Project

UTU-BE

Submitted by,

Uraj U.Sahu (202404104610003),

Zeenal K.Bhalodiya (202404104610029),

Abhishek J. Pandey (202404104610096),

Krishna P.Dudhat(202404104610121)

Guided by,

Mr. Amish Patel

for partial fulfillment of the requirements

for the Degree of Master of Computer Application,

Shrimad Rajchandra Institute of

Management and Computer

Application, Uka Tarsadia University,

March, 2025.

DECLARATION

We hereby declare that the project titled "UTU-BE" is fully implemented by us. It is neither paid nor copied. Even though, later on, in case of any infringement found for this project work, we are solely responsible for the same and understand that as per UGC norms, the University can revoke the degree conferred to us.

Enrollment No.	**Student Name**	**Signature**
202404104610003	Uraj U.Sahu	
202404104610029	Zeenal K.Bhalodiya	
202404104610096	Abhishek Pandey	
202404104610121	Krishna P.Dudhat	

As a guide, I assure you that there is no plagiarism found in the submitted document.

Guide Name	Signature

Date: **Place : Bardoli**

INDEX

1. Introduction Of System ---------- 1-2
   - 1.1 Purpose ---------- 1
   - 1.2 Definition ---------- 1
   - 1.3 Scope ---------- 2
   - 1.4 Tools & Technologies ---------- 2

2. System Analysis ---------- 3-8
   - 2.1 Requirements ---------- 3-4
     - 2.1.1 Functional Requirements ---------- 3-4
     - 2.1.2 Non- Functional Requirements ---------- 4
   - 2.2 Use Case Diagram ---------- 5-6
   - 2.3 Activity Diagram ---------- 7-8

3. System Design ---------- 9-16
   - 3.1 Database Design ---------- 9-11
   - 3.2 User Interface Design ---------- 12-16
     - 3.2.1 User Side ---------- 12-16

4. Data Model Specification ---------- 17-19
   - 4.1 Data Model Description ---------- 17-19

5. Glossary ---------- 20

INDEX OF FIGURE

Figure 2.2.1 – 1 Admin Use Case Diagram---------- 5
Figure 2.2.1 – 2 User Use Case Diagram ----------6
Figure 2.3.1 – 1 Upload Video Activity Diagram----------7
Figure 2.3.2 – 2 Customization Feed Activity Diagram----------8
Figure 2.3 – 1 Login Page ----------12
Figure 2.2.1 – 1 Login Successful Page ---------- 13
Figure 2.2.1 – 1 Home Page ----------14
Figure 2.2.1 – 1 Upload Video & Description Generate ---------- 15
Figure 2.2.1 – 1 Video Play Screen Page ---------- 16

Chapter 1 :- Introduction Of System

1.1 Purpose :-

UTU-Be is a university-based smart video content platform designed specifically for Uka Tarsadia University (UTU). It serves as a digital hub for students and faculty, enabling them to upload, explore, and manage educational, event-related, and entertainment videos in a seamless and organized manner.

Why Utu-Be ?

- Easy Access & Secure Login – Students and faculty can register and log in using their email and password.
- Trending Videos – Popular and important videos are highlighted based on views
- Quick Learning with Summaries – Automatic text summaries help users understand videos faster.
- Language Support – Real-time translation makes videos accessible to all.
- Notifications & Alerts – Keeps users updated on new uploads, reports, and system alerts.

1.2 Definition :-

UTU-Be is a smart video platform designed for Uka Tarsadia University (UTU) to help students and faculty easily upload, watch, and manage educational and event-related videos. It uses AI to recommend videos, translate content, summarize videos, and organize them into categories like education, events, and entertainment. The platform also provides secure login, trending video highlights, personalized feeds means a customized list of videos that matches a user's interests, preferences, and past activity.

1.3 Scope :-

- **Target Audience - Designed exclusively for Uka Tarsadia University (UTU) students and faculty.**

- **User Access - Provides secure access where students and faculty can register and log in using their email and password.**

1.4 Tools & Technologies :-

Database	**MYSQL(8.4.4)**
Frontend,Backend	**Flutter For - Frontend** **Django For - Backend**
IDE	**Android Studio ,Visual Studio Code**
Tools For Diagram	**Draw.io**
Tools For Development	**Whisper Large (v3), Speech-to-Text,movie.py**
Version Control	**GitHub**

Chapter 2 :- System Analysis

2.1 Requirements :-

The **UTU-Be system** is designed to provide a **smart video platform** for Uka Tarsadia University (UTU), allowing students and faculty to **upload, explore, and manage educational and event-related videos** efficiently.

2.1.1 Functional Requirements :-

- **Admin Functionality :-**
  - **Admin Login :-** Admin can login using their id and password.
  - **Manage User :-** The Admin has the ability to manage user accounts by performing actions such as activating, deactivating.
  - **Manage Video :-**The Admin can Manage the video likes handling all video-related activities like uploading, editing, deleting, and organizing videos.
  - **Report Generate** :- Admin can Generate Report using collected data or activities. t includes summarizing information, finding patterns, and showing results in a simple format like dashboard.

