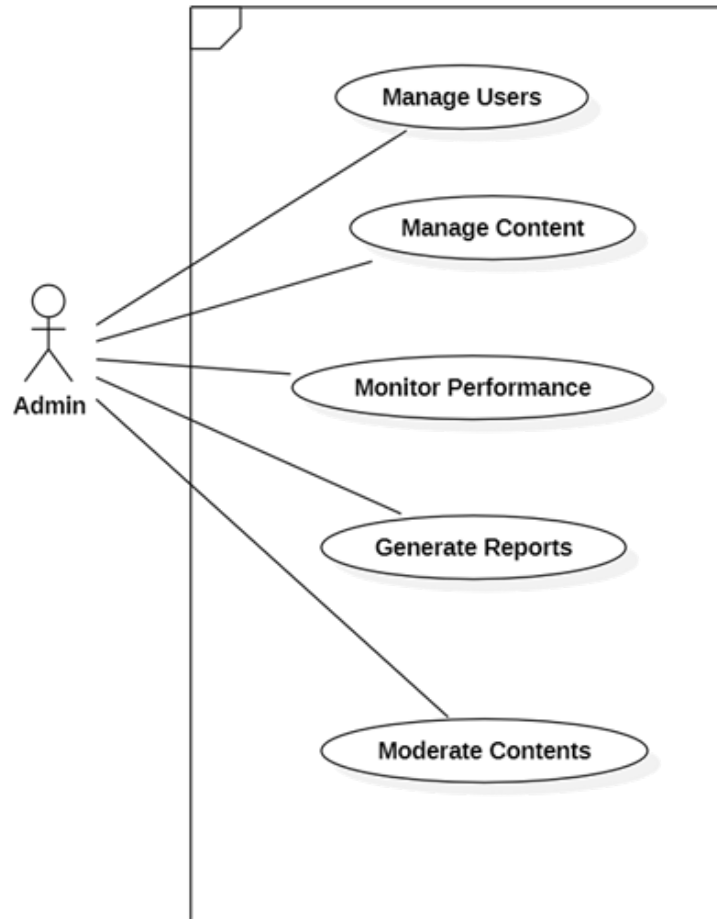


Comments to engage directly with viewers. This diagram effectively summarizes the Creator's essential tools for content management, audience engagement, and performance tracking on the platform.



**Figure 3.4:** Creator's Use Case Diagram

### 3.3.3 Admin's Use Case Diagram



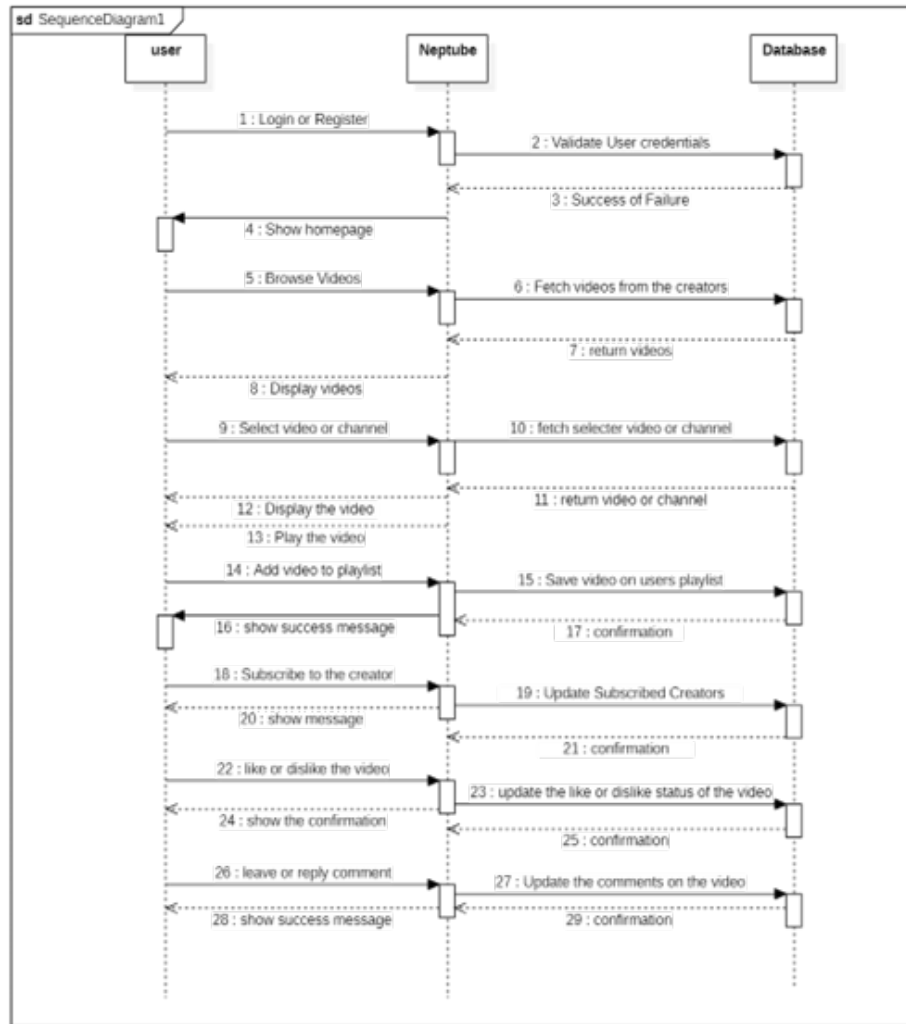
**Figure 3.5:** Admin's Use Case Diagram

