<Real-Time Deepfake Detection for Video Streaming Platforms>

Design Document

Version 1.0

Group Id: <>

Supervisor Name :<>

Revision History

Date (dd/mm/yyyy)	Version	Description	Author
27-Feb-2025	1.0	The project aims to develop a real-time deep-fake detection system that analyzes video streams to detect manipulated content, addressing growing concerns regarding media authenticity, misinformation and identity theft. With the emergence of deep-fake technology, there is a dire need for systems that can distinguish authentic videos from altered ones, particularly in live streaming and video conferencing contexts.	Bisma Azeem

Table of Contents

1. Introduction of Design Document:

This document outlines the design specifications for a "Real-Time Deepfake Detection System for Video Streaming Platforms". It serves as a blueprint for the development and implementation of the system, guiding the project throughout its lifecycle.

1.1 Purpose:

This design document serves as the foundational guide for the development of the "Real-Time Deepfake Detection System for Video Streaming Platforms." It outlines the architectural decisions, design choices, and implementation strategies that will shape the entire project. This document will act as a road-map, ensuring that the development process remains focused, organized, and aligned with the project's objectives. By meticulously adhering to the design specifications outlined herein, the project will progress efficiently, minimize unforeseen challenges, and ultimately deliver a robust and effective deepfake detection system.

1.2 Contents:

The contents of the design document are as follows:

- **Entity Relationship Diagram (ERD):** This section will include an ERD depicting the relationships between key entities within the system.

- **Sequence Diagrams:** This section will provide sequence diagrams for each of the use cases outlined in the SRS document, illustrating the sequence of interactions between actors involved in each use case.

- **Architecture Design Diagram:** This section will present a tiered architecture diagram illustrating the different components of the system.

- **Class Diagram:** This section will include a class diagram depicting the classes and their relationships within the system, providing a detailed view of the system's object-oriented structure.

- **Database Design:** This section will detail the organization of data within the system, according to the database model. It will present the database schema, defining the structure of data elements and their interrelationships.

- **Interface Design:** This section will include screen-shots of the proposed Graphical User Interface (GUI) for the system, showcasing 3-4 key interfaces to provide a visual representation of the user experience.

- **Test Cases:** This section will outline test cases for each of the use case scenarios.

1.3 Benefits:

- **Clear Communication:** This document provides a clear and concise roadmap for the development process, ensuring all stakeholders are aligned on the project's goals, scope, and technical approach.

- **Reduced Development Time:** By identifying and addressing potential challenges upfront, this document helps to minimize development time and prevent costly iterations.

- **Improved Project Quality:** A well-defined design leads to a more organized and structured development process, resulting in a higher quality and more robust system.

- **Enhanced Project Management:** This document serves as a valuable reference for project tracking and progress monitoring, facilitating effective project management.

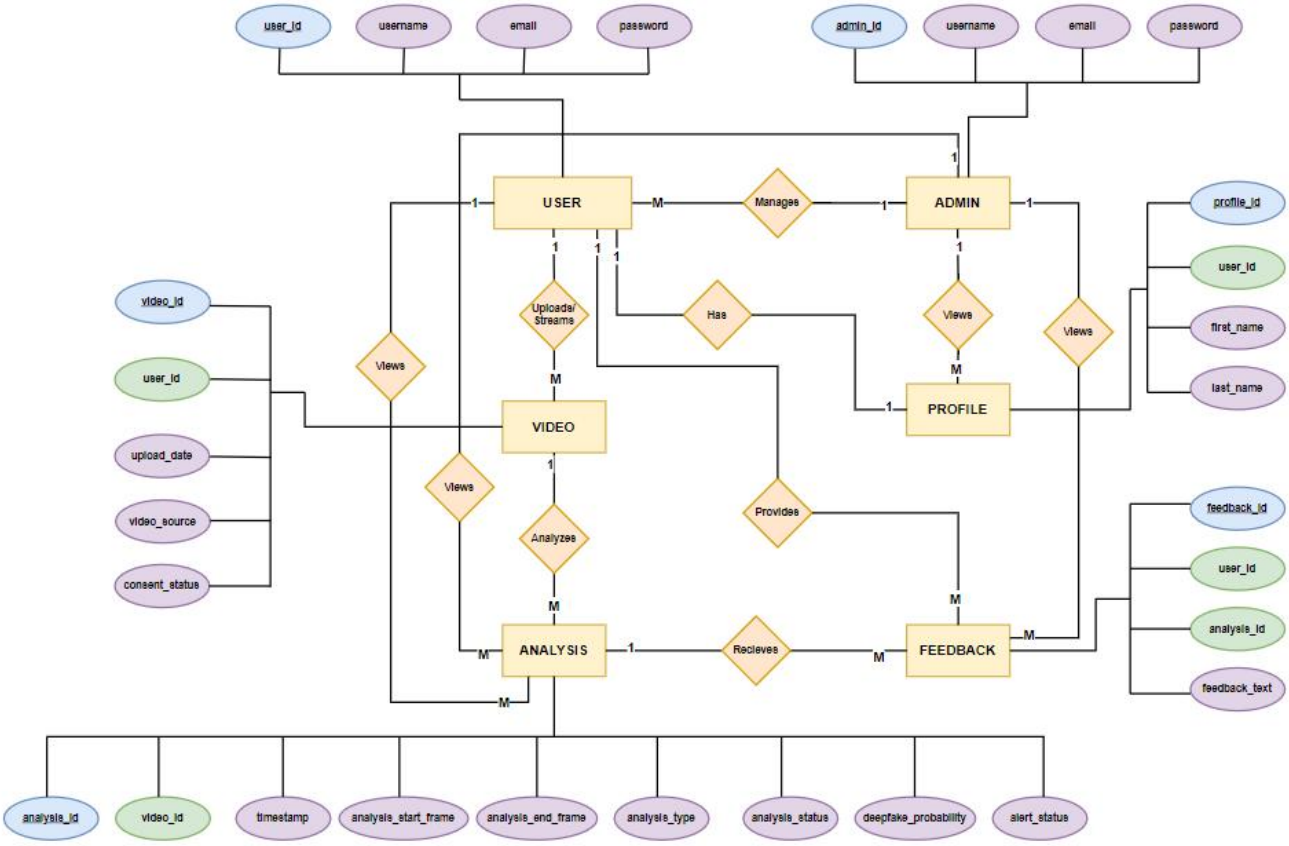
- **Improved Documentation:** The design document itself serves as valuable documentation for future reference, maintenance, and potential enhancements to the system.

- **Demonstrates Understanding:** This document demonstrates a thorough understanding of software engineering principles, design methodologies, and the specific requirements of the project.

- **Risk Reduction:** By identifying and mitigating potential risks early in the development cycle, the design document helps to reduce project risks and increase the likelihood of successful project completion.