- **User Functionality :-**
  - **User Login :-** Students can log in using their enrollment and password and faculty can login using their employee id and password.

- **Video Upload & Management :-** Users can upload, edit, and manage their videos.
- **Trending Video Algorithm :-** Displays popular videos based on views and likes.
- **Video Summarization :-** Automatically generates short text summaries of videos.
- **Intelligent Video Organization :-** Automatically categorizes videos into topics like education, events, and entertainment.
- **Notifications & Alerts :-** Users receive updates on new uploads, reports, and system alerts.
- **Real-time Language Translation :-** Converts video subtitles or spoken content into different languages.
- **Reports & Analytics :-** Generates reports on user activity, video engagement, and system performance.
- **Personalized Feed** – AI suggests videos based on user interests and past activities.

2.1.2 Non-Functional Requirements :-

- **Security & Authentication –** User data and video content should be protected with secure login and access control.
- **User-Friendly Interface –** The platform should be easy to navigate and accessible for all users.
- **Performance & Speed –** Videos should load quickly, and the recommendation system should work efficiently.
- **Reliability & Availability –** The system should be available 24/7 without major downtime.

2.2 Use Case Diagram :-

2.2.1 Admin Use Case Diagram :-

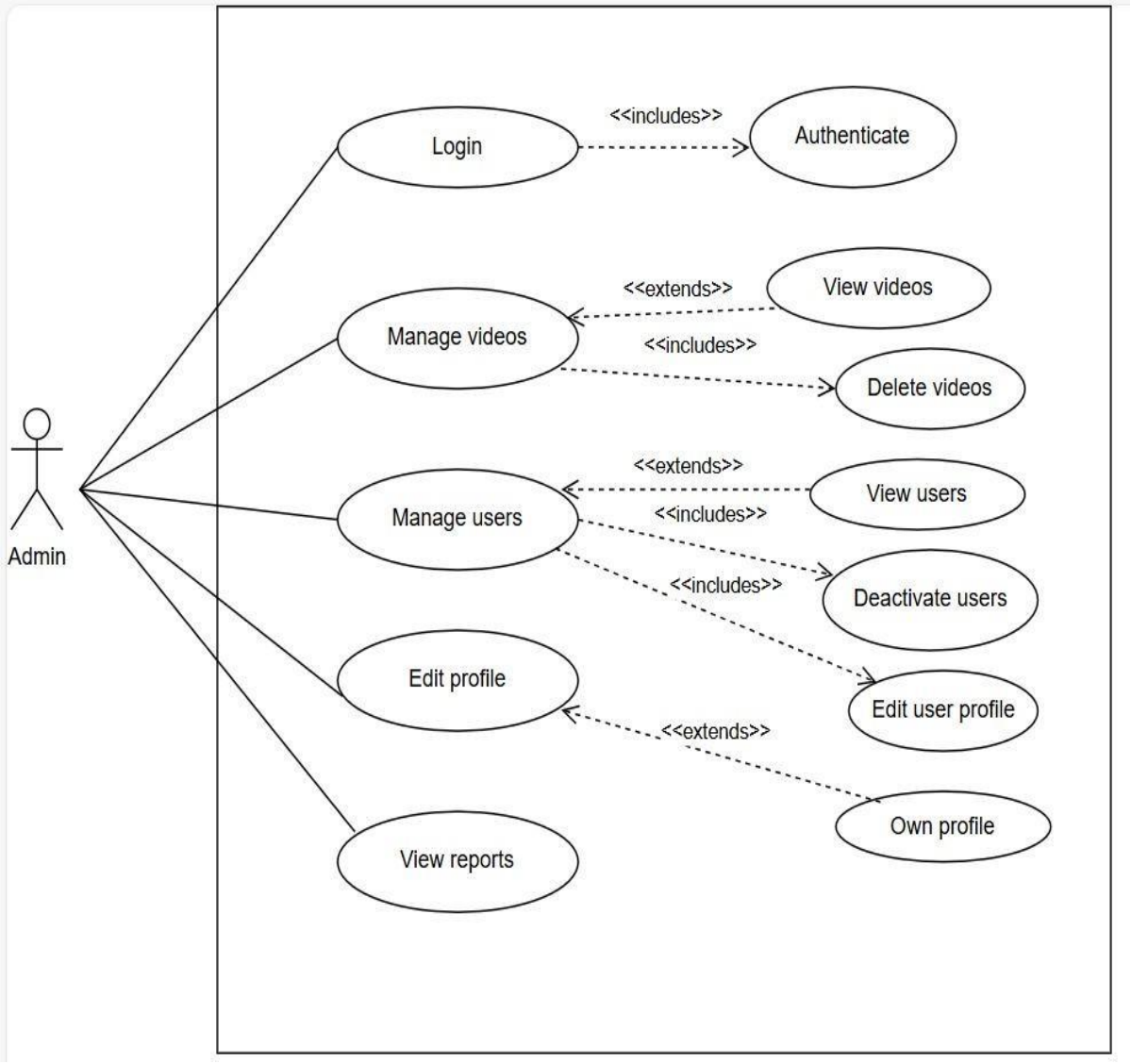

Figure 2.2.1 – 1 Admin Use Case Diagram

2.2.2 User Use Case Diagram :-

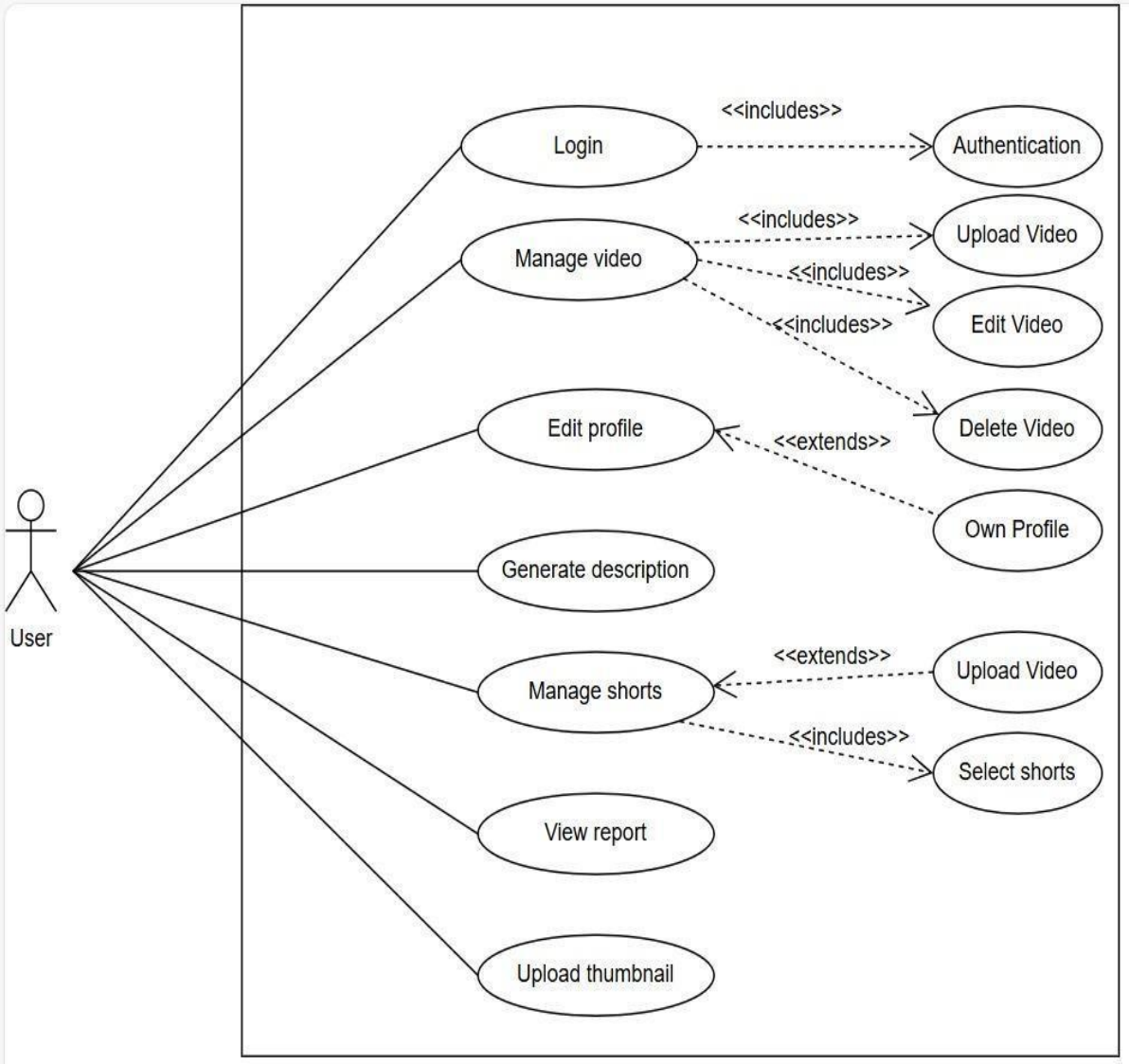

Figure 2.2.1 – 2 User Use Case Diagram

2.3 Activity Diagram :-

2.3.1 Upload Video Activity Diagram :-

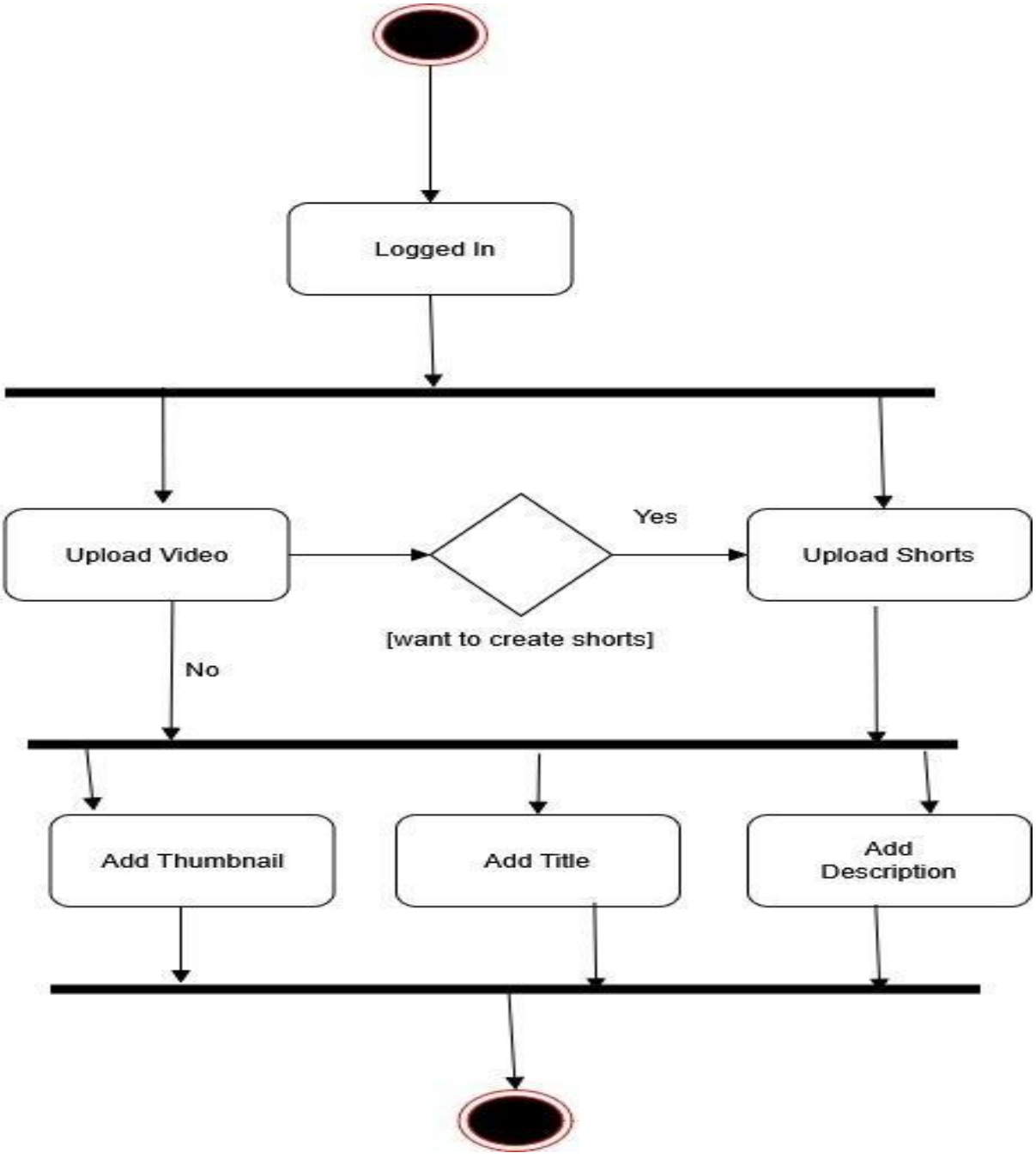

Figure 2.3.1 – 1 Upload Video Activity Diagram

2.3.2 Customization Feed Activity Diagram :-

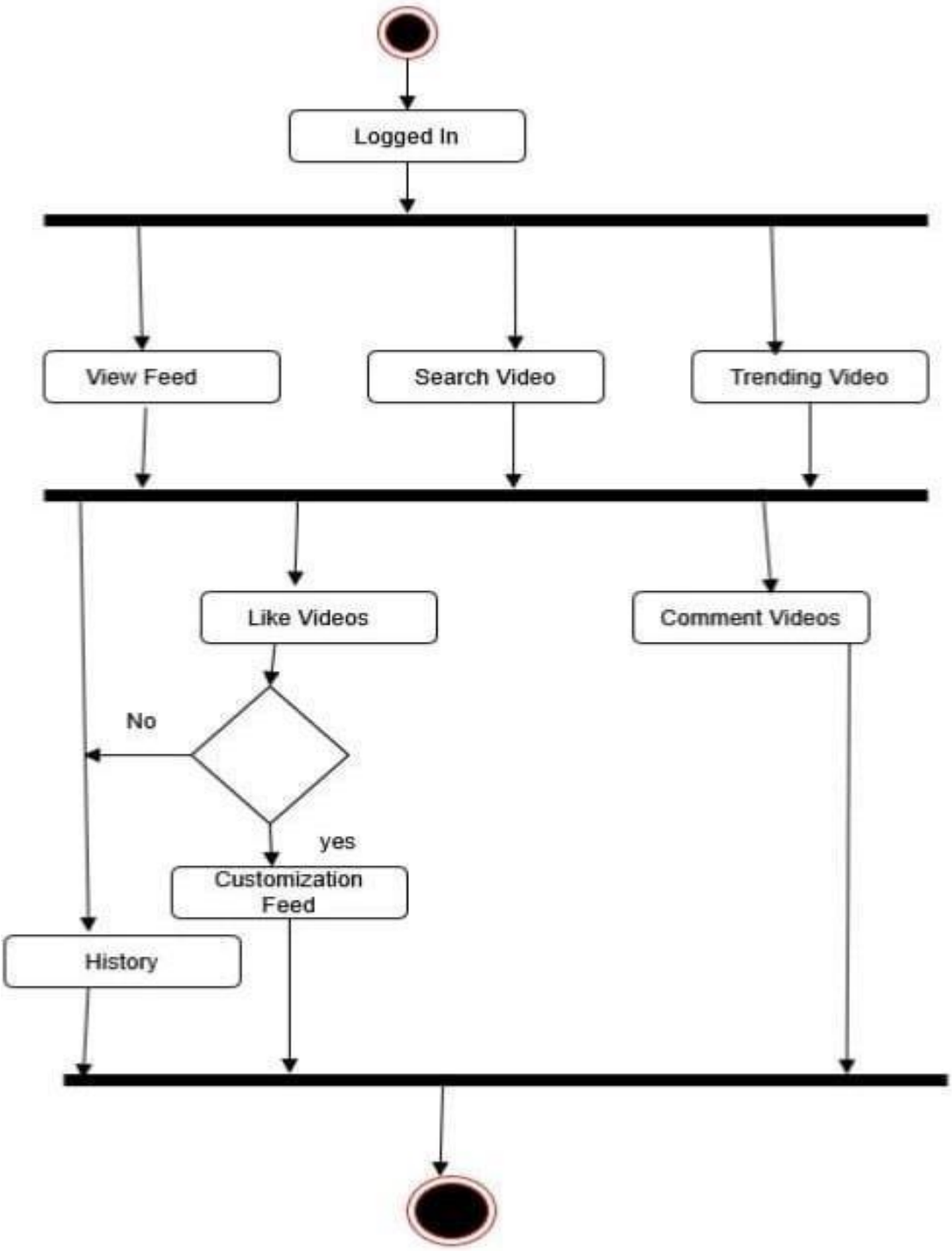

Figure 2.3.1 – 2 Customization feed Activity Diagram

Chapter 3 :- System Design

3.1 Database Design :-

3.1.1 Admin & User

1) users table :-

Purpose :- This table stores user authentication details, including enrollment numbers and passwords.			
Column name	**Data type**	**Constraint**	**Description**
enrollment	varchar(20)	Primary key , not null	Unique identifier for each user (e.g., student or employee ID).