## 3.4 Sequence Diagram

Sequence diagrams help us visualize how different parts of a system communicate with each other over time. They show the step-by-step flow of messages between users and the system or between different parts of the system itself. This makes it easier to understand the exact order in which things happen—like when a user books a service and how the system responds. By breaking down the process into a timeline of interactions, sequence diagrams help developers plan, build, and troubleshoot the system more effectively.

### **3.4.1 User's Sequence Diagram**

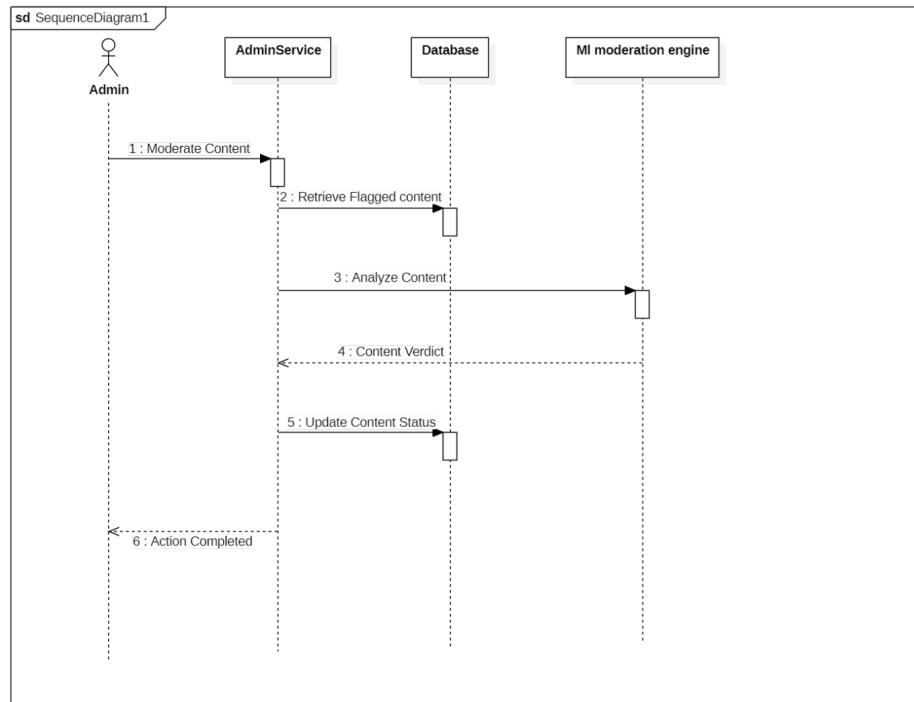
The user's sequence diagram illustrates the interaction between the user, the Neptune platform, and the database during various actions. It begins with the user logging in or registering, prompting Neptune to validate credentials with the database and return a success or failure message. Upon successful login, the homepage is shown, and the user browses videos. Neptune fetches the videos from creators, retrieves them from the database, and displays them to the user. When a video or channel is selected, Neptune retrieves the corresponding data and displays the video for playback. If the user adds the video to a playlist, it is saved in the database, confirmed, and a success message is shown. Subscribing to a creator updates the subscription list in the database, followed by a confirmation and display of a message. Liking or disliking a video updates the status in the database and shows a confirmation. Leaving or replying to a comment updates the comment section, confirms the update, and concludes with a success message.