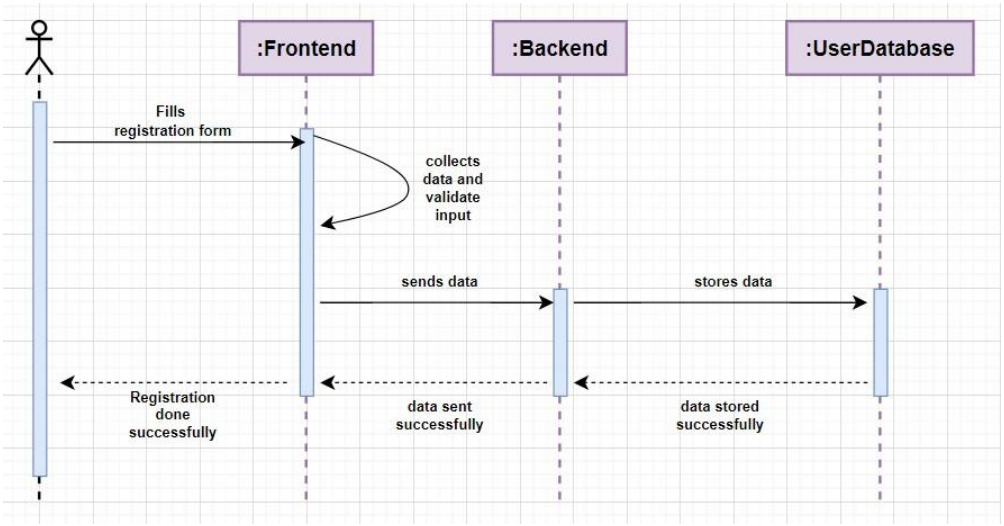
- **Enhanced Maintainability:** A well-structured design makes the system easier to maintain, update, and modify in the future, reducing the effort and cost associated with ongoing maintenance.

2. Entity Relationship Diagram (ERD):

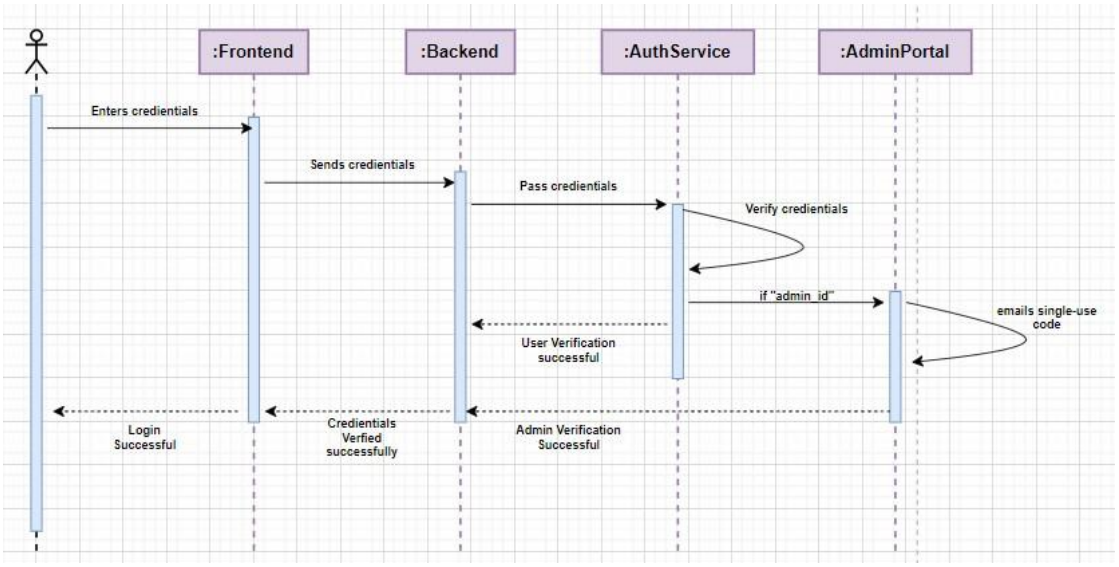


3. Sequence Diagrams:

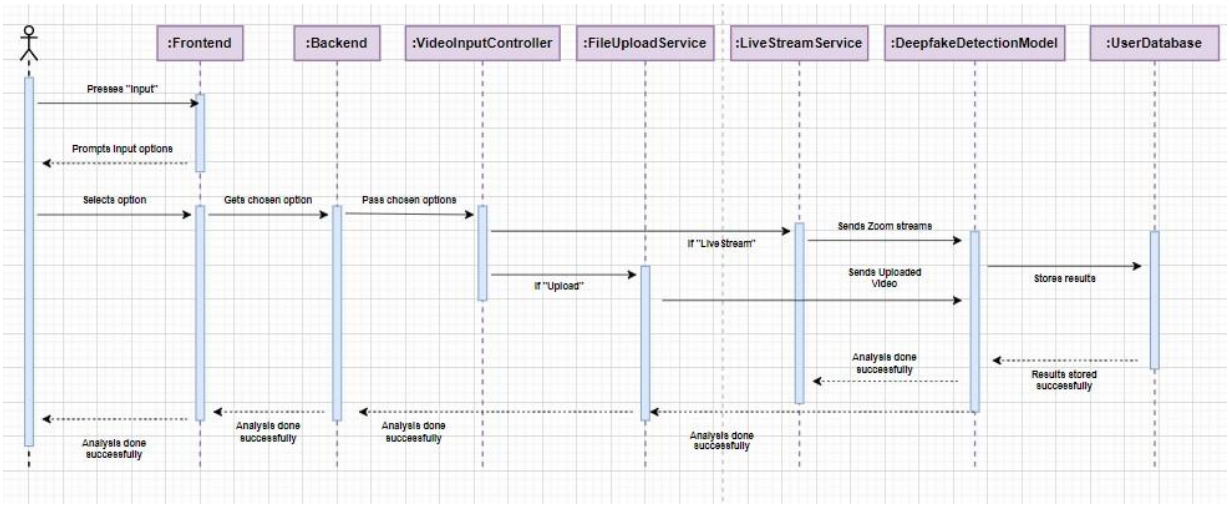
3.1: Register/Signup:



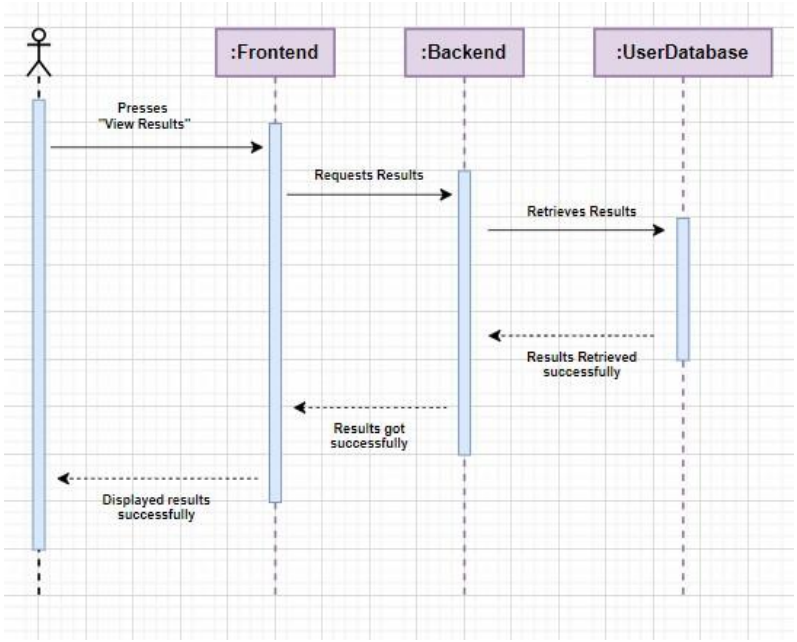
3.2 : Login:



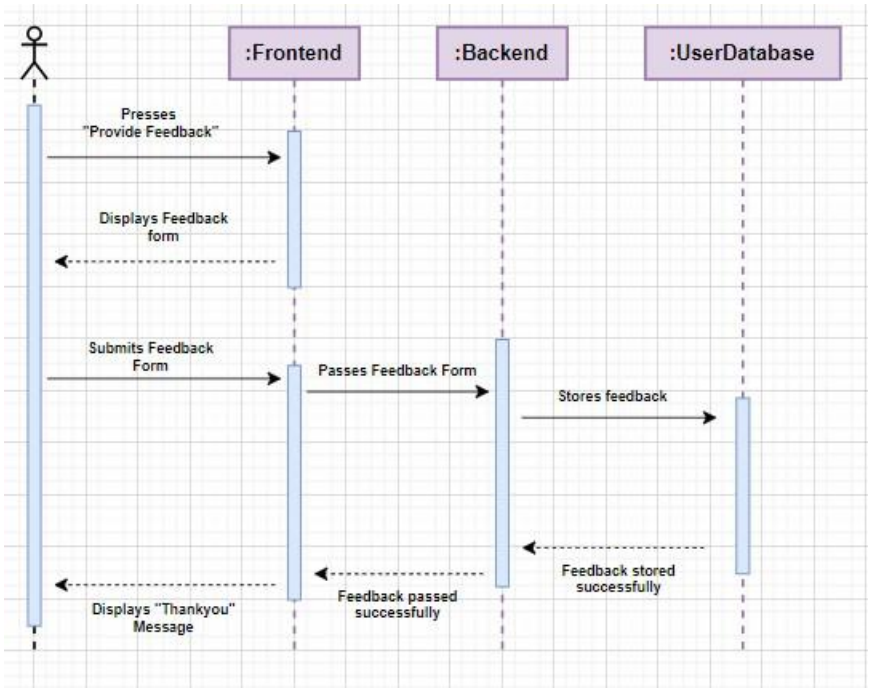
3.3 : Video Input & Processing:



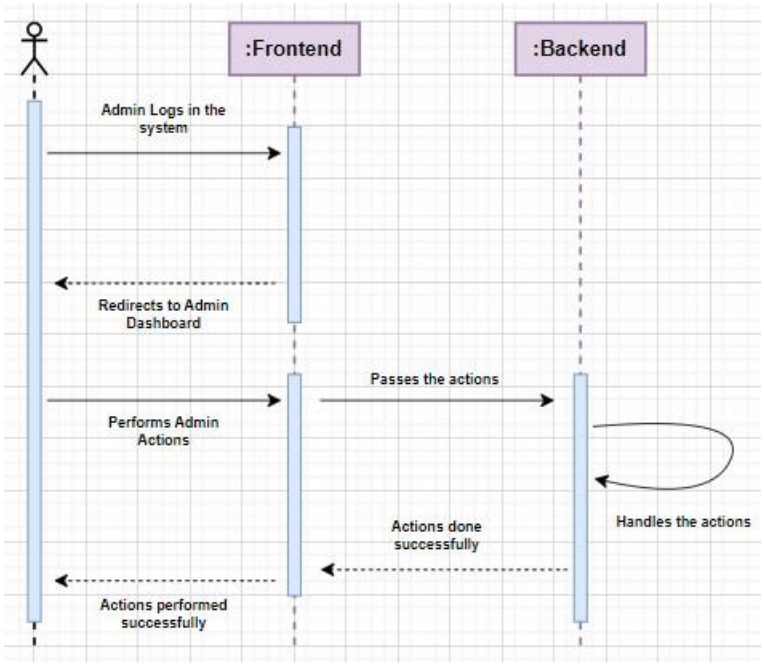
3.4 : View Results:



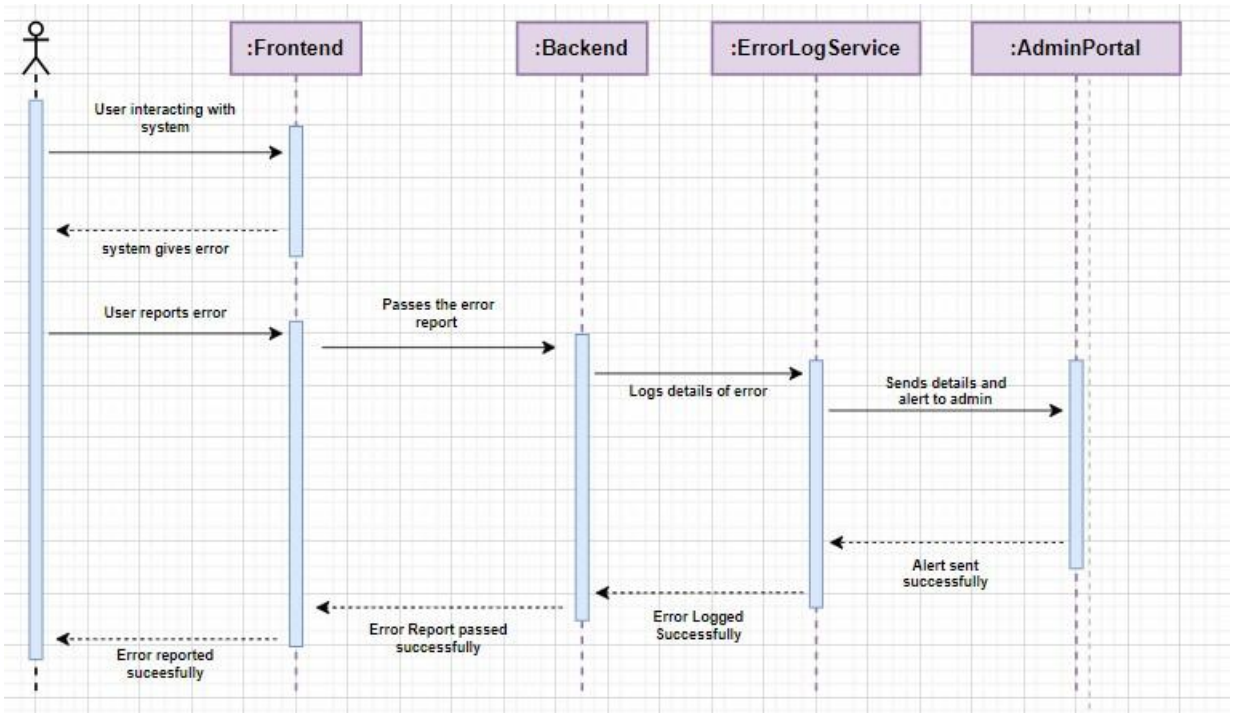
3.5 : User Feedback:



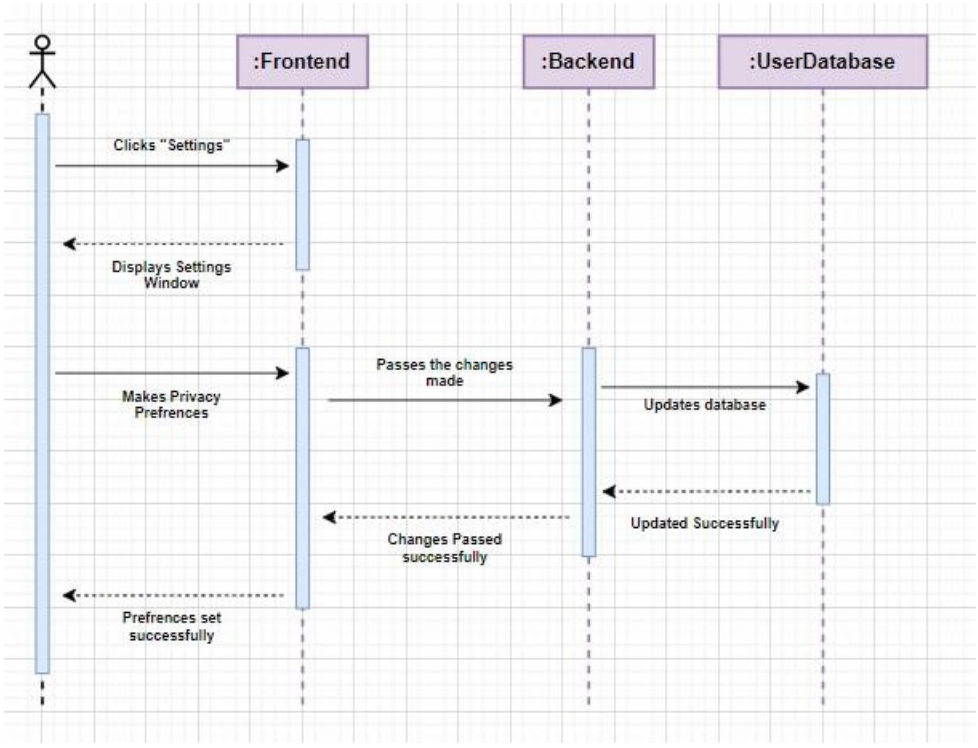
3.6 : Admin Management:



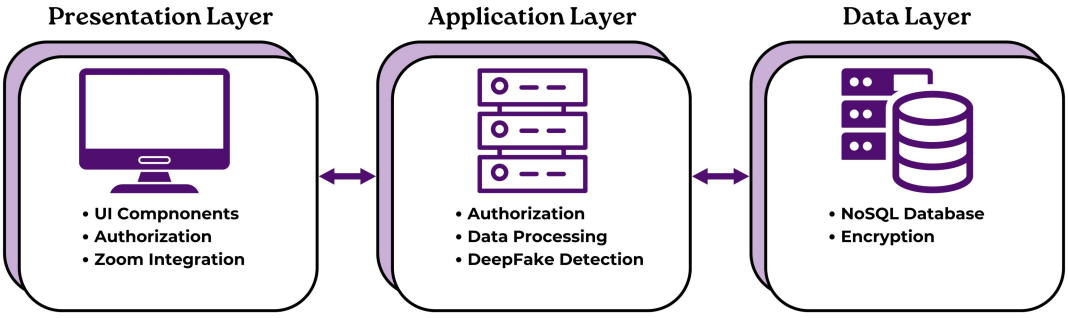
3.7 : Error Handling:



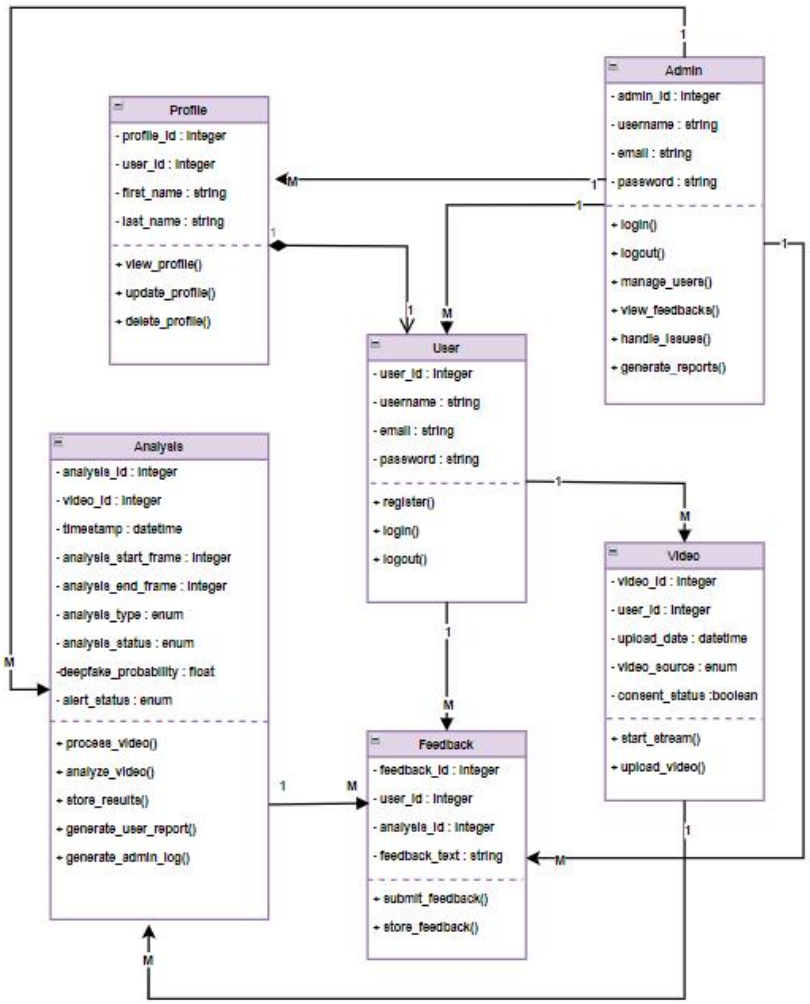
3.8 : Data Privacy Management:



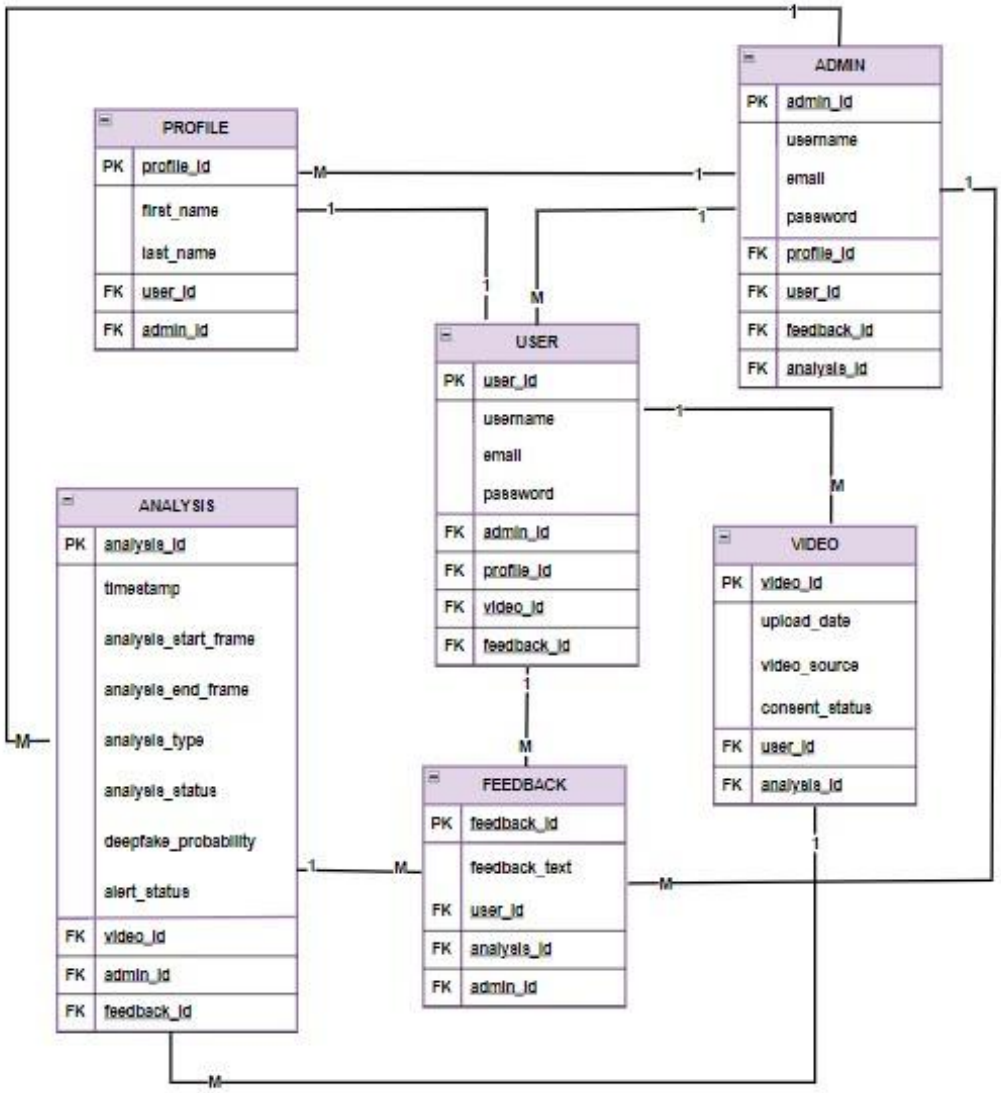
4. Architecture Design Diagram:



5. Class Diagram:

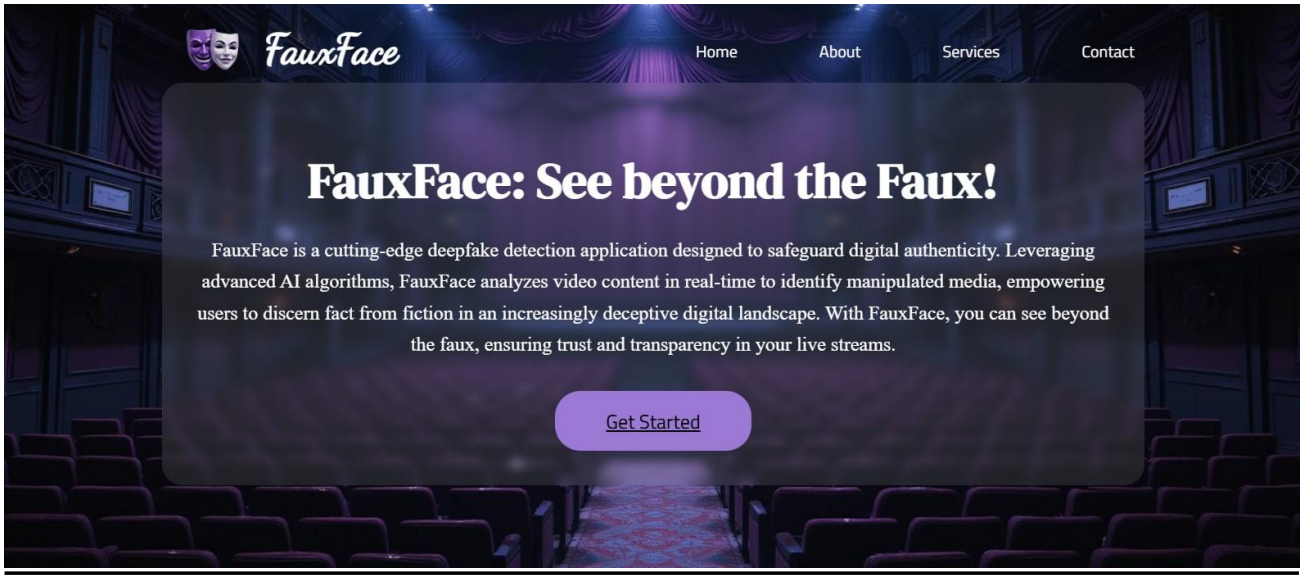


6. Database Design:

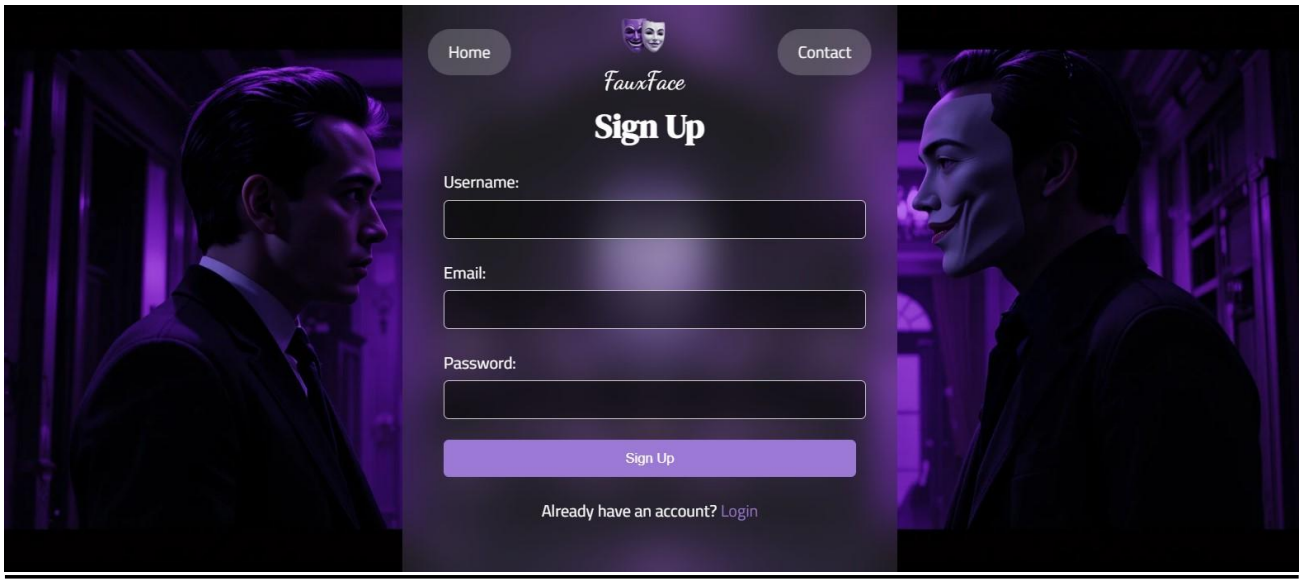


7. Interface Design:

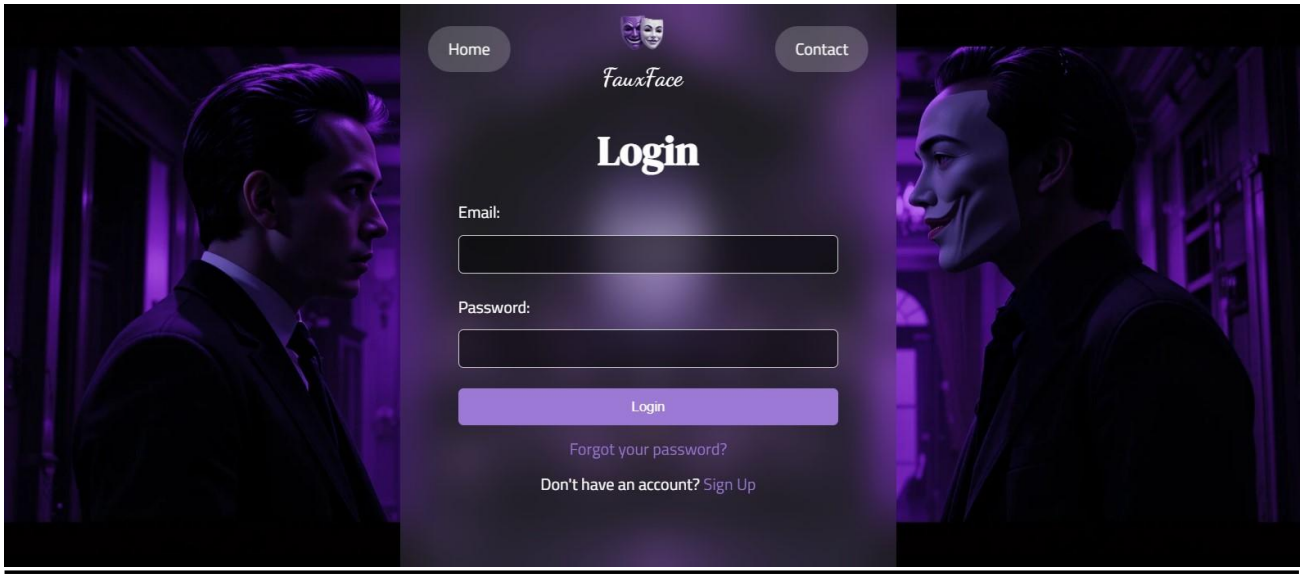
Homepage



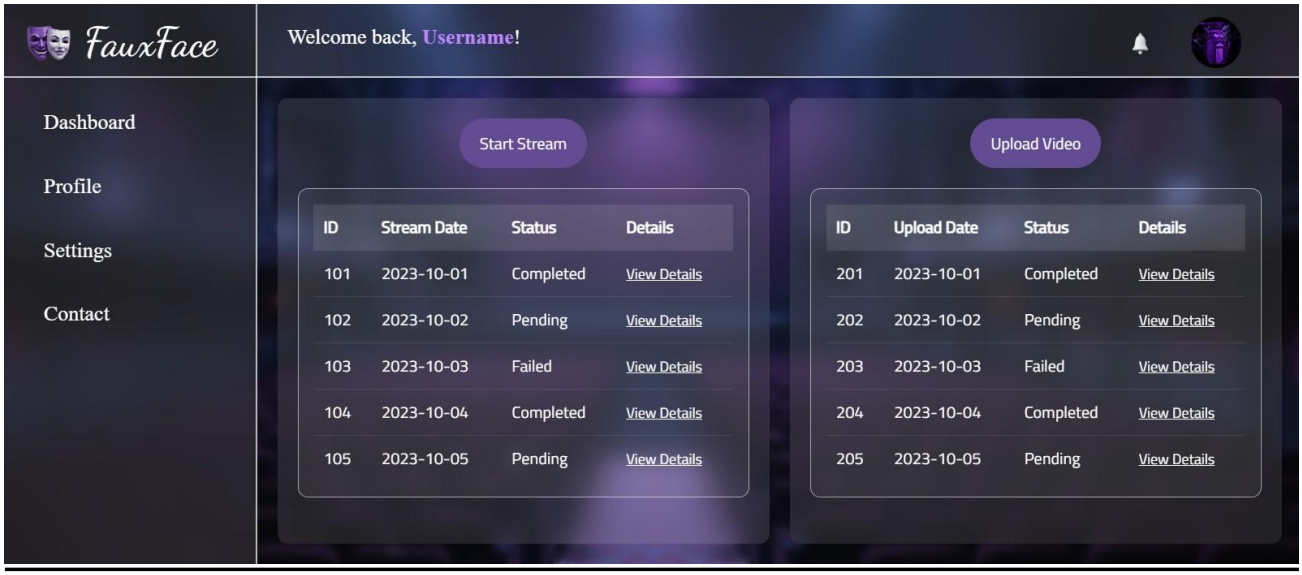
Sign-up Page



Login Page



User Dashboard



8. Test Cases:

Test Case Title:	***Successful User Registration***
Test Case ID:	TC-UC01-01
Use Case ID:	UC-01
Description:	Verify that a user can successfully register with valid details and complete email verification.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Registration page is accessible.
Task Sequence:	1. Open the registration page. 2. Enter valid credentials. 3. Submit the form. 4. Check email for verification link. 5. Click the verification link.
Post-conditions:	A new user account is created.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Registration with Existing Email***
Test Case ID:	TC-UC01-02
Use Case ID:	UC-01
Description:	Verify that the system prevents registration with an email that already exists in the system.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Registration page is accessible. 4. Email already exists in the system.
Task Sequence:	1. Open the registration page. 2. Enter valid credentials of already existing email. 3. Submit the form.
Post-conditions:	No new user account is created and “Email Already exists” is displayed.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Registration with Invalid Email***
Test Case ID:	TC-UC01-03
Use Case ID:	UC-01
Description:	Verify that system prevents registration with an invalid email format.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Registration page is accessible.
Task Sequence:	1. Open the registration page. 2. Enter credentials with invalid email (e.g user@com) 3. Submit the form.
Post-conditions:	No new user account is created and "Invalid Email Format" is displayed.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Registration with Weak Password***
Test Case ID:	TC-UC01-04
Use Case ID:	UC-01
Description:	Verify that system prevents registration with a weak password.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Registration page is accessible.
Task Sequence:	1. Open the registration page. 2. Enter credentials with weak password. (e.g "1234") 3. Submit the form.
Post-conditions:	No new user account is created and error message is displayed "Password must be at least 8 characters long and include 1 special characters."
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Failure to Send Verification Email***
Test Case ID:	TC-UC01-05
Use Case ID:	UC-01
Description:	Verify that system handles failure to send a verfication email.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Registration page is accessible. 4. Email service is down.
Task Sequence:	1. Open the registration page. 2. Enter valid credentials. 3. Submit the form.
Post-conditions:	No new user account is created and error message is displayed "Failed to send verification email. Please try again later. "
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful User Login***
Test Case ID:	TC-UC02-01
Use Case ID:	UC-02
Description:	Verify that a registered user can log in successfully with valid credentials.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Login page is accessible. 4. User has valid credentails. 5. Authentication service is active.
Task Sequence:	1. Open the login page. 2. Enter valid credentials. 3. Click "Login"
Post-conditions:	User is redirected to the dashboard.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Login with Invalid Credentials***
Test Case ID:	TC-UC02-02
Use Case ID:	UC-02
Description:	Verify that the system prevents login with invalid credentials.
Pre-conditions:	1. User has internet access. 2. Web app is running. 3. Login page is accessible. 4. User has invalid credentails. 5. Authentication service is active.
Task Sequence:	1. Open the login page. 2. Enter invalid credentials. 3. Click “Login”
Post-conditions:	User remains on login page and a error message displayed “Invalid Credentials”.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Forgot Password Functionality***
Test Case ID:	TC-UC02-03
Use Case ID:	UC-02
Description:	Verify that the “Forgot Password” option works as expected.
Pre-conditions:	1. User has forgoted their password. 2. Authentication service is active.
Task Sequence:	1. Open the login page. 2. Click “Forgot Password”. 3. Enter registered email. 4. Submit the form. 5. Check email for password reset link.
Post-conditions:	User recieves a password reset link via email.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Video Upload***
Test Case ID:	TC-UC03-01
Use Case ID:	UC-03
Description:	Verify that the user can successfully upload a pre-recorded video in the valid format and size.
Pre-conditions:	1. User has logged in. 2. Device supports video upload.
Task Sequence:	1. Click on “Upload Video” button. 2. Navigate to video input page. 3. Choose a valid video file. 4. Click “Submit”.
Post-conditions:	Video is submitted successfully for processing.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Live Streaming***
Test Case ID:	TC-UC03-02
Use Case ID:	UC-03
Description:	Verify that the user can successfully initiate live streaming using Zoom.
Pre-conditions:	1. User is logged in. 2. Device supports live streaming. 3. Zoom SDK is integrated and configured. 4. Camera access is granted.