password	varchar(20)	not null	Hashed password for authentication.

2) video table :-

Purpose :- This table store metadata of uploaded video,including uploaders enrollment number, file details and description			
Column name	**Data type**	**Constraint**	**Description**
id	int	AUTO_INCREMENT , Primary Key	Unique identifier for each video
video_name	varchar(50)	not null	name of video file
video_file	varchar(50)	not null	path or url of video file
upload_date	datetime	not null	timestamp of video uploded
description	varchar(290)	not null	description of video
enrollment	varchar(20)	not null,Foreign Key	enrollment number of user who upload video

3) user_comment table :-

Purpose :- This table stores comments made by users on videos, tracking the commenter, timestamp, and associated video.			
column_name	**Data type**	**Constraint**	**Description**
id	int	AUTO_INCREMENT , Primary Key	Unique identifier for each comment.
enrollment	varchar(20)	Not Null ,Primary Key	Enrollment ID of the user who made the comment (references `users` table).
comment	varchar(100)	NOT NULL	Comment of user
comment_at	Timestamp	DEFAULT CURRENT_TIMESTAMP	Timestamp when the comment was made.
video_id	int,Foreign Key	NOT NULL, Foreign Key	The ID of the video the comment is associated with (references `videos` table).

4) video_like table :-

Purpose :- Tracks user likes/dislikes on videos to analyse engagement and prevent duplicate reactions.			
Column_name	**Data_type**	**Constraint**	**Description**
id	int	AUTO_INCREMENT, Primary Key	unique id for each record
user_id	varchar(20)	Not Null	id of user who like video
video_id	int,Foreign Key	Not Null, Foreign Key (video table)	video which is like by user

status	varchar(10)	check("like","dislike")	store status whether like or not
time_stamp	datetime	default current timestamp	store time when like perform occur

5) utube_videoviewlog table :-

Purpose: Logs video views with user enrollment details and timestamps for tracking watch history and engagement analytics.			
Column_name	**Data_type**	**Constraint**	**Description**
id	int	AI, PK	uniquely identify each videos
enrollment	varchar(20)	NOT NULL	enrollment num of user who watch video
video_id	int	NOT NULL,FK	ID of video being watched
video_id	datetime	NOT NULL	timestamp when video watch