**Figure 3.6:** User's Sequence Diagram

### 3.4.2 Admin's Sequence Diagram

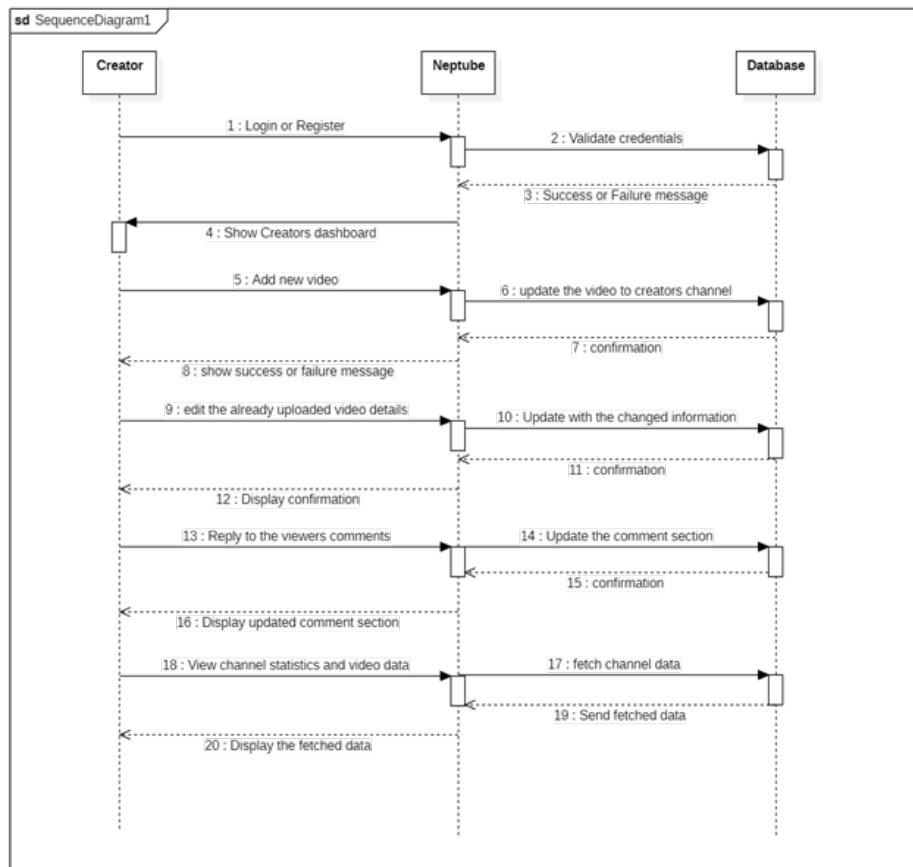
The admin's sequence diagram illustrates how flagged content is moderated through interaction among the Admin, AdminService, Database, and the AI moderation engine. The process starts when the Admin initiates content moderation. The AdminService retrieves the flagged content from the database, then sends it to the AI moderation engine for analysis. After analyzing the content, the engine returns a verdict to the AdminService. Based on this verdict, the AdminService updates the content status in the database. Once the update is complete, the system notifies the Admin that the action has been completed.