Task Sequence:	1. Click on “Start Stream” button. 2. Navigate to video stream page. 3. Grant camera access when prompted. 4. Zoom SDK initializes and connects to the live stream. 5. Live streaming begins.
Post-conditions:	Live Streaming is successfully initiated, and video is processed by the system.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Live Streaming with camera access denied***
Test Case ID:	TC-UC03-03
Use Case ID:	UC-03
Description:	Verify that the system handles the scenario where camera access is denied during live streaming.
Pre-conditions:	1. User is logged in. 2. Device supports live streaming. 3. Zoom SDK is integrated and configured.
Task Sequence:	1. Click on “Start Stream” button. 2. Navigate to video stream page. 3. Deny camera access when prompted.
Post-conditions:	User cannot proceed with live streaming and a error message is displayed “Camera access is required for live streaming”.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Uploading an unsupported video format***
Test Case ID:	TC-UC03-04
Use Case ID:	UC-03
Description:	Verify that the system prevents uploading a video in an unsupported format.
Pre-conditions:	1. User has logged in. 2. Device supports video upload.
Task Sequence:	1. Click on “Upload Video” button. 2. Navigate to video input page. 3. Choose a invalid video file.(e.g AVI) 4. Click “Submit”.
Post-conditions:	Video is not submitted for processing and an error message is displayed “Unsupported video format. Please upload a valid format (e.g MP4)”
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Deepfake Detection in Uploaded Video***
Test Case ID:	TC-UC04-01
Use Case ID:	UC-04
Description:	Verify that the system successfully detects deepfake content in an uploaded video
Pre-conditions:	1. Video input is done successfully. 2. Deepfake detection model is operational.
Task Sequence:	1. System processes the uploaded video frame by frame. 2. Generates detection results with confidence scores. 3. Flags manipulated content
Post-conditions:	Manipulated content is detected and flagged.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Deepfake Detection in Live Stream***
Test Case ID:	TC-UC04-02
Use Case ID:	UC-04
Description:	Verify that the system successfully detects deepfake content during live stream and triggers alerts.
Pre-conditions:	1. Live streaming is active 2. Deepfake detection model is operational.
Task Sequence:	1. System processes the live stream frames in real-time. 2. Generates detection results with confidence scores. 3. Triggers real-time alerts if deepfake content is detected.
Post-conditions:	Manipulated content is detected in real-time and alerts are triggered successfully.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Network connectivity failure during uploaded video processing.***
Test Case ID:	TC-UC04-03
Use Case ID:	UC-04
Description:	Verify that the system handles network connectivity issues during deepfake detection for an uploaded video.
Pre-conditions:	1. Video input is done successfully. 2. Deepfake detection model is operational. 3. Network Connectivity is lost during processing.
Task Sequence:	1. System processes the uploaded video frame by frame. 2. Network Connectivity is lost. 3. System detects the network failure.
Post-conditions:	Processing is paused until Network Connectivity is restored and displays "Network Connectivity lost. Processing paused."
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Deepfake Detection system crashes during Live Stream***
Test Case ID:	TC-UC04-04
Use Case ID:	UC-04
Description:	Verify that the system handles a crash in the deepfake detection model during live stream.
Pre-conditions:	1. Live streaming is active 2. Deepfake detection model is operational. 3. Detection system crashes during processing.
Task Sequence:	1. System processes the live stream frames in real-time. 2. Deepfake detection model crashes 3. System detects the crash.
Post-conditions:	Live stream processing is paused and displays "Deepfake detection system crashed. Please wait a moment or try again."
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Viewing of Analysis Results***
Test Case ID:	TC-UC05-01
Use Case ID:	UC-05
Description:	Verify that the user can successfully view the analysis results.
Pre-conditions:	1. Detection process has been completed. 2. Results are available for viewing.
Task Sequence:	1. User navigates to results page. 2. System displays the report with flagged frames and confidence scores. 3. User reviews the results.
Post-conditions:	User can review flagged frames and cofidence scores.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Fail to View Analysis Results***
Test Case ID:	TC-UC05-02
Use Case ID:	UC-05
Description:	Verify that the system handles the scenario where results cannot be viewed due to system error.
Pre-conditions:	1. Detection process has been completed. 2. System error occurs while fetching results.
Task Sequence:	1. User navigates to results page. 2. System encounters an error while fetching results. 3. System displays error message.
Post-conditions:	User cannot view the results and is prompted “Try again later.”
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Submission of user feedback***
Test Case ID:	TC-UC06-01
Use Case ID:	UC-06
Description:	Verify that the user can successfully submit feedback.