3.2 User Interface Design :-

1) Login

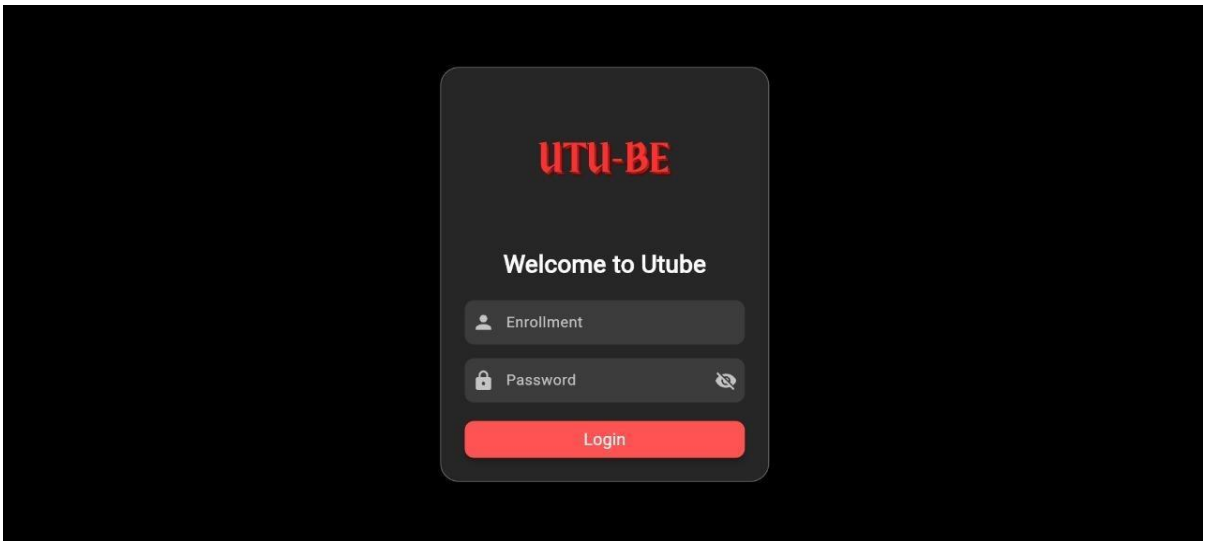

Figure 3.2 – 1 Login Page

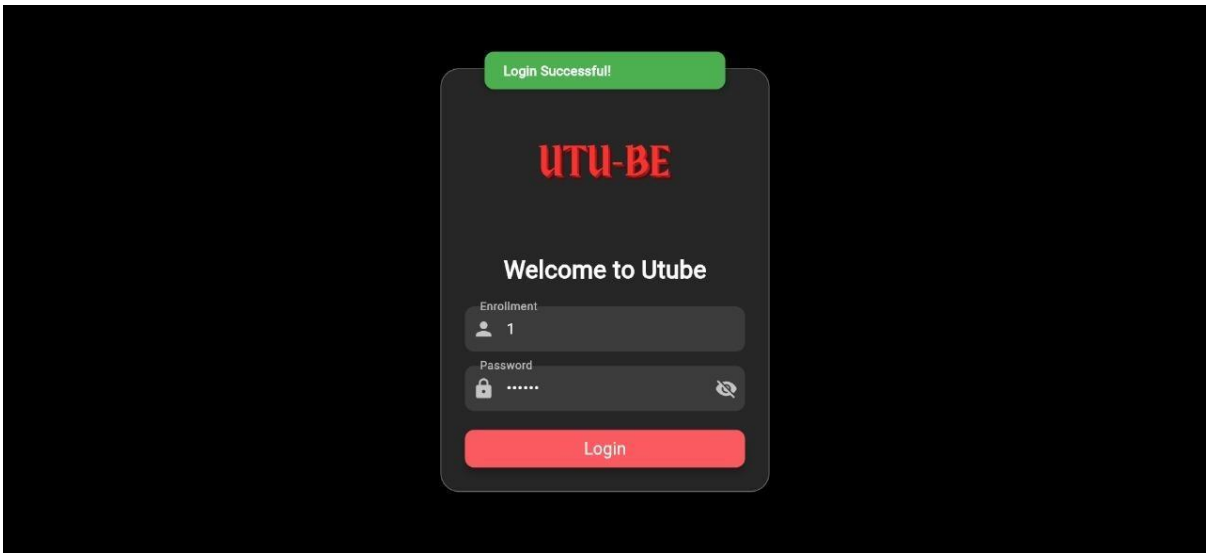

Figure 3.2 – 2 Login Successful Page

Description	This is login page of system
Data from	Fetch data to match enrollment and password from users table.
Data to	
Critical Validations	Users can login with valid credentials. If credentials invalid then system will not allow for login. Admin email and password remained static.