**Figure 3.7:** Admin’s Sequence Diagram

### 3.4.3 Creator’s Sequence Diagram

The creator’s sequence diagram illustrates the interactions between a content creator, the Neptune platform, and its database, focusing on video management, viewer engagement, and analytics. The process begins when the creator logs in or registers, prompting Neptune to validate credentials through the database and return a success or failure message. Upon successful login, the system displays the creator’s dashboard. The creator can then add a new video, which is sent to Neptune and updated in the database. A confirmation is returned, followed by a success or failure message shown to the creator. If needed, the creator can edit the details of an already uploaded video, with the changes being updated in the database and confirmed back to the creator. Next, the creator can engage with viewers by replying to comments; these replies are updated in the comment section and confirmed by the database, leading to the display of the updated comment section. Finally, the creator may choose to view analytics such as channel statistics and video performance data. This request prompts Neptune to fetch the relevant data from the database, and once retrieved, the data is displayed to the creator. This entire sequence ensures seamless content management and audience interaction for creators on the platform.



**Figure 3.8:** Creator's Sequence Diagram

## **Chapter 4**

### **Implementation and Discussion**

#### **4.1 Tasks Completed**

The following tasks have been successfully completed during the initial phase of development:

- User sign-in, sign-up, and secure protected routes were added.
- Database was set up using Neon PostgreSQL and Drizzle ORM.
- Tables for users, videos, comments, likes, and subscriptions were created.
- tRPC API was built for user, video, comment, and subscription features.
- Admin panel for stats, user roles, user ban features was completed.
- UI layout with navbar and sidebar was designed.
- Responsive design and shadcn/ui components were fully integrated.

#### **4.2 Tasks Remaining**

- Machine learning implementation for personalized recommendations.
- Video interaction features (likes, comments, replies, subscriptions) enhancement.
- AI-based content generation for video titles, descriptions, and thumbnails.

#### **4.3 Test Cases**

For implementation of our project, we have done with some special tests such as white box testing and black box testing. White box testing is testing of the internal working or code of software application. Black box testing is system testing from perspective of user or we can say external working.

**Table 4.1:** Test Cases for Core Functionalities

ID	Test Case	Test Steps	Test Data	Expected Result	Actual Result	Status
1	Check user login with valid data	Navigate to /sign-in and confirm the SignIn form	Email=test1@gmail.co password=test123	User should be able to sign in to the system	As expected	Pass
2	Check user registration with valid data	Navigate to /sign-up and confirm the SignUp form	Email=test2@gmail.co Password=test123	User should be able to sign up to the system	As expected	Pass
3	Sign-out functionality	Log in, open the avatar menu, and click Sign Out	Email=test1@gmail.co Logged-in user	User logs out and returns to home	As expected	Pass
4	Database connection	Check src/db/index.ts for Drizzle + Neon setup	DATABASE_URL env	DB instance connects and exports	As expected	Pass
5	Admin dashboard page	Log in as admin and open /admin	Admin user	Dashboard with stats displayed	As expected	Pass
6	Admin users page	Open /admin/users and confirm the user list shows	Admin user	User management table displayed	As expected	Pass
7	Check user ban functionality	Open admin dashboard, navigate to user tab and select the user to ban	No data required	Selected user should be banned	As expected	Pass



## 4.4 Gantt Chart

**Table 4.2:** Project Development Timeline (Gantt Chart)

Project Phase	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10
Pre Analysis Phase	✓									
Project Proposal		✓								
Detailed Study & Analysis			✓							
Initial Prototype				✓	✓					
Implementation Phase I						✓	✓			
Implementation Phase II								○	○	
Testing & Documentation										○

**Legend:** ✓ Completed    ○ Planned

## **Chapter 5**

### **Conclusion**

The Neptune project aims to create a user-centric web-based video streaming platform that combines modern web technologies with machine learning and artificial intelligence to enhance the overall user experience. By enabling both content creation and consumption, the platform fosters an interactive community where users can engage through comments, likes, and subscriptions. The integration of personalized recommendations ensures that viewers receive content tailored to their preferences, improving content discovery and user satisfaction.

Moreover, Neptune's focus on promoting Nepali and regional content addresses the need for localized digital entertainment, providing a platform that supports and encourages local creators. With scalable architecture and AI-powered tools, the system is designed to be efficient, adaptable, and easy to use.

