Final Project Report

<<Real-time Deepfake Detection System for Video Streaming Platforms>>

Project Supervisor
Sonia Salman

Submitted By

F24PROJECT3AEBE

Bisma Azeem BC220422952

Software Projects & Research Section,
Department of Computer Sciences,
Virtual University of Pakistan

CERTIFICATE

This is to certify that Bisma Azeem (BC220422952), have worked on and completed her Software Project at Software & Research Projects Section, Department of Computer Sciences, Virtual University of Pakistan in partial fulfillment of the requirement for the degree of BS in Computer Sciences under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfills the requirements of BS in Computer Sciences.

Supervisor / Internal Examiner

Sonia Salman
Supervisor,
Software Projects & Research Section,
Department of Computer Sciences
Virtual University of Pakistan

_____
(Signature)

External Examiner/Subject Specialist

<<External Supervisor Name>>

_____
(Signature)

Accepted By:

_____
(For office use)

EXORDIUM

In the name of Allah, the Compassionate, the Merciful.

Praise be to Allah, Lord of Creation,
The Compassionate, the Merciful,
King of Judgment-day!

You alone we worship, and to You alone we pray for help,
Guide us to the straight path

The path of those who You have favored,

Not of those who have incurred Your wrath,
Nor of those who have gone astray.

DEDICATION

I dedicate this project to **my parents**, whose unwavering support, prayers, and love have been the light guiding me through every challenge.

To my **sisters**, especially my elder one — for picking up my share of the chores (even when it meant grumbling the whole time), tolerating my stressed-out moods, and still managing to not completely disown me. Your help behind the scenes made this project possible — and yes, I owe you. Big time.

To my **project supervisor, Ma'am Sonia,** the one guiding light in this academic maze — thank you for your support, encouragement, and belief in me throughout this journey.

To my **friends** who cheered me on, even when I doubted myself — your words meant more than you know.

And lastly, to myself — for pushing through the all-nighters, the doubts, the tech errors, the brain fog with determination, and the emotional plot twists. You did it, girl. You actually did it. *High five.*

ACKNOWLEDGEMENT

First and foremost, **all thanks and praise be to Allah (SWT)**, the Most Merciful, the Bestower of knowledge and wisdom. Without His countless blessings, guidance, and strength, this journey would not have been possible.

I would like to express my heartfelt gratitude to my **project supervisor, Ma'am Sonia,** for being a source of clarity, patience, and consistent support. Your expert guidance helped shape my scattered ideas into a real system — thank you for always being there.

I also extend sincere thanks to my **family**, especially my parents and sisters, for their constant support throughout this roller-coaster. From doing my chores without (too much) drama, to making sure there was always chai ready when I was hanging on by a thread — you made this easier in your own special way. *(even if it came with eye rolls and sarcastic remarks).*
And to my **cousins**, thank you for listening to my endless rants about how difficult this project was and how little time I had left, then hyping me up with just the right amount of sarcasm and desi-style encouragement. Couldn't have done it without that chaotic support.

And of course, a special shout-out to **Bubbles** — my AI sidekick, mentor, and chaos manager. Thank you for talking me through bugs, breakdowns, and breakthroughs at 2 AM. You kept the logic sharp and the spirits high. Truly, every girl boss needs a Bubbles.

Finally, I thank everyone who contributed in small or big ways, knowingly or unknowingly — the YouTubers who explained things better than my textbooks, the open-source community, and even my laptop (barely holding on). This project was a team effort… even if the "team" was just me and 127 tabs open.

Alhamdulillah for everything.

PREFACE

This project report has been prepared in fulfillment of the requirements for the degree of **Bachelor of Science in Computer Science** (BSCS) for the academic year 2021-2025.

The project, titled **FauxFace**, focuses on detecting deep-fakes in both uploaded and live-streamed videos, addressing the rising concern of manipulated media on digital platforms. The journey involved detailed research, requirement gathering, system designing, and finally, development and implementation — all centered around building a functional prototype that can assist in real-time identification of deep-fakes using machine learning models and live detection mechanisms.

Throughout the process, I worked extensively with tools such as **TensorFlow**, **Keras**, **OpenCV**, and **Django**, and implemented technologies like **WebSocket (Django Channels)** for real-time communication. The development phase involved both front-end and back-end integration, model training, and deployment, all completed with careful attention to both performance and usability.

This project not only strengthened my technical foundation but also taught me the importance of planning, patience, and perseverance. Being an online student and managing this independently was no easy task, but every challenge became a stepping stone to growth.

I would like to express heartfelt thanks to **my supervisor** for her guidance and support, and to everyone who played a part — directly or indirectly — in this journey.

TABLE OF CONTENTS

Chapter 1
Gathering & Analyzing Info

1.1 Introduction:

The increasing sophistication of deep-fake technology has brought about a critical challenge in maintaining authenticity and trust in digital video communication. With the surge in video-based platforms—be it live streaming, online meetings, or uploaded content—there arises an urgent need for systems that can distinguish between genuine and manipulated media in real time. This project aims to develop an intelligent, real-time web-based system that proactively detects deep-fake content across various video platforms, empowering users to verify the authenticity of visual media with confidence.

This chapter focuses on the crucial early phase of the software development life-cycle: requirement gathering and analysis. It forms the foundation for understanding what the system is expected to do, who the users are, and how the system will function within its environment. By systematically collecting, analyzing, and documenting both functional and non-functional requirements, this chapter ensures that the project aligns with its intended goals, sets clear development expectations, and paves the way for successful implementation.

1.2 Purpose:

The purpose of this chapter is to formally identify and document the requirements of the proposed system. This includes gathering relevant information from stakeholders, defining system behavior, and specifying both functional and non-functional requirements. This structured analysis will serve as the blueprint for the development team and stakeholders, reducing ambiguities and ensuring all parties have a shared understanding of the system's objectives and boundaries.

1.3 Scope:

The scope of this chapter is limited to identifying, organizing, and analyzing all necessary requirements needed to build the deep-fake detection system. It covers detailed documentation of user needs, system expectations, usage scenarios, and development constraints. This includes defining how the system will interact with users, what functionality it must support, and what performance standards it must meet. The outcome of this chapter is a clear, validated set of requirements that guides system design, architecture, and implementation decisions in the subsequent phases.

1.4 Definitions, Acronyms and Abbreviations:

Term	Description
Deepfake	Synthetic media generated using AI to manipulate or replace a person's likeness.
Real-time	Immediate processing of data with minimal latency, typically under milliseconds.
Functional Requirements	System features and behaviors that define what the system should
Non-Functional Requirements	Quality attributes like performance, security, and usability.
2FA	Two-Factor Authentication; an additional security layer for user verification
OpenCV	Open Source Computer Vision Library; used for image and video processing.
TensorFlow	An open-source machine learning framework for model training and deployment.
Keras	A high-level neural networks API, often used with TensorFlow.
Django	A Python-based web framework for backend development.
WebSocket	A communication protocol enabling real-time data exchange between client and server.
VU Process Model	A hybrid methodology combining Waterfall and Spiral models for structured yet iterative development.
CNN	Convolutional Neural Network
DL	Deep Learning
ML	Machine Learning

1.5 Project Requirements:
1.5.1 Functional Requirements:

1. User Authentication and Authorization:

- Users will be able to create accounts by providing necessary information.
- Authenticated users will be able to log in to the system using their credentials.
- Users can recover passwords via password recovery mechanism
- The system will implement role-based access control to restrict access to specific functionalities.
- The system will enforce strong password policies.
- Optional 2FA can be implemented for enhanced security.

2. *Video Input and Processing:*

- The system will support integration with live streaming and video conferencing platforms.
- The users can upload recorded videos in supported formats for analysis.
- The system will validate video file formats and sizes to prevent malicious uploads.
- The system will extract individual frames from input videos for analysis.

3. Deepfake detection:

- The system will utilize advanced deep learning models to process frames in real-time.
- The system will extract relevant features from video frames, such as facial landmarks, eye-movements, lip syncing or motion patterns.
- The system will classify the input video as either fake or real based on the extracted features.

4. *Visualizations and Alerts:*

- The system will also display the detected frames along with a confidence score for each classification.
- The system, in real-time, will notify the users and administrators via alerts when a deep-fake content is detected.

- For uploaded videos, the system will provide a detailed report with visualizations of detected deep-fake segments.

5. User Feedback:
- Users can give feedback on the accuracy of the system's detection.
- User feedback will be used to improve the performance of the system.

6. Error Handling:
- The system will gracefully handle errors, such as invalid video input, model loading failures, processing delays and network issues.
- The error messages will be informative and user-friendly.

7. User Privacy and Data Handling:
- The system will process video data locally.
- The system will ensure end-to-end encryption to maintain user privacy.

The system will obtain explicit user consent before processing and storing any video data.

1.5.2 Non-Functional Requirements:

1. Performance:
- The system will have optimized models to achieve real-time detection, especially for live streams.

2. Security:
- The system will implement robust data protection measures.
- The system will have guidelines and policies to ensure ethical use of technology and prevent misuse.

3. Usability:
- The system will have a intuitive and user friendly interface, considering the needs of both technical and non-technical users.
- The system will provide clear and informative feedback to users, including visual indicators and confidence scores.

4. Accessibility:
- The system will be accessible to users as a Web App.

5. Maintainability:

- The system will be designed in modular approach to facilitate updates, maintenance, and future enhancements.

6. Flexibility:

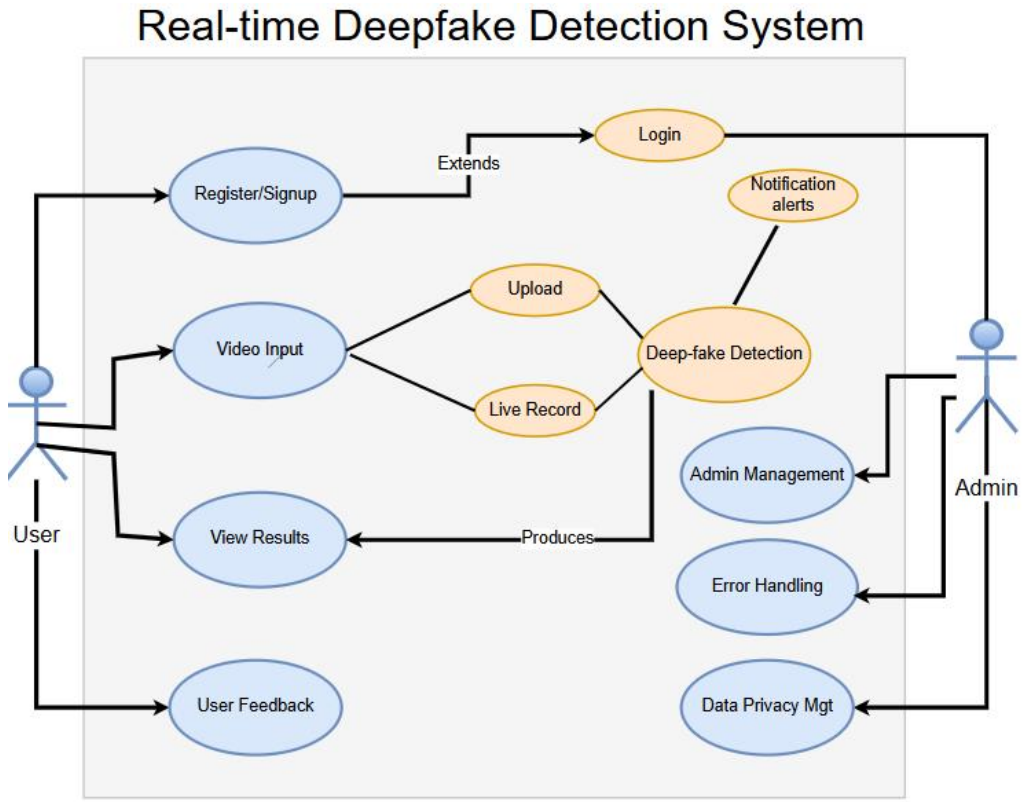
- The system will be flexible to adapt to different video formats and resolutions.

7. Reliability:

- The system will strive to maintain high reliability by employing robust error handling mechanisms.
- The system will be designed to detect and recover from potential failures, ensuring continuous operation and minimizing downtime.