2) Home Page

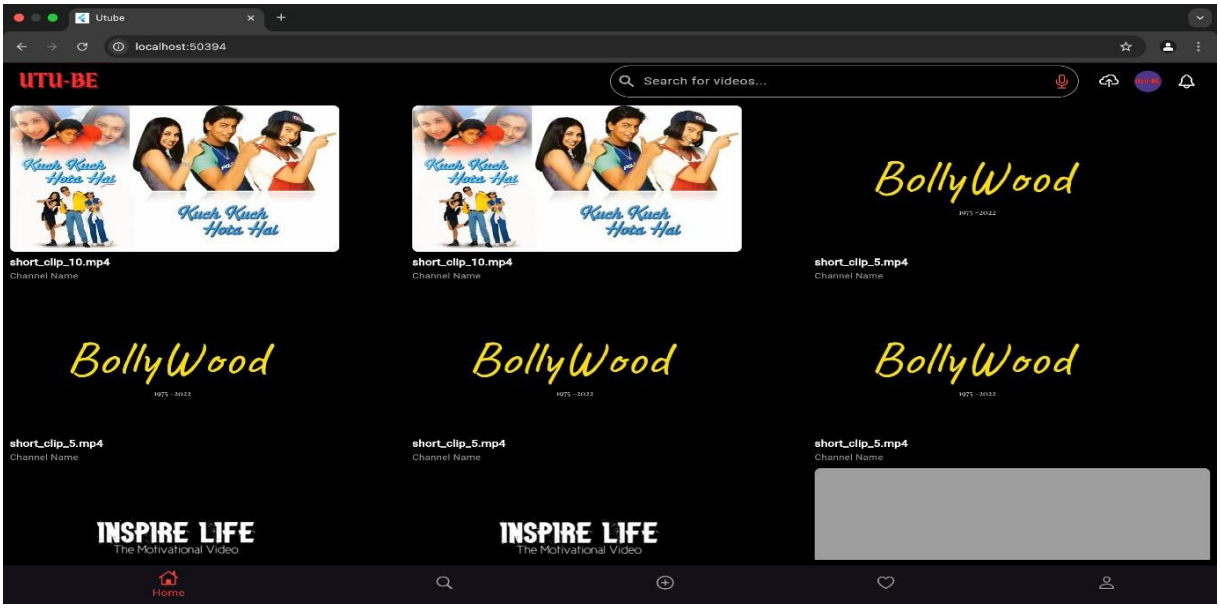

Figure 3.2 – 3 Home Page

Description	This is homepage of system. In it video are show , user able to search video using text and voice, access user profile . user able to upload video .
Data from	Fetch video from videos table and and also show video related to user preference .
Data to	
Critical Validations	only show videos related to user preference and trending videos

3) upload video and description generation

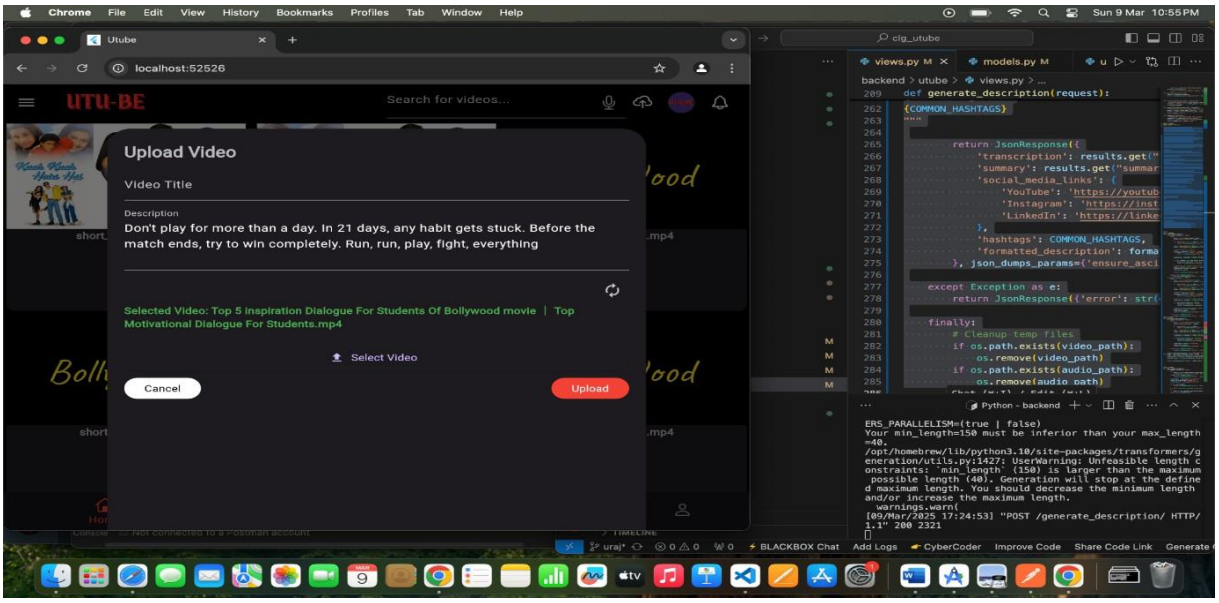

Figure 3.2 – 4 Video Upload & Description Generate Page

Description	The user is able to upload video from his/her account. user also generate description automatically .
Data from	
Data to	The video which uploaded by user will store in videos table with all details
Critical Validations	Title is not more than 100 words and description not more than 250 words and if video has only music then it not generate description.