In summary, Neptune holds the potential to enrich Nepal's digital entertainment landscape by offering a modern, engaging, and culturally relevant video streaming experience.

## Bibliography

- [1] D. Dugan *et al.*, “Generative AI in YouTube Content Creation,” *arXiv*, 2025. [Online]. Available: <https://arxiv.org/html/2503.03134v1>
- [2] DaamiDeal, “OTT Platforms in Nepal,” 2025. [Online]. Available: <https://www.daamideal.com/blogs/popular-ott-platforms-for-online-movies-in-nepal>
- [3] OpenAI, GPT-4 [Large language model], 2023. [Online]. Available: <https://openai.com/product/gpt-4>
- [4] GoogleCloudPlatform, “Generative-AI: Sample Code and Notebooks for Generative AI on Google Cloud, with Gemini on Vertex AI.” [Online]. Available: <https://github.com/GoogleCloudPlatform/generative-ai>
- [5] Twitter, “The-Algorithm-ML: Source Code for Twitter’s Recommendation Algorithm.” [Online]. Available: <https://github.com/twitter/the-algorithm-ml>
- [6] Ingrade, “How YouTube Recommendation Works: A Deep Dive into AI, Deep Learning, and Collaborative Filtering,” 2025. [Online]. Available: <https://ingrade.io/how-youtube-recommendation-works-a-deep-dive-into-ai-deep-learning-and-collaborative-filtering/>
- [7] The PostgreSQL Global Development Group, “PostgreSQL Documentation,” 2025. [Online]. Available: <https://www.postgresql.org/docs/current/>
- [8] B. Joshi *et al.*, “Sentiment Analysis in Nepali using LLMs,” *ACM TALIP*, vol. 23, no. 2, 2025. [Online]. Available: <https://dl.acm.org/doi/fullHtml/10.1145/3647782.3647804>
- [9] FlowPlayer, “Adaptive Bitrate Streaming (ABR): The Ultimate Guide to ABR Video,” 2023. [Online]. Available: <https://flowplayer.com/blog/adaptive-bitrate-streaming>
- [10] S. Kharel, “Social Media Impact on Nepalese Teenagers,” *Theseus.fi*, 2024. [Online]. Available: [https://www.theseus.fi/bitstream/10024/815842/2/Kharel\\_Shreya.pdf](https://www.theseus.fi/bitstream/10024/815842/2/Kharel_Shreya.pdf)

- [11] J. M. Ph.D. and E. Kavlakoglu, “What Is Content-Based Filtering?,” *IBM*, 2025. [Online]. Available: <https://www.ibm.com/think/topics/content-based-filtering>
- [12] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*, 2nd ed. New York, NY, USA: Springer, 2015.
- [13] AlmaBetter, “How Netflix Uses ML & AI For Better Recommendation for Users,” 2023. [Online]. Available: <https://www.almabetter.com/bytes/articles/how-netflix-uses-machinelearning-to-keep-users-glued-to-the-screen>
- [14] P. Song, “How Does Netflix Use Machine Learning for Recommendations?,” *ML Journey*, 2025. [Online]. Available: <https://mljourney.com/how-does-netflix-use-machine-learning-for-recommendations/>
- [15] M. AI, “YouTube’s Machine Learning (ML) Algorithm,” *DEV Community*, 2022. [Online]. Available: [https://dev.to/mage\\_ai/youtubes-machine-learning-ml-algorithm-ej0](https://dev.to/mage_ai/youtubes-machine-learning-ml-algorithm-ej0)
- [16] Google Research, “Deep Neural Networks for YouTube Recommendations.” [Online]. Available: <https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf>
- [17] Elinext, “AI-Driven Personalization: YouTube, Netflix & Amazon,” 2025. [Online]. Available: <https://www.elinext.com/solutions/ai/trends/ai-driven-personalized-contentrecommendation/>
- [18] I. Chowdhury, “Impact of Artificial Intelligence on Content Creation,” *IJRPR*, vol. 5, no. 7, 2024. [Online]. Available: <https://ijrpr.com/uploads/V5ISSUE7/IJRPR31275.pdf>
- [19] A. Pujari and R. Patel, “Exploring the MERN Stack: An In-Depth Analysis of Full-Stack JavaScript Development for Scalable and Efficient Web Applications,” *JETIR*, vol. 10, no. 5, pp. 1–7, May 2023. [Online]. Available: <https://www.jetir.org/papers/JETIRTHE2177.pdf>
- [20] Vercel, *Next.js Documentation: Server-Side Rendering and Data Fetching*, 2024. [Online]. Available: <https://nextjs.org/>