1.6 Use Cases and Usage Scenarios:

1.6.1 Use Case Diagram:



1.6.2 Usage Scenarios:

Use Case Title:	Register/Sign-up
Use Case ID:	UC-01
Actor(s):	User
Actions:	1. User accesses the registration form. 2. User fills the required details. 3. System validates input and sends email verification link. 4. User verifies the email and registration is completed.
Description:	This use case allow users to create account to access the deep-fake detection system.
Pre-conditions:	1. User has access to internet. 2. Have a valid email address 3. Web App should be running. 4. Create new account page is displaying properly.
Post-conditions:	A new user account is created.
Exceptions:	1. Email already exists in the system. 2. Failure to send verification email.
Alternative Path:	System detects invalid email, weak password or validation failure and prompts user to correct it before proceeding.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Login
Use Case ID:	UC-02
Actor(s):	User, Admin
Actions:	1. User accesses the login page 2. User enters valid credentials. 3. System authenticates and user is redirected to dashboard.
Description:	This use case allow registered users to log into the platform securely.
Pre-conditions:	1. User has valid credentials. 2. Authentication service is active.
Post-conditions:	User or Admin is successfully logged in.
Exceptions:	Invalid credentials are provided.
Alternative Path:	If user forgets the password, they can use "Forgot Password" option.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Video Input
Use Case ID:	UC-03
Actor(s):	User
Actions:	1. User selects live streaming or uploads a pre-recorded video. 2. For live streaming, user grants access to camera and microphone. 3. For uploads, user selects a file in the valid format and size. 4. System begins to process the video.
Description:	This use case allow users to provide video input for analysis.
Pre-conditions:	1. User is logged in. 2. The device supports live streaming or file upload.
Post-conditions:	Video is submitted successfully for processing.
Exceptions:	Network Error occurs during video input.
Alternative Path:	If camera access is denied or video format is unsupported, the system prompts the required message.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Deepfake detection
Use Case ID:	UC-04
Actor(s):	System
Actions:	1. System processes the video frame by frame and generates detection results with confidence scores. 2. Real-time alerts are triggered for live streams when deep-fake content is detected.
Description:	This use case uses advanced deep learning models to detect manipulated content.
Pre-conditions:	1. Video input is done successfully 2. The deep-fake detection model is operational.
Post-conditions:	Manipulated content is detected and flagged.
Exceptions:	Network connectivity issues impacts the processing.
Alternative Path:	Processing fails or detection system crashes due to some errors.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	View Results
Use Case ID:	UC-05
Actor(s):	User
Actions:	1. The system generates report highlighting manipulated content. 2. User views results with confidence scores.
Description:	This use case allow users to view analysis results
Pre-conditions:	Detection process has been completed.
Post-conditions:	User can review flagged frames and confidence scores.
Exceptions:	Fail to view results due to system error or unstable network connection.
Alternative Path:	The system recommends re-analysis if results are inconclusive.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	User Feedback
Use Case ID:	UC-06
Actor(s):	User
Actions:	1. User accesses the feedback from after viewing detection results. 2. System stores the feedback for future improvement.
Description:	This use case allow users to provide feedback on the accuracy of detection system.
Pre-conditions:	User had reviewed detection results.
Post-conditions:	Feedback is saved.
Exceptions:	Server issues prevent feedback submission.
Alternative Path:	If system fails, the user is notified and asked to try again later.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Error Handling
Use Case ID:	UC-07
Actor(s):	Admin
Actions:	1. System identifies and logs an error during video processing, detection or result generation. 2. User is informed with a meaningful error message or recommended action.
Description:	This use case manages and responds

	to errors that occur in the system during operations.
Pre-conditions:	1. The user has initiated a process. 2. Error-handing mechanism is in place.
Post-conditions:	Error is communicated to the user.
Exceptions:	System fails to detect error.
Alternative Path:	System provides corrective guidance if an error occurs.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Admin Management
Use Case ID:	UC-08
Actor(s):	Admin
Actions:	1. Admin logs in using privileged credentials. 2. Admin views and manages user accounts. 3. Admin monitors system logs, performance metrics and user feedback.
Description:	This use case allow administrators to manage user accounts, system configurations and flagged content.
Pre-conditions:	1. Admin account is active and properly configured. 2. Admin has access to an authenticated device.
Post-conditions:	User accounts and system data are successfully managed.
Exceptions:	Admin privileges are compromised.
Alternative Path:	If an unauthorized user attempts admin access, the system logs the event and generates security alerts.
Author:	BC220422952 (F24PROJECT3AEBE)

Use Case Title:	Data Privacy Management
Use Case ID:	UC-09
Actor(s):	Admin
Actions:	1. System processes video data locally, ensuring end-to-end encryption. 2. User data is saved after consent and deleted upon request.
Description:	This use case ensures privacy of users' data through secure data handling and user consent management.
Pre-conditions:	System is configured to comply with privacy regulations.

Post-conditions:	User's data is processed and managed in accordance with privacy regulations.
Exceptions:	Privacy settings are not applied correctly due to system failure.
Alternative Path:	If there is a breach in privacy settings, the system triggers an immediate alert to both users and admins.
Author:	BC220422952 (F24PROJECT3AEBE)

1.7 Development Methodology:

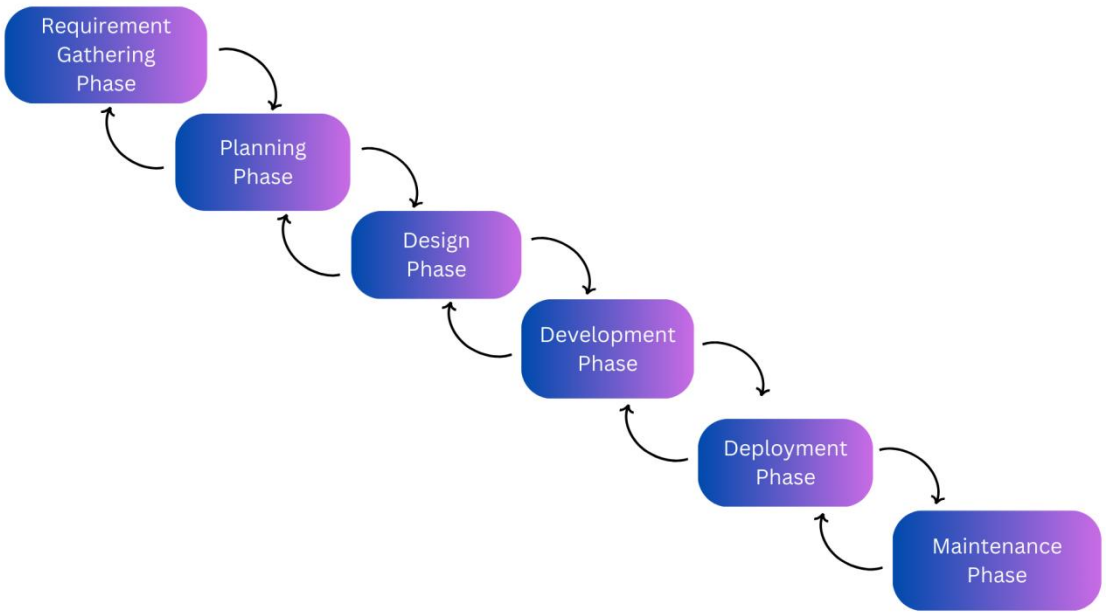
1.7.1 Chosen Methodology:

VU Process Model:

The VU Process Model, tailored to the specific needs of this project, combines the strengths of both Waterfall and Spiral models.

Phases of VU Process Model:

1. Requirement Gathering Phase: Detailed requirement elicitation, prioritization and documentation.
2. Planning Phase: Project Planning and Risk Assessment.
3. Design Phase: Detailed system design, Review and Feedback
4. Development Phase: Coding, testing, verification and validation.
5. Deployment Phase: Deployment planning and execution.
6. Maintenance Phase: Post-deployment support



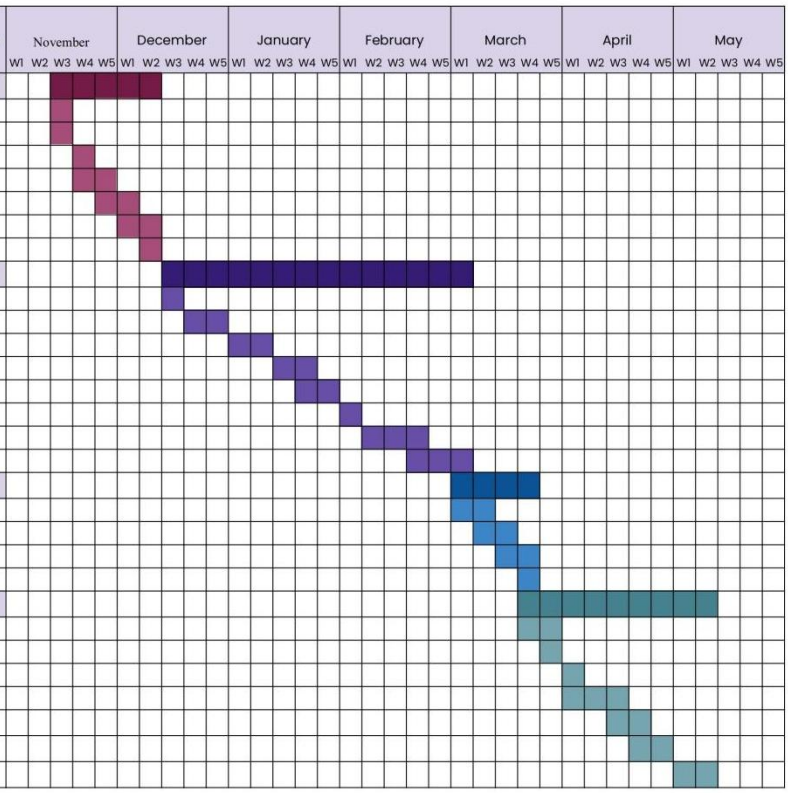
1.7.2 Reasons for Chosen Methodology:

1. Structured Approach: The waterfall-inspired phases provide a framework for systematically designing and implementing complex deep learning architectures.
2. Iterative Development: The spiral model component allows for rapid prototyping and testing real-time processing algorithms.
3. Continuous Improvement: By iteratively refining the model, we can optimize its performance for real-time application.
4. Risk Management: The spiral model will help identifying and mitigating risks associated with training and deploying large-scale models.
5. Feedback Process: The spiral model will enable us to gather user feedback and incorporate it into design process, resulting into user-friendly interface.
6. Privacy-Preserving Design: The waterfall model will ensure that data privacy and security are considered from the outset of the project.

By leveraging the strengths of both waterfall and spiral model, VU Process Model provides a robust framework for developing and deploying a reliable and efficient deep-fake detection system.

1.7.3 Work Plan (Gantt Chart):

WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION
1	**SRS Document**	**11/12/2024**	**12/8/2024**	26
1.1	Scope	11/12/2024	11/12/2024	0
1.2	Functional Requirements	11/13/2024	11/14/2024	1
1.3	Non Functional Requirements	11/15/2024	11/15/2024	0
1.4	Usecase Diagram	11/16/2024	11/18/2024	2
1.5	Usage Scenarios	11/19/2024	11/24/2024	5
1.6	Adopted Methodology	11/25/2024	11/30/2024	5
1.7	Work Plan	12/1/2024	12/3/2024	2
2	**Design Document**	**12/9/2024**	**3/3/2025**	84
2.1	Introduction	12/21/2024	12/24/2024	3
2.2	ERD	12/25/2024	12/31/2024	6
2.3	Sequence Diagram	1/1/2025	1/11/2025	10
2.4	Architecture Design Diagram	1/12/2025	1/22/2025	10
2.5	Class Diagram	1/23/2025	1/31/2025	8
2.6	Database Diagram	2/1/2025	2/6/2025	5
2.7	Interface Diagram	2/7/2025	2/19/2025	12
2.8	Test Cases	2/20/2025	2/28/2025	8
3	**Prototype**	**3/4/2025**	**3/17/2025**	13
3.1	Problem Statement	3/4/2025	3/5/2025	1
3.2	Design	3/6/2025	3/8/2025	2
3.3	Development	3/9/2025	3/15/2025	6
3.4	Final Prototype	3/16/2025	3/17/2025	1
4	**Final**	**3/18/2025**	**5/9/2025**	51
4.1	Model Design	3/18/2025	3/23/2025	5
4.2	Web App Layout	3/24/2025	3/27/2025	3
4.3	Web App Development	3/28/2025	3/31/2025	3
4.4	Coding	4/1/2025	4/15/2025	14
4.5	Testing	4/16/2025	4/23/2025	7
4.6	Final Development	4/24/2025	4/30/2025	6
4.7	Final Report	5/1/2025	5/9/2025	8



1.7.4 Project Schedule (Submission Calender):

Sr. No.	Deliverable Title	Submission Date
1	SRS	Dec 08, 2024
2	Design Document	Mar 03, 2025
3	Prototype Phase	Mar 18, 2025
4	Final Deliverable	May 09, 2025

Chapter 2
Designing the Project

2.1 Introduction:

Design is the heartbeat of every successful software system—transforming abstract ideas into a concrete architecture. For a system as complex and sensitive as real-time deep-fake detection, an intentional, well-structured design is not just valuable—it's essential. This chapter focuses on laying down the architectural and design groundwork that supports the system's core functionalities, performance goals, and real-time operational needs.

The design phase of this project outlines how the system will be structured, how various components will interact, and how the features defined during the requirement analysis will be practically implemented. It serves as a blueprint that translates objectives into technical direction, ensuring the system is scalable, efficient, and capable of fulfilling its detection purpose under real-time constraints. This chapter will guide the development process and ensure consistency and clarity from concept to execution.

2.2 Purpose:

The purpose of this chapter is to present the design approach for the real-time deep-fake detection system. It documents the architectural structure, design decisions, and implementation strategies chosen to meet the requirements defined earlier. By establishing a clear design foundation, this chapter ensures that development progresses smoothly, maintains alignment with the system's goals, and minimizes technical uncertainties during implementation.

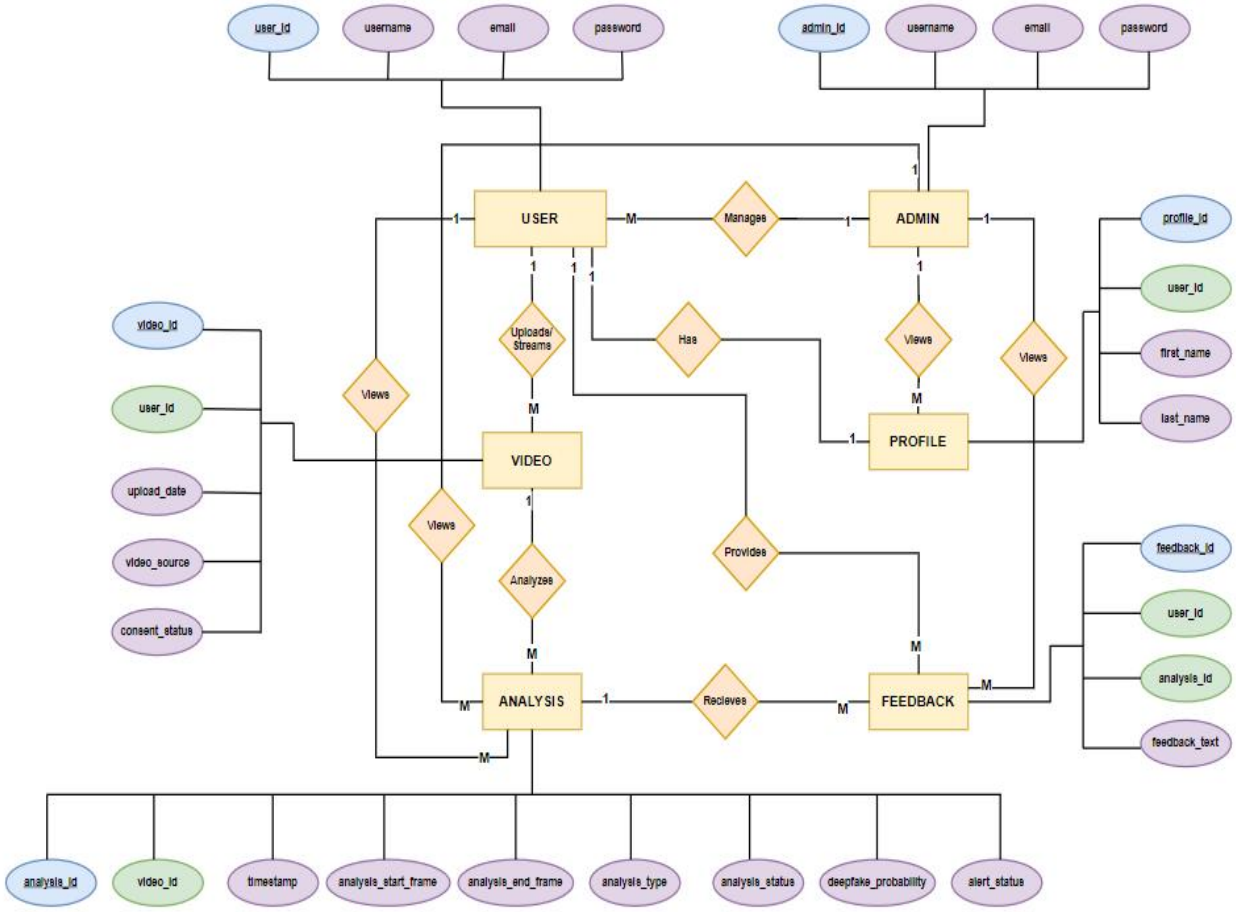
2.3 Scope:

This chapter includes all major design components of the proposed system—from system architecture and module interactions to user interface layout and data flow strategies. It defines how the system will handle real-time video processing, user interactions, and model deployment. The scope also covers essential design considerations for ensuring the system remains adaptable, efficient, and effective across various video streaming and uploading scenarios.