4) video play screen

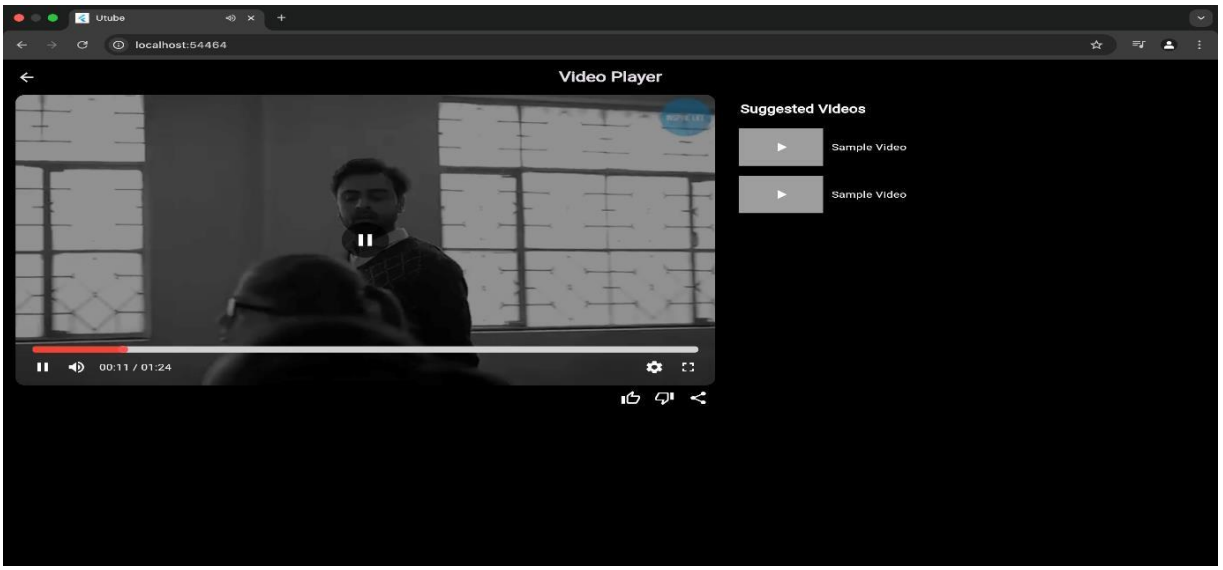

Figure 3.2 – 5 Video Play Screen Page

Description	In this user able to watch video like youtube , like , share ,comment on video ,suggested video and generate summary of video
Data from	The video which is play fetch from videos table .
Data to	like,comment
Critical Validations	

Chapter 4 :- Data Model Specification

4.1 Data Model Description :-

➢ Introduction :-

- A **data model** defines the structure, organization, and relationships of data within the system. It provides a blueprint for how data is stored, accessed, and managed to ensure consistency, efficiency, and scalability..

➢ A well-defined data model helps to :-

- **Organize data in a structured way –** Makes it easy to store, retrieve, and manage information efficiently.
- **Maintain data accuracy and consistency –** Uses rules like primary keys, foreign keys, and constraints to prevent duplicate or incorrect data.
- **Improve communication between teams –** Provides a clear structure that helps developers, analysts, and business teams understand how data is connected.

➢ For Our project We Used the Agile Data Model :

➢ What is Agile :-

- The Agile Model is a software development approach that focuses on flexibility, collaboration, and continuous improvement. Unlike traditional models like the Waterfall Model, where all planning happens at the beginning of a process, Agile allows teams to develop software in small, manageable steps while continuously gathering feedback and making improvements.

➢ System Approach Flow of the Agile :-

1) Requirement Analysis & Planning

- **Set Clear Goals** – Decide what the project should achieve.
- **Understand Business Needs** – Identify important features and how they help users.

2) Sprint(Short) Planning & Task Allocation

- **Break Work into Small Tasks –** Divide the project into short work cycles (Sprints).
- **Assign Work –** Give tasks to team members based on their skills.

3) Design & Development

- **Start with the Main Features –** Build the most important parts first.
- **Use a Flexible Design –** Create a system that can grow and adapt.

4) Testing & Quality Assurance

- **Check System Features –** Ensure everything functions smoothly.
- **Fix Problems –** Solve any bugs or slow performance issues.

5) Feedback & Review

- **Ask Users –** Collect opinions and suggestions.
- **Find What Needs Improvement –** Identify issues or missing features.

- **Make Changes –** Update the system based on feedback**.**

6) Deployment & Implementation

- **Make It Live** – Release the system for users.
- **Step-by-Step Deployment –** Release it step by step to prevent problems.
- **Track Performance –** Monitor how well it runs in real time.

7) Continuous Monitoring & Improvement

- **Monitor Performance –** Check how well the system is working**.**
- **Fix Problems & Improve Speed –** Solve issues and make it run faster**.**
- **Upgrade Features –** Add updates based on business needs.

Chapter 5 :- Glossary

1) **Whisper Large (v3)** - An AI-based speech recognition model used for video transcription in UTU-Be.
2) **Speech-to-Text** - A technology that converts spoken language into written text.
3) **Transformers** – Can be used for tasks like generating movie summaries or analyzing subtitles.
4) **Django REST Framework (DRF)** – To create an API that serves movie-related data.
5) **PySceneDetect** – To detect scene changes in a movie/video file.
6) **CSRF Protection** – Ensuring secure API requests.

Reference

StackOverFlow :- https://stackoverflow.com/

Research Paper :- https://www.researchgate.net/publication/372339111_Video_Transcript_Summarizer

GitHub:- https://github.com/huggingface/transformers/blob/main/docs/source/en/model_doc/whisper.md