`docs/pages/building-your-application/data-fetching/  
get-server-side-props`

- [21] M. G. Choi, H. J. Jo, and D. H. Kim, “A Real-Time Blind Quality-of-Experience Assessment Metric for HTTP Adaptive Streaming,” in *Proc. 2023 IEEE 14th Int. Conf. Mobility, Sensing and Networking (MSN)*, 2023, pp. 58–65.

# Appendix A

## Technology Stack

This appendix provides a comprehensive overview of the technologies, frameworks, and tools used in the development of the Neptune platform.

### A.1 Frontend Technologies

**Table A.1:** Frontend Technology Stack

Technology	Purpose
Next.js 14	React framework with server-side rendering and App Router
React 18	Component-based UI library for building interactive interfaces
TypeScript	Type-safe JavaScript for improved code quality and maintainability
Tailwind CSS	Utility-first CSS framework for rapid UI development
shadcn/ui	High-quality, accessible UI components built on Radix UI
React Query	Server state management and data fetching library

## A.2 Backend Technologies

**Table A.2:** Backend Technology Stack

Technology	Purpose
Node.js	JavaScript runtime for server-side execution
tRPC	End-to-end typesafe API layer for seamless client-server communication
Drizzle ORM	TypeScript ORM for database operations with type safety
PostgreSQL	Robust relational database for data persistence
Neon	Serverless PostgreSQL hosting with auto-scaling capabilities
Clerk	Authentication and user management service

## A.3 AI/ML Technologies

**Table A.3:** AI/ML Technology Stack

Technology	Purpose
OpenAI API	GPT-4 for generating video titles, descriptions, and thumbnails
TensorFlow.js	Client-side machine learning for real-time recommendations
Python (Flask)	Microservice for ML model inference and training
Scikit-learn	Collaborative filtering and content-based recommendation algorithms

## A.4 DevOps & Infrastructure

**Table A.4:** DevOps and Infrastructure Stack

Technology	Purpose
Vercel	Deployment platform for Next.js with edge functions
GitHub	Version control and collaborative development
GitHub Actions	CI/CD pipeline for automated testing and deployment
Cloudflare	CDN for video content delivery and caching
Mux	Video encoding, storage, and adaptive bitrate streaming



## Appendix B

### Database Schema

The Neptune database is designed using PostgreSQL with Drizzle ORM for type-safe database operations. The schema follows normalized database design principles to ensure data integrity and efficient querying.

#### B.1 Core Tables

**Table B.1:** Database Tables Overview

Table Name	Description
users	Stores user profile information, roles, and authentication data
videos	Contains video metadata, URLs, thumbnails, and publishing status
comments	User comments on videos with parent-child relationships for replies
likes	Tracks user likes/dislikes on videos and comments
subscriptions	Manages channel subscriptions between users
playlists	User-created playlists with video collections
watch_history	Tracks user viewing history for recommendations
notifications	System and user notifications for engagement

#### B.2 Key Relationships

- **One-to-Many:** Users  $\rightarrow$  Videos (one user can upload many videos)
- **One-to-Many:** Videos  $\rightarrow$  Comments (one video can have many comments)
- **Many-to-Many:** Users  $\leftrightarrow$  Videos (through likes and watch history)
- **Many-to-Many:** Users  $\leftrightarrow$  Users (through subscriptions)
- **Many-to-Many:** Playlists  $\leftrightarrow$  Videos (playlist video associations)