Pre-conditions:	1. User has reviewed detection results. 2. Feedback form is accessible.
Task Sequence:	1. User navigates to feedback form. 2. User provides feedback. 3. User submits the feedback. 4. System stores the feedback.
Post-conditions:	Feedback is saved successfully.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Failure in Submission of user feedback***
Test Case ID:	TC-UC06-02
Use Case ID:	UC-06
Description:	Verify that the system handles scenario where feedback cannot be submitted due to server issues.
Pre-conditions:	1. User has reviewed detection results. 2. Server is down or experiencing issues.
Task Sequence:	1. User navigates to feedback form. 2. User provides feedback. 3. User attempts submits the feedback. 4. System fails to process the submission due to server issues.
Post-conditions:	Feedback submission failed and user is prompted “Failed to submit feedback. Please try later.”
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful in Error Handling***
Test Case ID:	TC-UC07-01
Use Case ID:	UC-07
Description:	Verify that the system successfully identifies, logs and communicates error during video processing
Pre-conditions:	1. User has initiated video processing. 2. Error-handling mechanism is in place.
Task Sequence:	1. System encounters error during video processing. 2. System logs the error. 3. System displays meaningful message to the user.
Post-conditions:	Error is logged for the admin and comminucated to the user.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Failure in Error Handling***
Test Case ID:	TC-UC07-02
Use Case ID:	UC-07
Description:	Verify that the system fails to identify, log and communicate error during video processing
Pre-conditions:	1. User has initiated video processing. 2. Error-handling mechanism is in place. 3. System fails to detect the error.
Task Sequence:	1. System encounters error during video processing. 2. System fails to detect the error. 3. System display no meaningful message to the user.
Post-conditions:	System fails to log the error for the admin and comminucate to the user.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Admin login and user account management***
Test Case ID:	TC-UC08-01
Use Case ID:	UC-08
Description:	Verify that the admin successfully login and manage user accounts.
Pre-conditions:	1. Admin account is active and properly configuired.
Task Sequence:	1. Admin enters the previledged credentials. 2. Admin gets a single use authetication code via email. 3. Admin enters the code and is redirected to Admin dashboard. 4. Admin navigates to user management section and manages user accounts.
Post-conditions:	User accounts are managed successfully.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Unauthorized access attempt to admin panel***
Test Case ID:	TC-UC08-02
Use Case ID:	UC-08
Description:	Verify that the system handle Unauthorized access attempts to admin panel
Pre-conditions:	1. Admin account is active and properly configured. 2. Unauthorized user attempts to access admin panel.
Task Sequence:	1. Unauthorized user attempts to log in the admin panel. 2. 2. System detects invalid credentials. 3. System logs the event and generates a security alert.
Post-conditions:	Unauthorized user attempt is logged, and a security alert is prompted “Unauthorized access attempt detected.”
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Fails to handle unauthorized access attempt***
Test Case ID:	TC-UC08-03
Use Case ID:	UC-08
Description:	Verify that the system fails to handle unauthorized access attempts to admin panel
Pre-conditions:	1. Admin account is active and properly configured. 2. Unauthorized user attempts to access admin panel. 3. System fails to detecte unauthorized attempt.
Task Sequence:	1. Unauthorized user attempts to login to access admin panel. 2. System fails to detect invalid credentials. 3. Unauthorized user gains access to admin panel.
Post-conditions:	Unauthorized user gained access to admin panel and no security alert was triggered.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Successful Data Privacy Management***
Test Case ID:	TC-UC09-01
Use Case ID:	UC-09
Description:	Verify that the system successfully processes video data locally with end-to-end encryption and manages user data based on content.
Pre-conditions:	1. System is configured to comply with privacy regulations. 2. User has provided consent for data processing.
Task Sequence:	1. System processes video data locally with end-to-end encryption. 2. User data is saved after consent. 3. User data is managed as per request.
Post-conditions:	User data is processed and managed in accordance with privacy regulations.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

Test Case Title:	***Failure in Data Privacy Management***
Test Case ID:	TC-UC09-02
Use Case ID:	UC-09
Description:	Verify that the system fails to apply privacy settings due to system failure.
Pre-conditions:	1. System is confiquired to comply with privacy regulations. 2. System failure occurs during data processing.
Task Sequence:	1. System attempts to process data locally with end-to-end encryption. 2. System fails to apply privacy settings. 3. Data is processed without encryption.
Post-conditions:	User data is not processed and managed in accordance with privacy regulations.
Tested by:	Bisma Azeem
Result:	Pass/Fail
Author:	Bisma Azeem

END