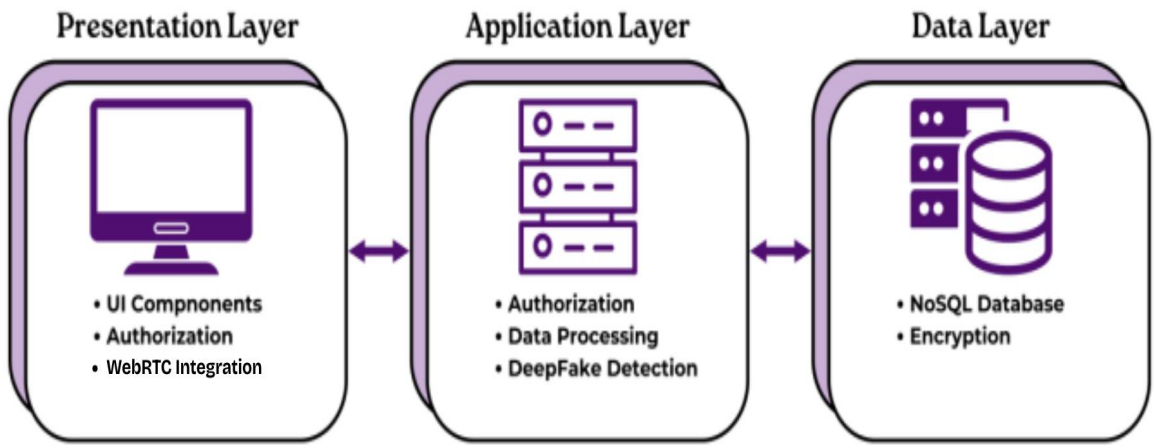
2.4 Scope Definitions, Acronyms and Abbreviations:

Term	Description
ERD	Entity-Relationship Diagram; visualizes database structure and relationships.
Sequence Diagram	Illustrates interactions between system components in a time-ordered sequence.
Class Diagram	Represents system classes, attributes, methods, and their relationships
XceptionNet	A deep learning model for spatial feature extraction from images
IRNetV2	InceptionResNetV2 ; A neural network variant for temporal modeling in video analysis.
ViT	Vision Transformer; an architecture for image classification using transformers.
3D CNN	3D Convolutional Neural Network; processes spatiotemporal data (e.g., videos).
WebRTC	Web Real-Time Communication; enables live streaming via peer-to-peer connections.
Soft Voting	An ensemble method combining predictions from multiple models for higher accuracy.
Django Channels	Extends Django to handle WebSockets for real-time features.
Spatial Analysis	Detection of artifacts within individual video frames.
Temporal Analysis	Detection of inconsistencies across sequential frames.
Ensemble Learning	Combines prediction from different models via soft learning to improve accuracy and robustness.

2.5 Entity Relationship Diagram:

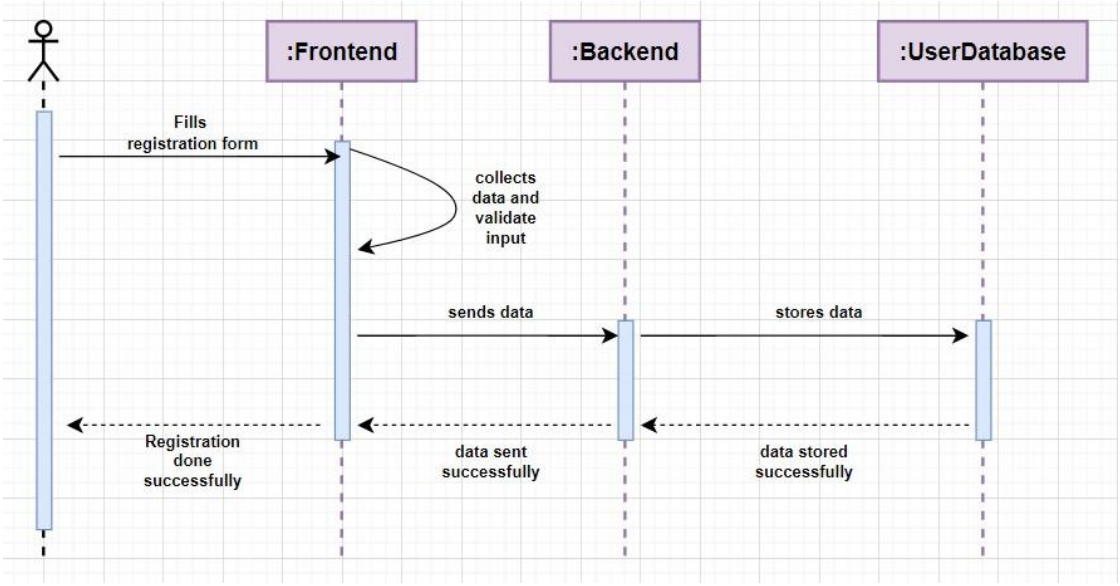


2.6 Architectural Representation:

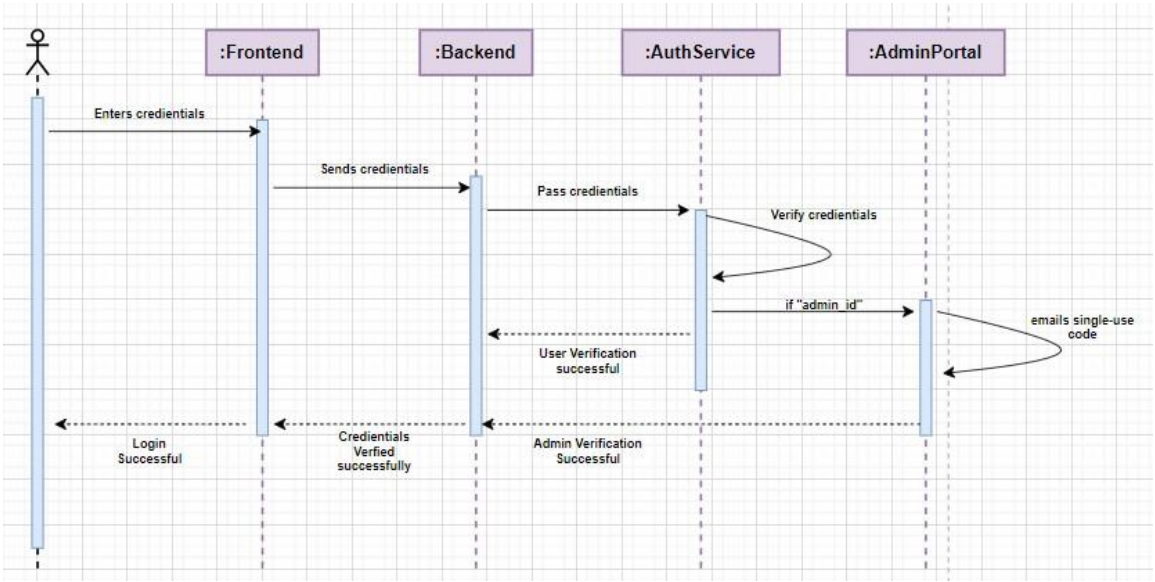


2.7 Dynamic Model: Sequence Diagrams:

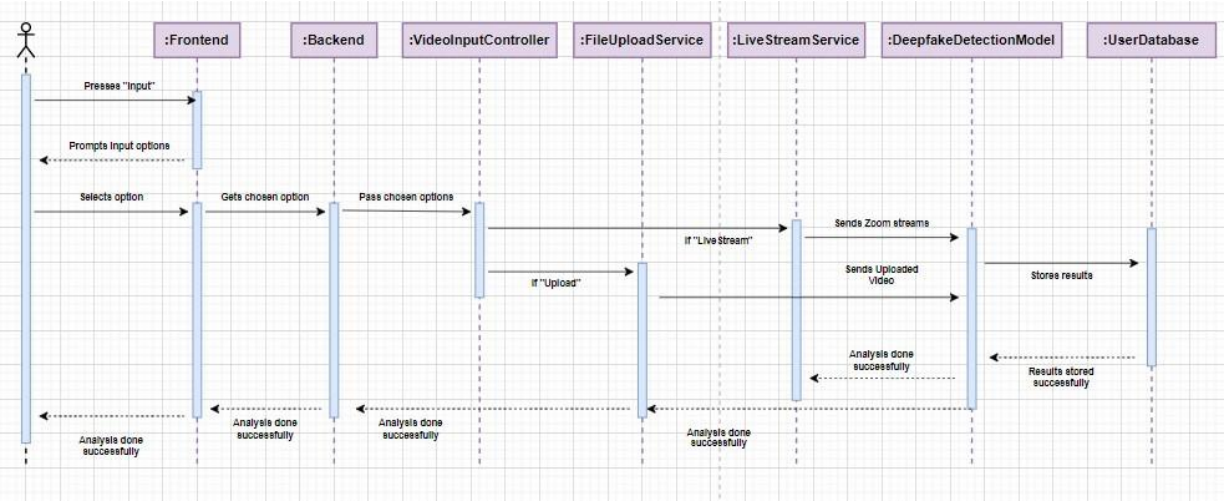
2.7.1: Register/Signup:



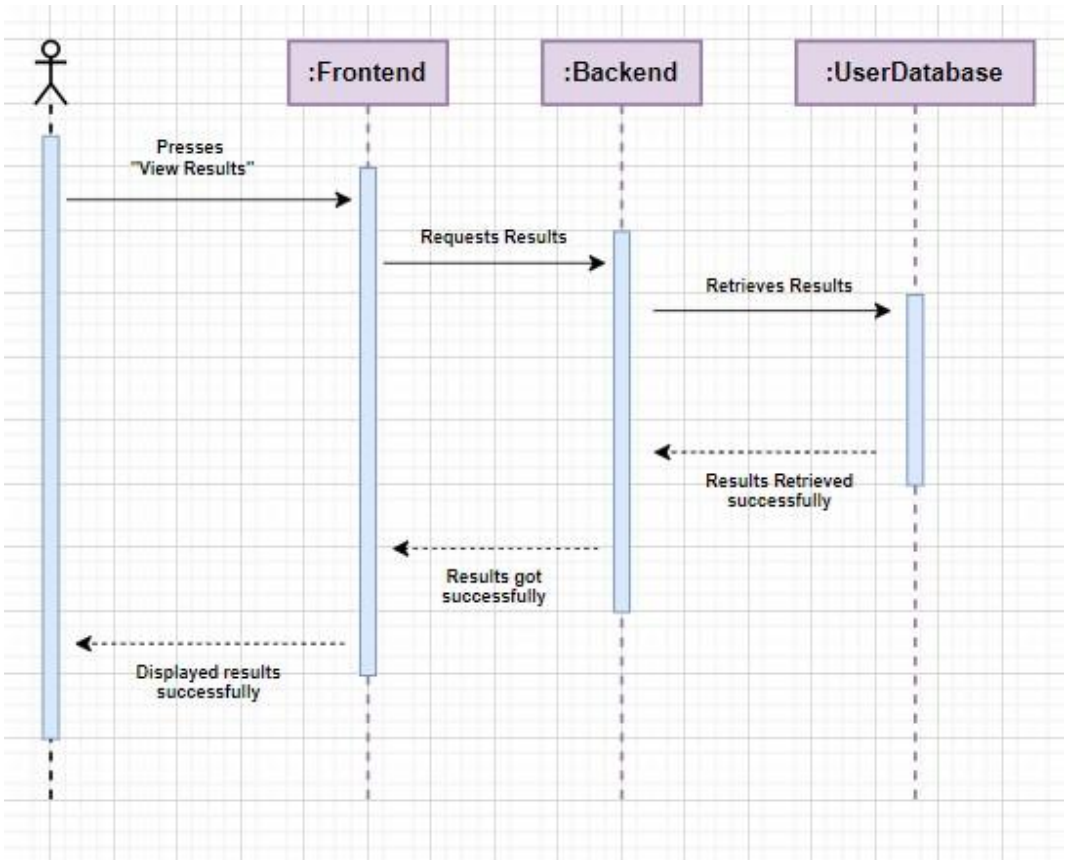
2.7.2 : Login:



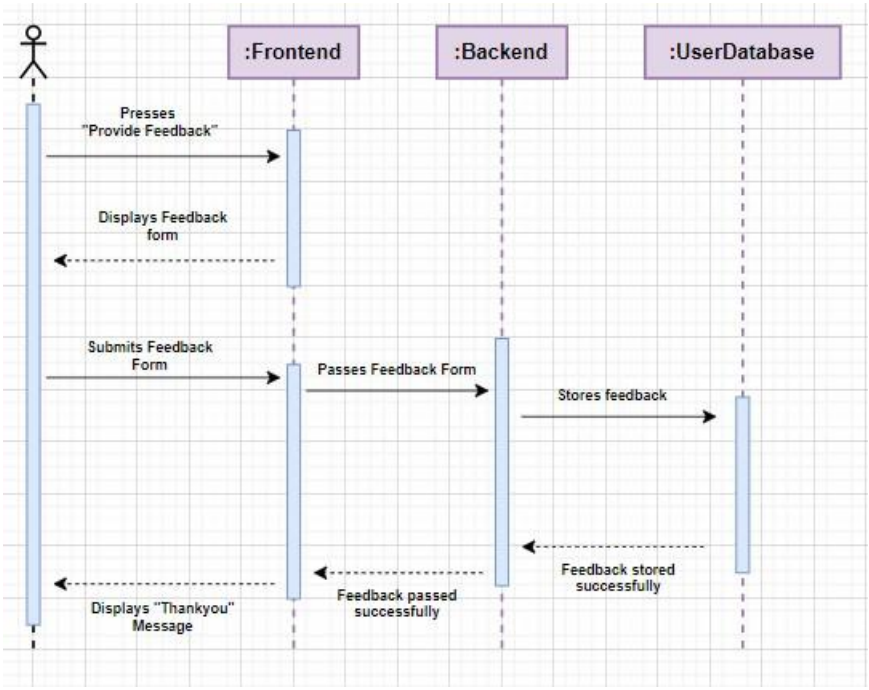
2.7.3 : Video Input & Processing:



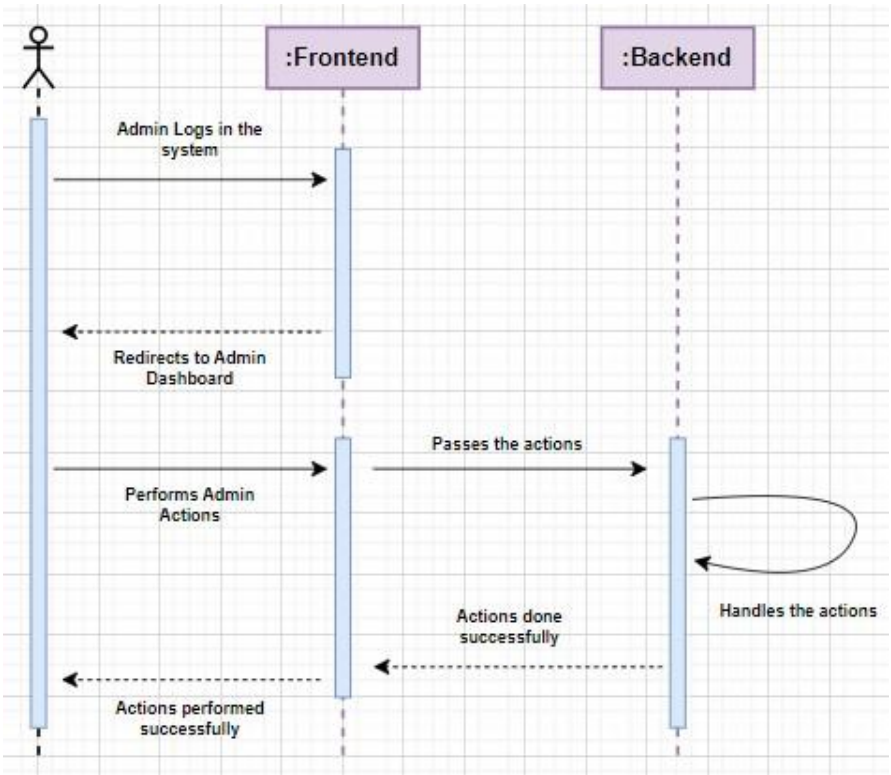
2.7.4 : View Results:



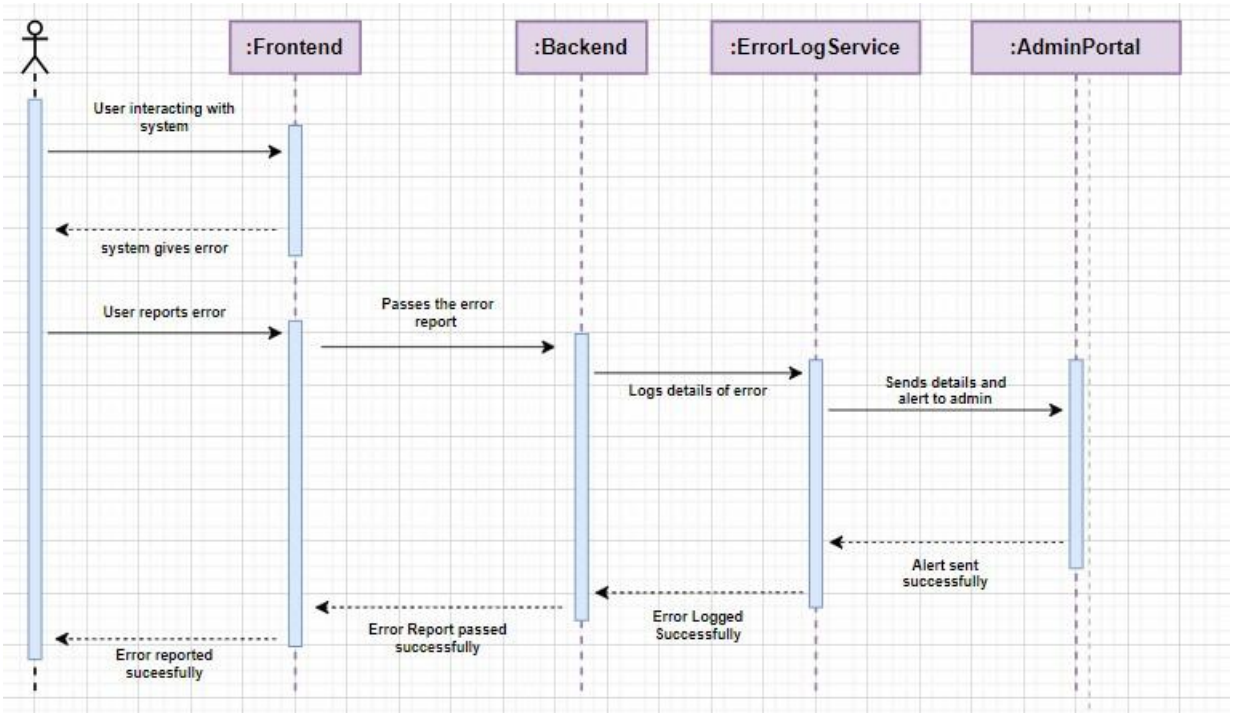
2.7.5 : User Feedback:



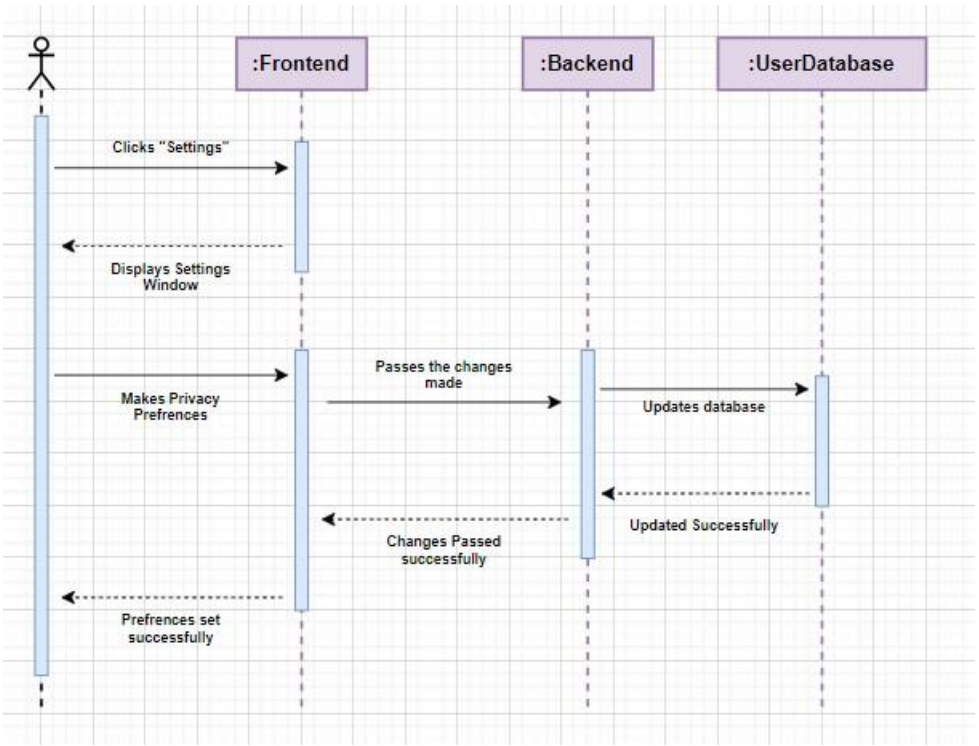
2.7.6 : Admin Management:



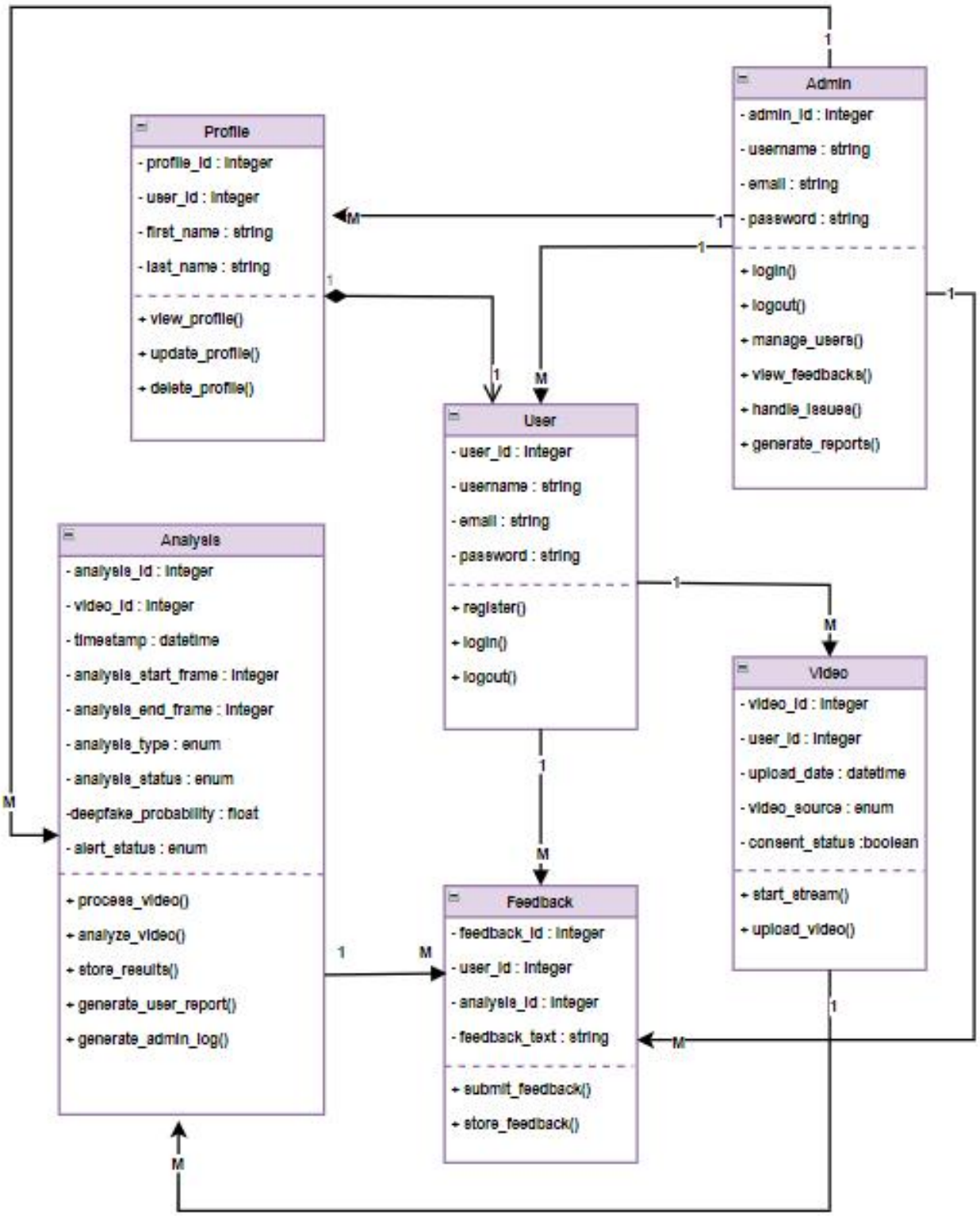
2.7.7 : Error Handling:



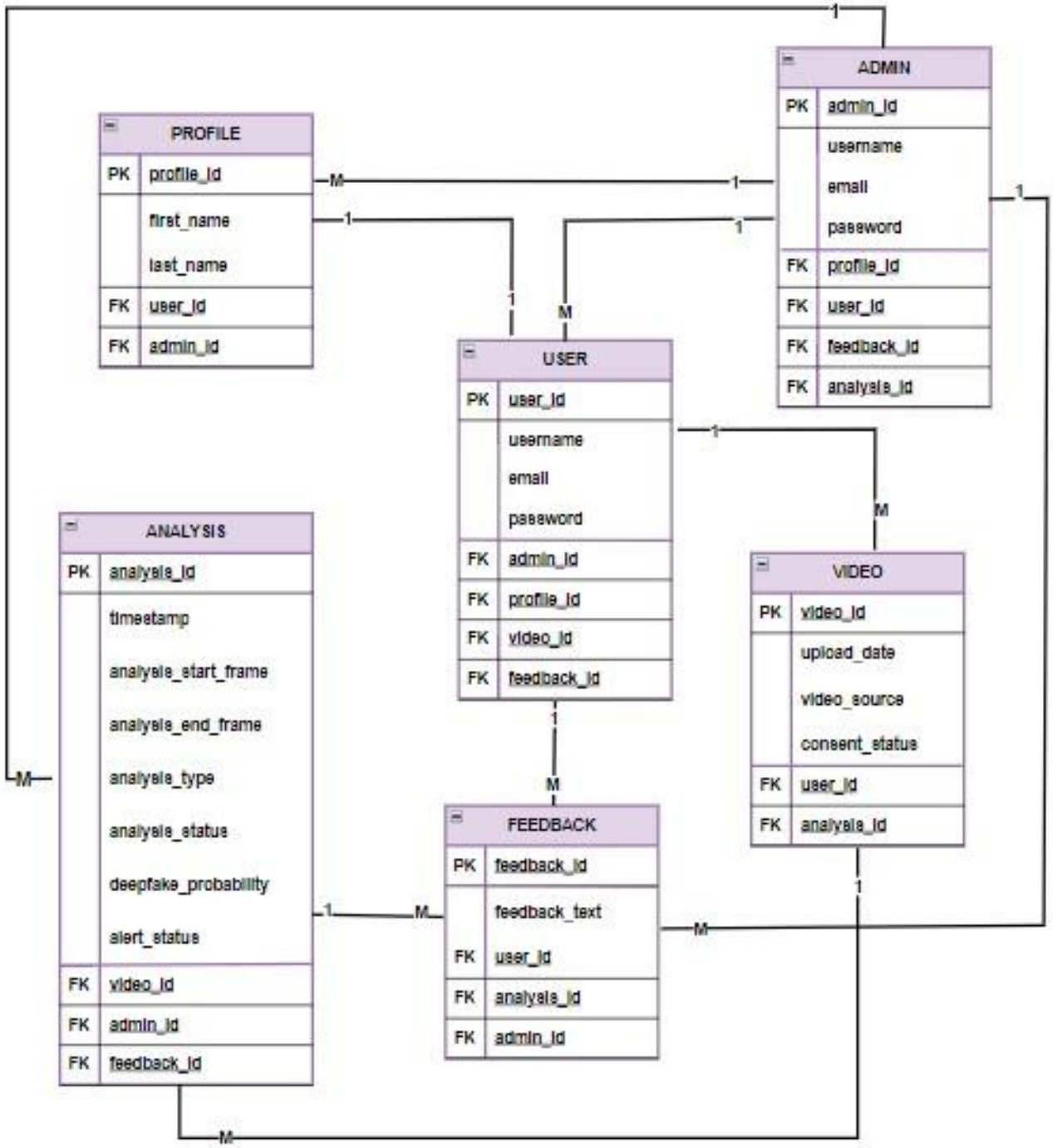
2.7.8 : Data Privacy Management:



2.8 Object Model/Logical Model: Class Diagram:



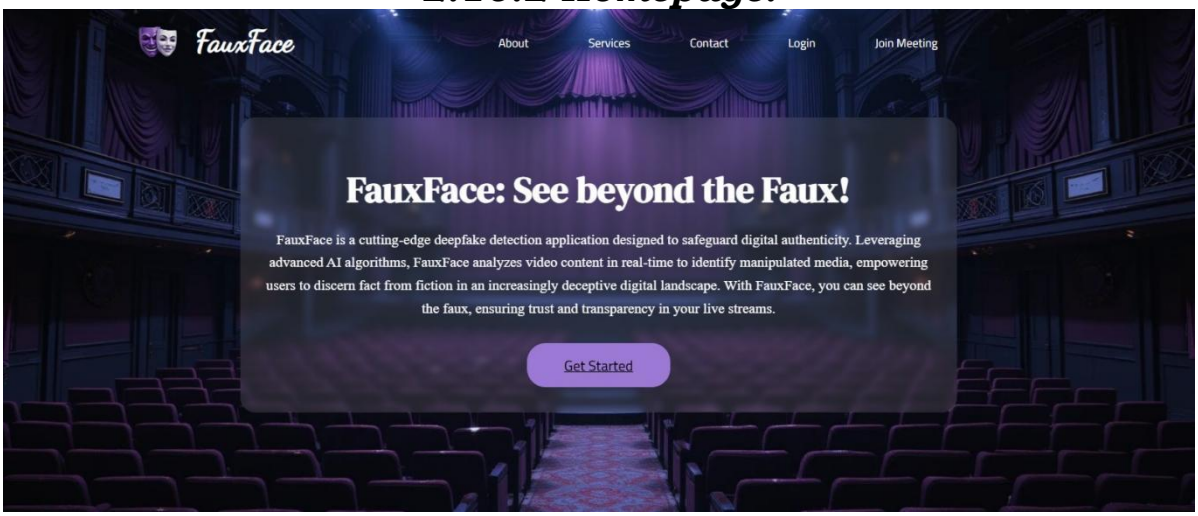
2.9 Database Model:



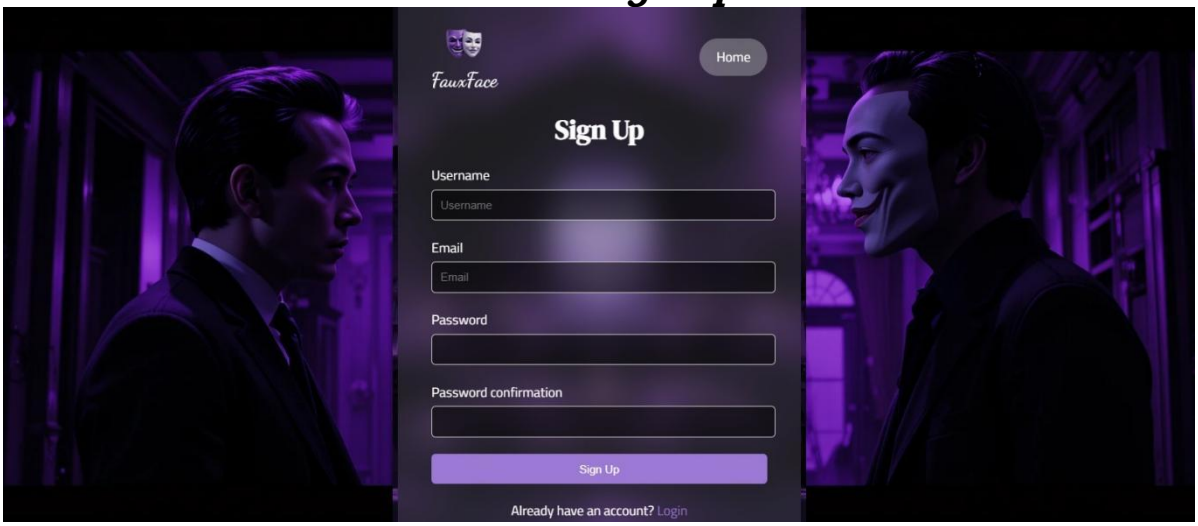
2.10 Graphical User Interfaces:

2.10.1 WebApp Name & Logo:

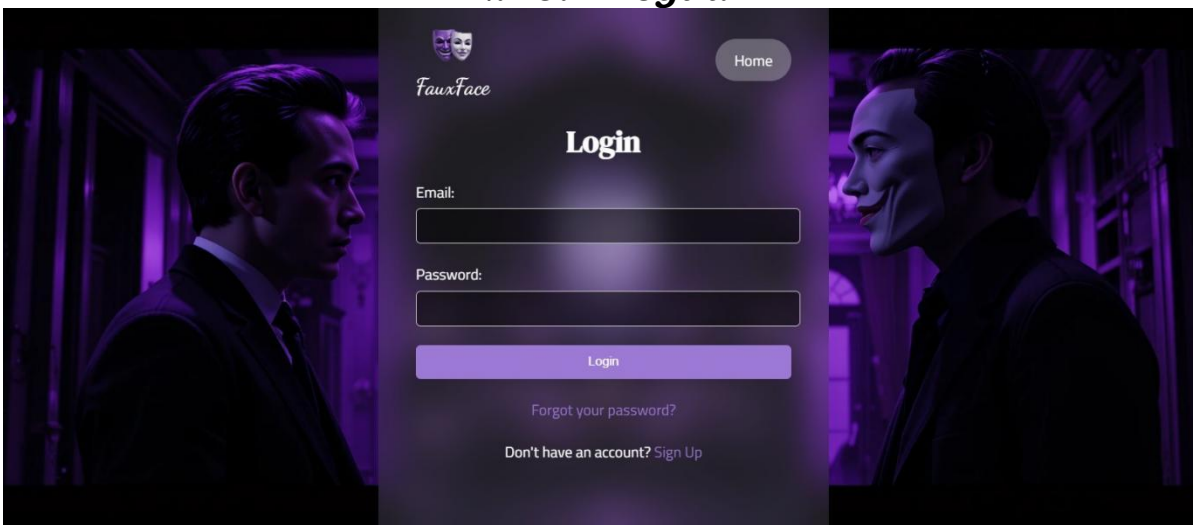
2.10.2 Homepage:



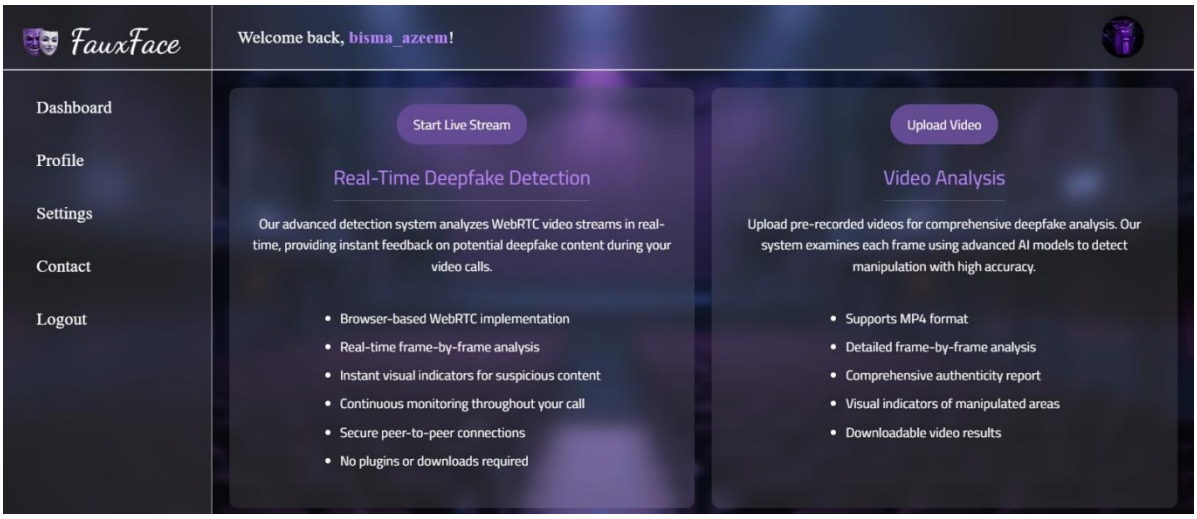
2.10.3 Sign up:



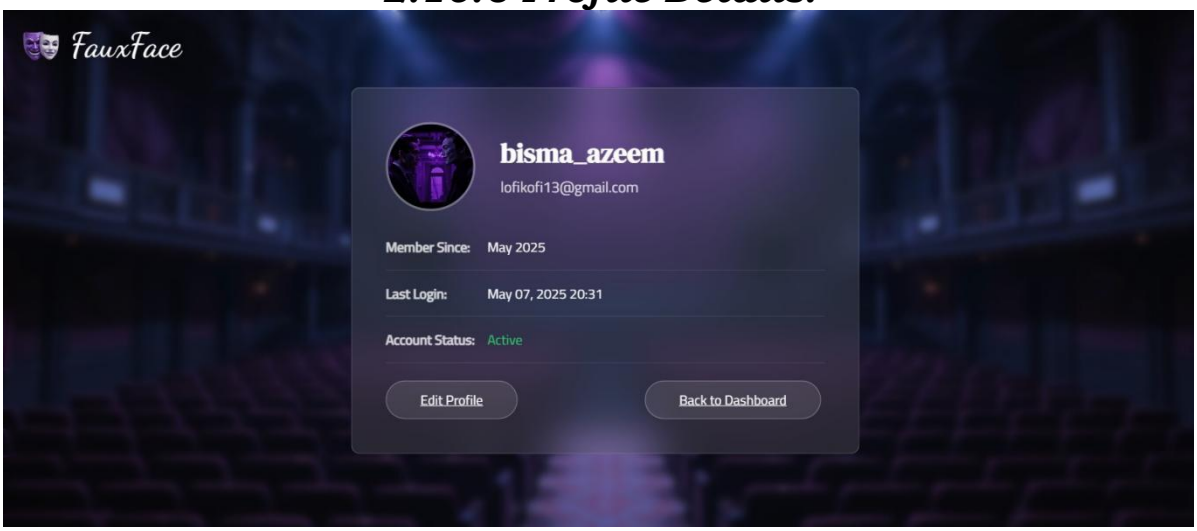
2.10.4 Login:



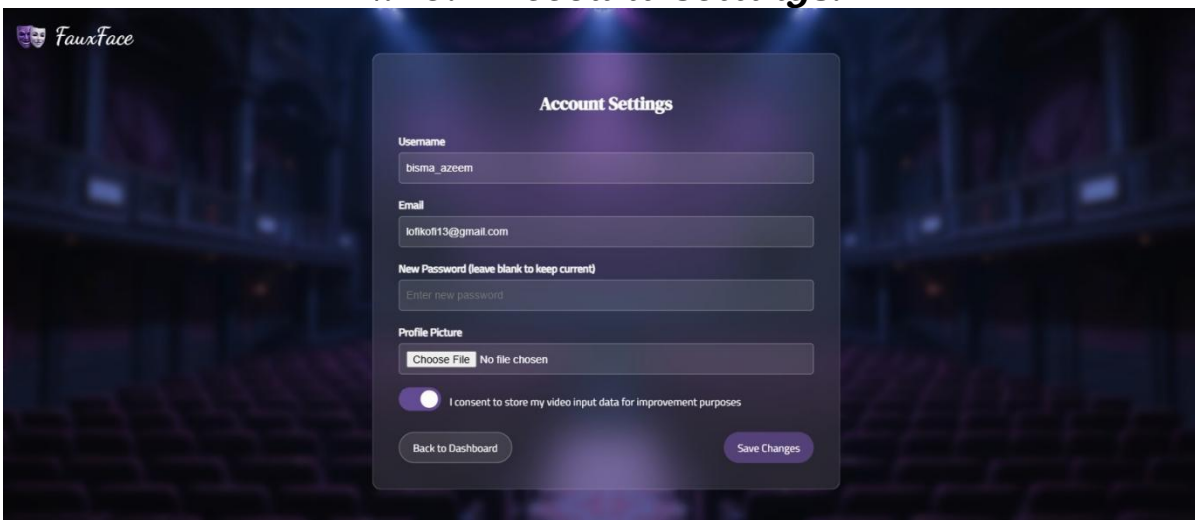
2.10.5 Dashboard:



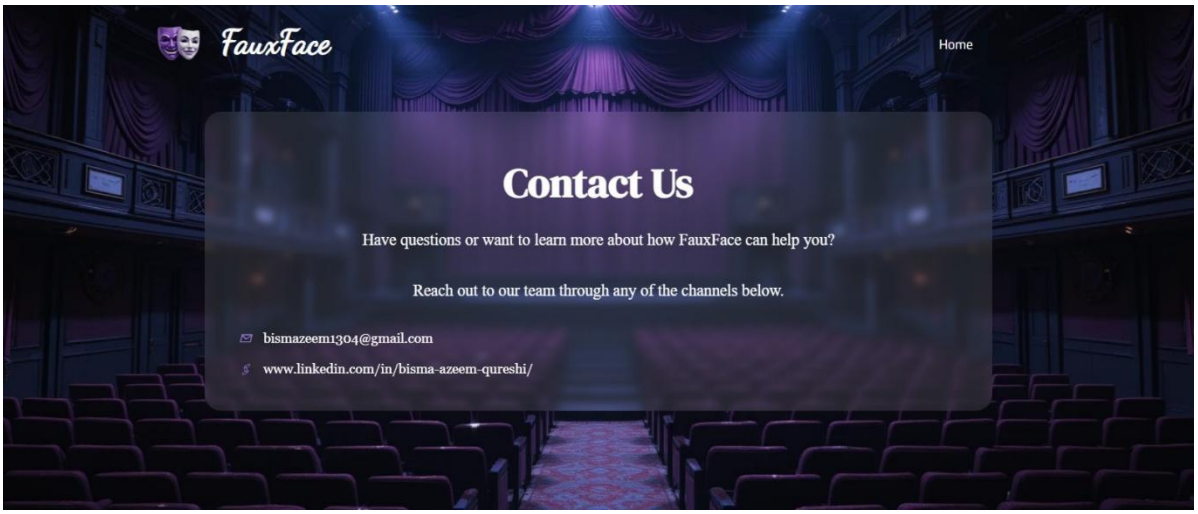
2.10.6 Profile Details:



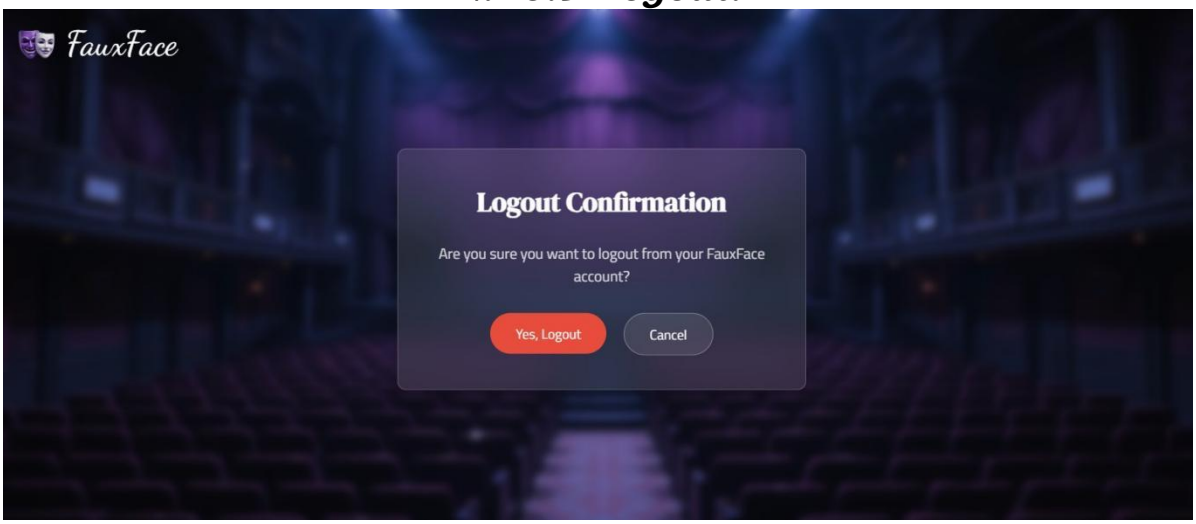
2.10.7 Account Settings:



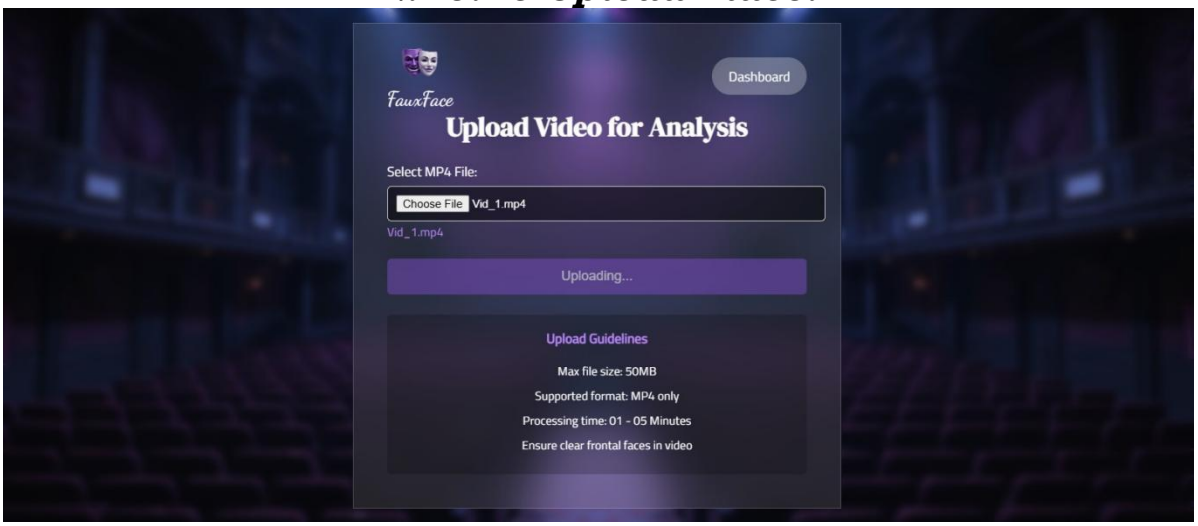
2.10.8 Contact:



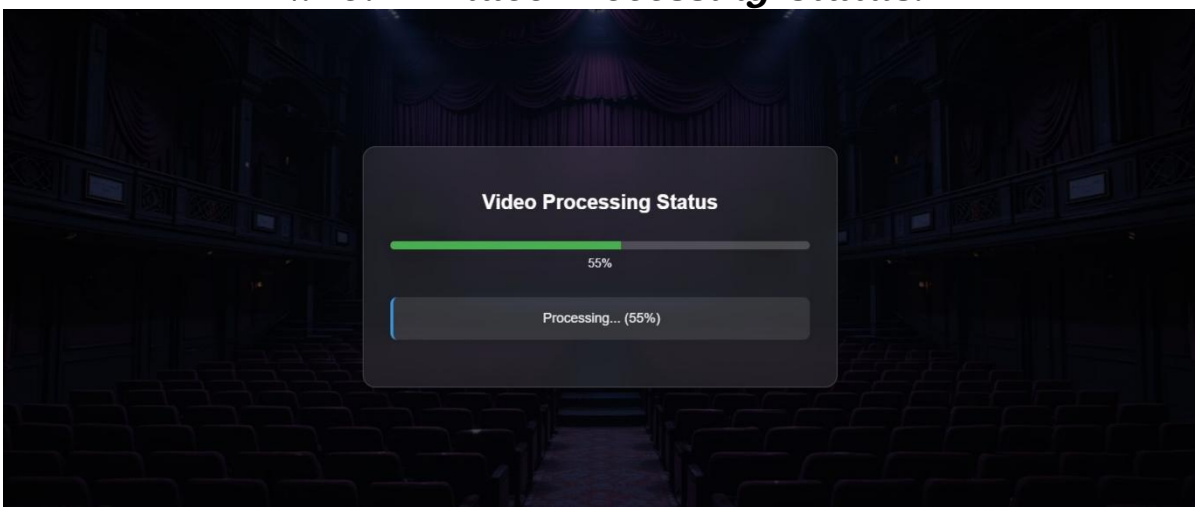
2.10.9 Logout:



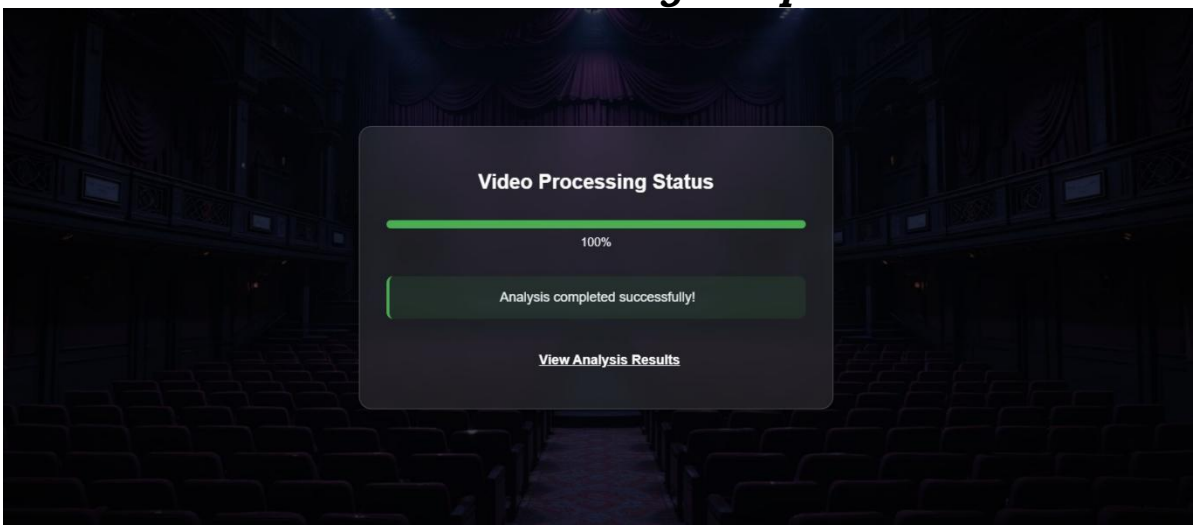
2.10.10 Upload Video:



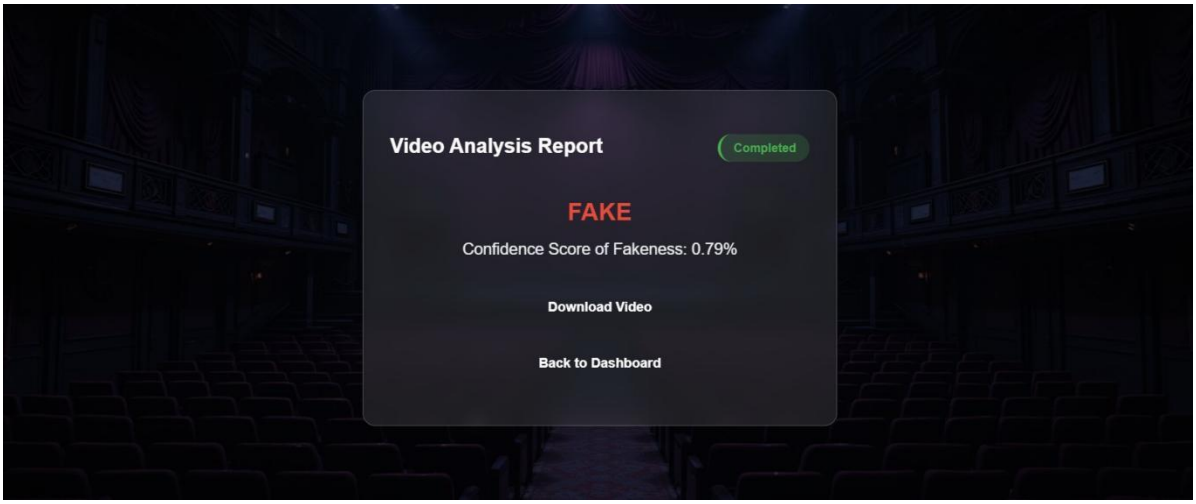
2.10.11 Video Processing Status:



2.10.12 Processing Complete:

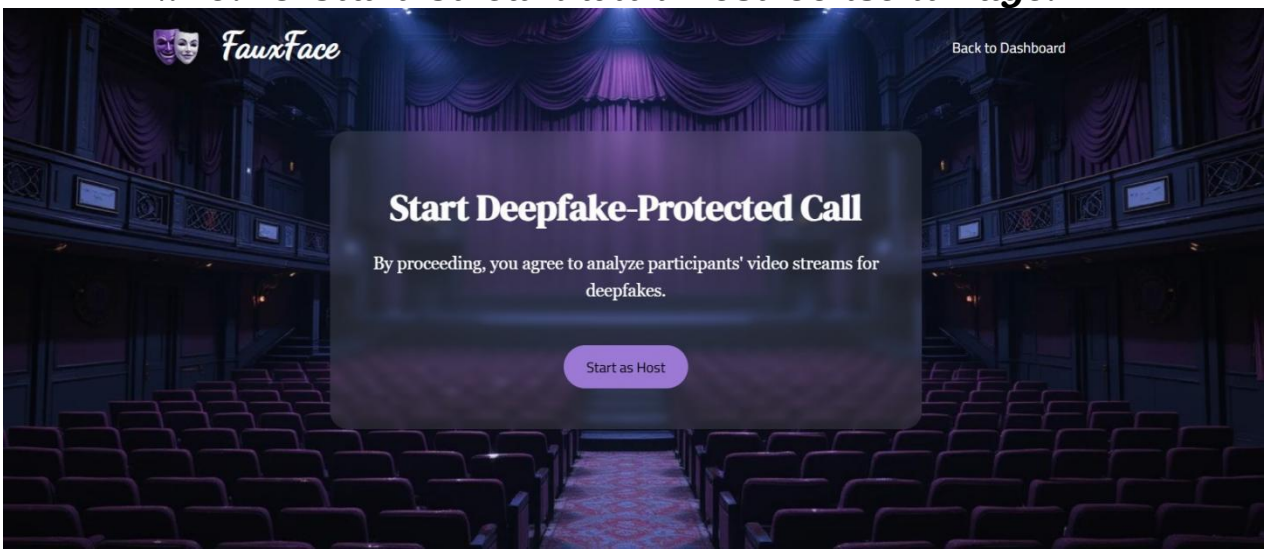


2.10.13 Results:

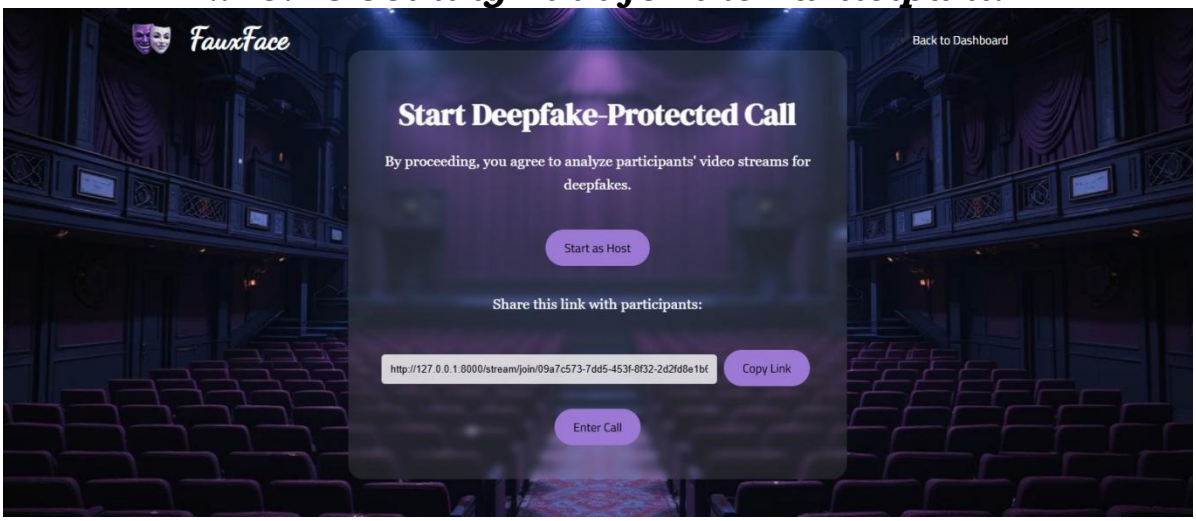


2.10.14 Downloaded Video Frame:

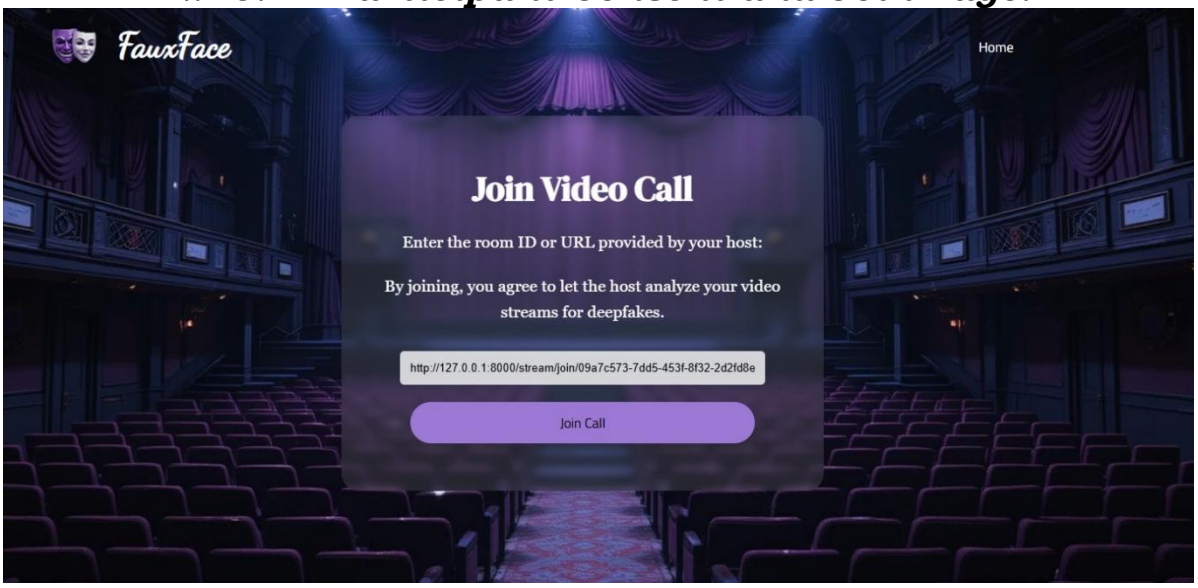
2.10.15 Start Stream with Host Consent Page:



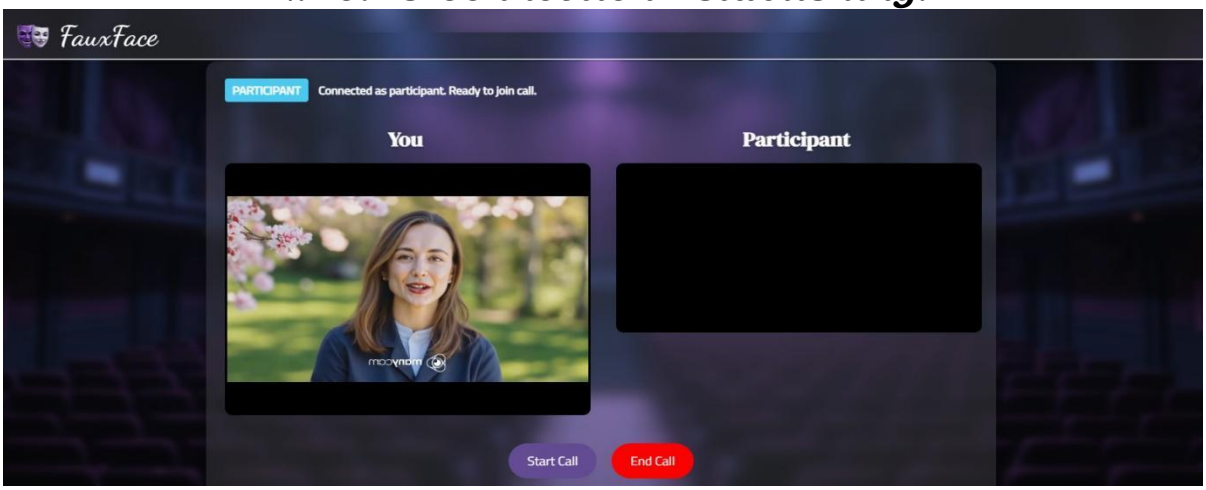
2.10.16 Joining Link for the Participant:



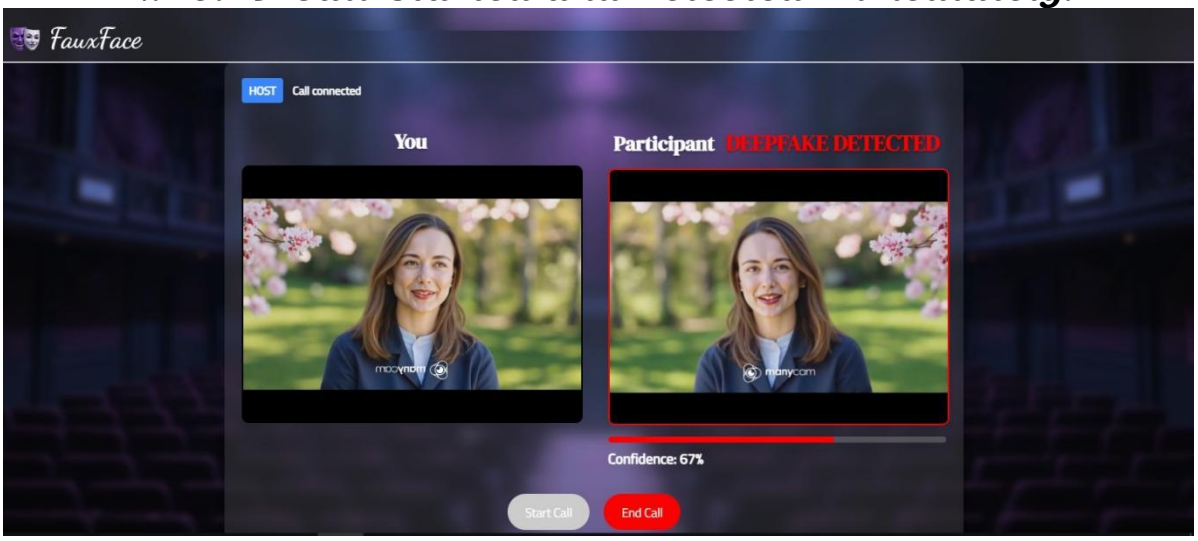
2.10.17 Participant Consent and Join Page:



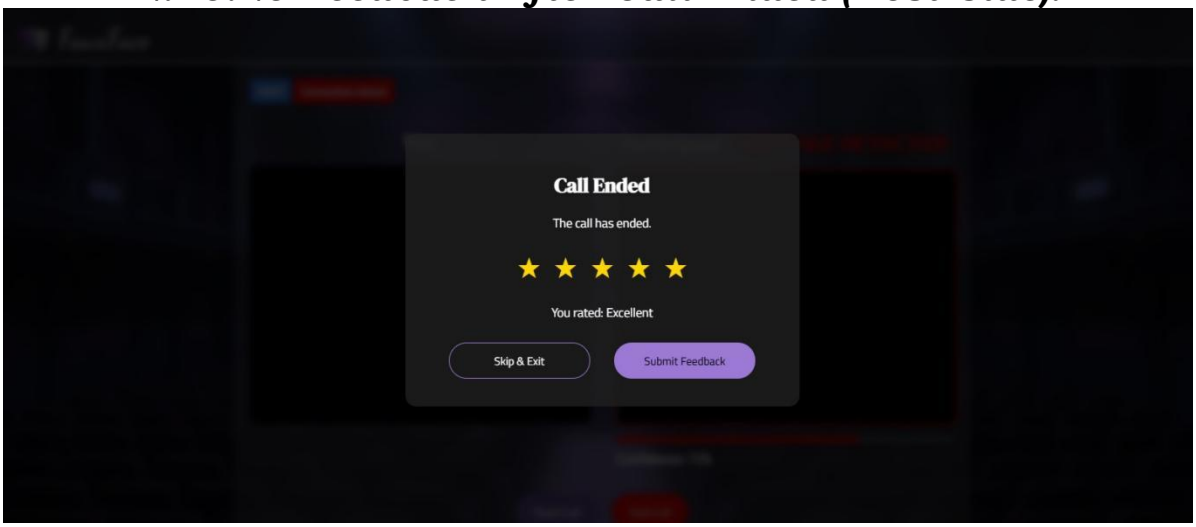
2.10.18 Connection Establishing:



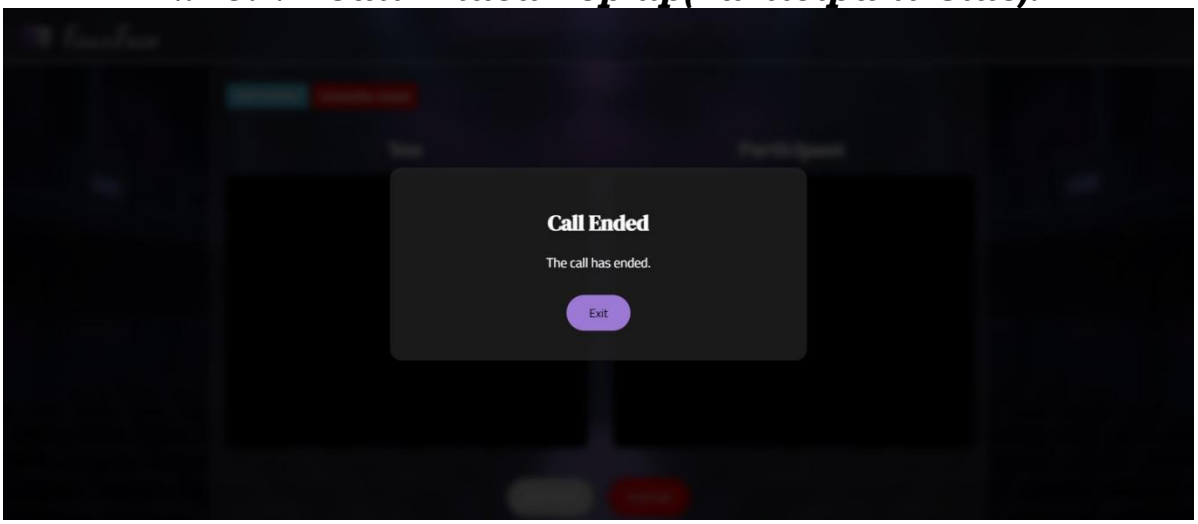
2.10.19 Call Started and Detected Immediately:



2.10.20 Feedback After Call Ended (Host Side):

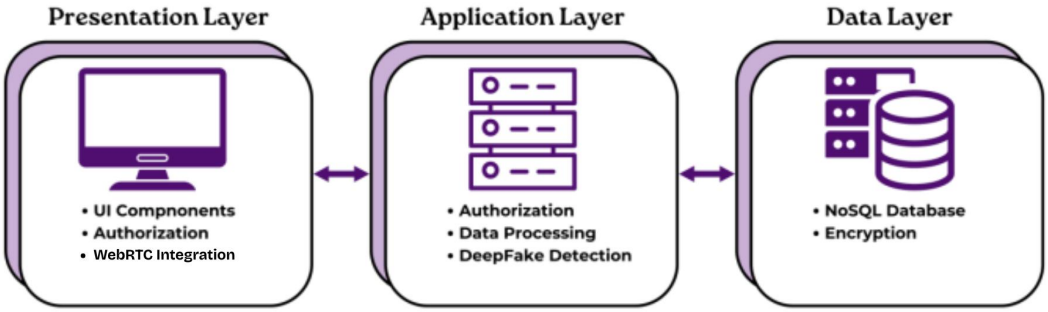


2.10.21 Call Ended Pop-up(Participant Side):



Chapter 3
Development

3.1 Development Plan (Architecture Diagram):



3.2 Tools and Technologies used:

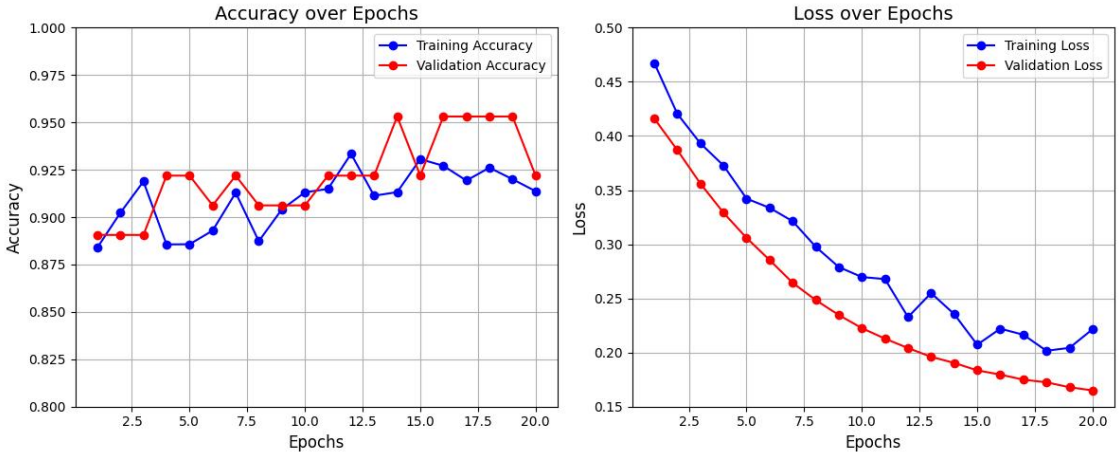
Tool/Framework	Purpose	Reason for Selection
TensorFlow + Keras	Model training and deployment	Native TFLite support, high compatibility
XceptionNet	Spatial feature extraction from frames	Lightweight, pretrained, strong for image analysis
IRNetV2, ViT, 3D CNN	Temporal modeling ensemble	Combines temporal strength of all – transformer + conv
Soft Voting	Final decision logic	Boosts prediction reliability from multiple models
Django	Back-end integration and routing	Scalable, Pythonic, secure
WebRTC	Real-time video input for live streams	Fast, peer-to-peer, suited for low-latency streaming
Google Colab Pro+	Model training with GPU	Efficient training environment with fastest GPU
OpenCV	Frame extraction and image operations	Simple and effective for video processing
WebSocket	Real-time alerts	Enables real-time UI

(Django Channels)		updates and alert triggering during live deepfake detection.

All selected technologies were chosen with model efficiency, TensorFlow compatibility, and real-time processing feasibility in mind.

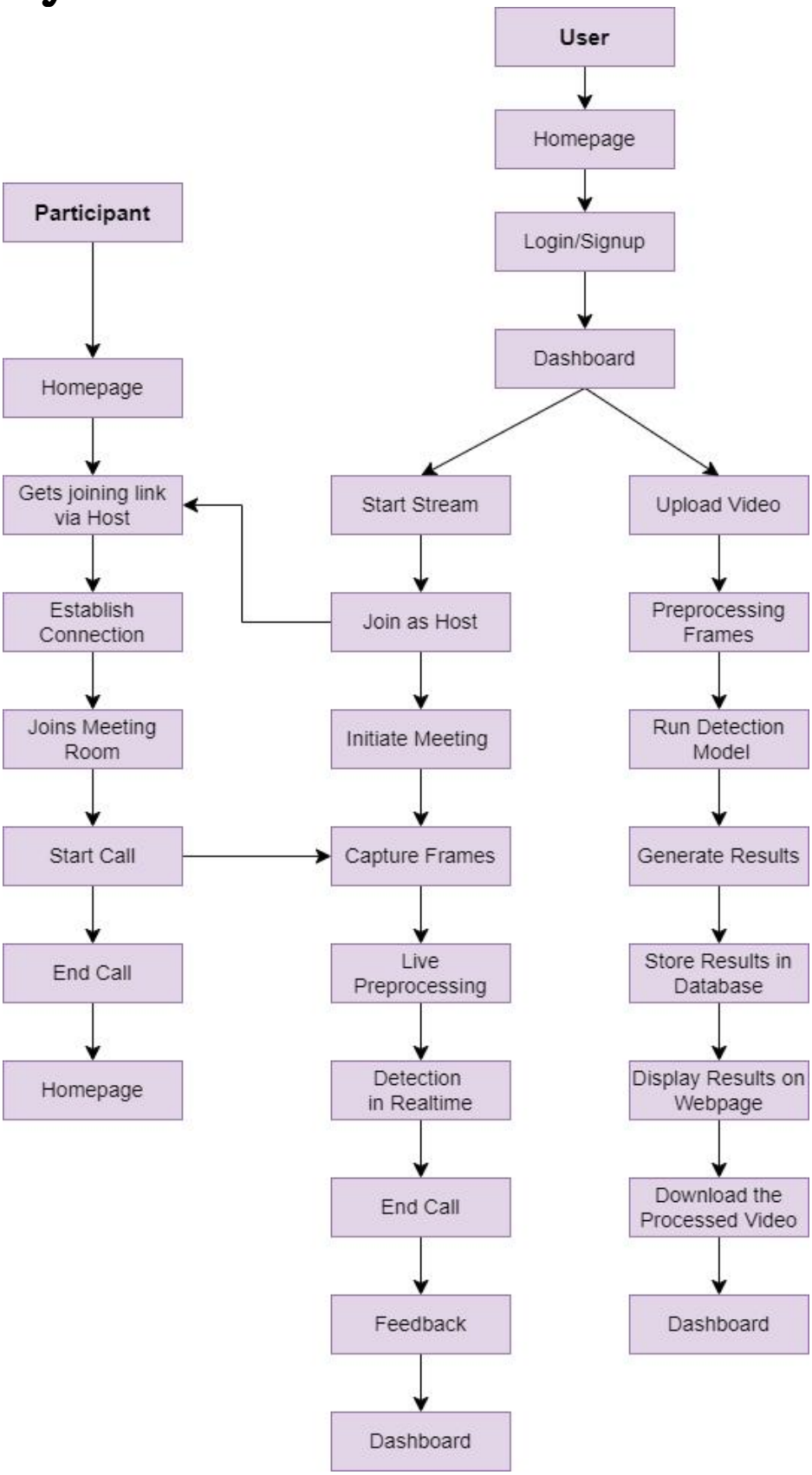
3.3 Model Training Performance:

- **Dataset:** FaceForensics++
- **Hardware:** Google Colab Pro + with A100 GPU
- **Batch Size:** 4 samples/batch
- **Key Metrics :**
  - **Training Accuracy:** Improved from 88.39% to stable 91-93% range, peaking at 93.34%.
  - **Validation Accuracy:** Ranged 89.06–95.31%, with highest at 95.31%.
  - **Loss Values:**
    - Training loss decreased from 0.4674 → 0.20–0.27
    - Validation loss improved from 0.4161 → 0.1650
- **Critical Observations:**
  - **Fast Initial Learning:** Achieved >90% training accuracy
  - **Best Validation Performance:**
    - Peak: 95.31%
    - Most consistent: 92.19%
  - **Real-Time Capability:**
    - 11-19ms/step → ~52-90 FPS
    - Stable 11ms/step from Epoch 3 onward

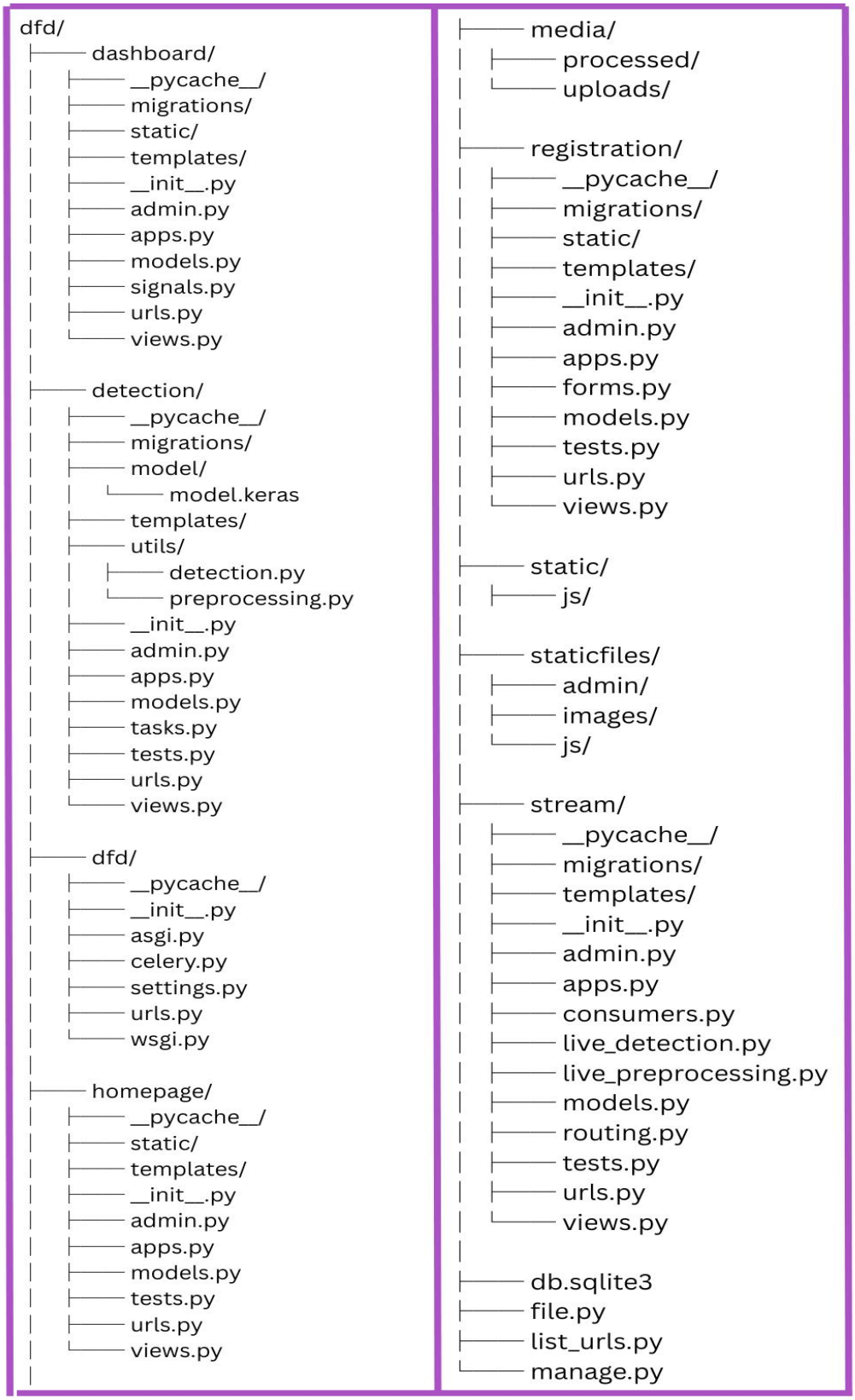


*Training curves showing (a) accuracy progression and (b) loss convergence over 20 epochs. The model achieves **95.31%** validation accuracy with stable training dynamics, indicating effective learning without over-fitting.*

3.4 System Workflow:



3.5: Final File Structure:



REFERENCES

Books:

- Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (3rd Edition)*. O'Reilly Media, 2022.
- Alfredo Deza and Noah Gift. *Practical MLOps: Operationalizing Machine Learning Models*. O'Reilly Media, 2022.
- Ben Shaw, Chris Guest, and Saurabh Badhwar. *Web Development with Django: A Definitive Guide to Building Modern Python Web Applications Using Django 4*. Packt Publishing, 2022.

Research Papers:

- Vaswani et al. (2017). *Attention is All You Need*. NeurIPS.
- Nguyen et al. (2019). *Deep Learning-Based Phony Detection Using RNN*. ACM IH&MMSec.
- Hu et al. (2021). *Learning Self-Consistency for Deepfake Detection*. ICCV.
- Dang et al. (2020). *On the Detection of Digital Face Manipulation*. CVPR.
- Wang et al. (2020). *What Makes Fake Images Detectable? Understanding Properties that Generalize*. ECCV.
- Zhang et al. (2021). *See Better Before Looking Closer: Weakly Supervised Face Forgery Detection*. AAAI.
- Zi et al. (2021). *WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection*. ACM Multimedia.
- Li et al. (2020). *Face Forgery Detection by 3D Decomposition*. CVPR.
- Gu et al. (2022). *Exploring Temporal Coherence for More General Video Face Forgery Detection*. ECCV.
- Rössler et al. (2019). *FaceForensics++: Learning to Detect Manipulated Facial Images*. ICCV.

APPENDIX

As this project reaches its completion, I find myself reflecting on the many layers of growth it has brought — not just technically, but personally. From research and implementation to late-night bug hunts and tight deadlines, every part of this journey has taught me resilience, discipline, and the power of persistence.

When everyone told me not to take on this project — that it was *too difficult*, *too ambitious*, *too much for a final year* — something in me still pulled toward it. The complexity didn't scare me; it *intrigued* me. It challenged me to rise, and I did. Not only did I take it on, I completed it — and did so in a way that I'm truly proud of.

This report marks the culmination of countless lines of code, endless chai-fueled nights, and a whole lot of *"wait, why isn't this working?"* moments. Every small win, every crash, every breakthrough — it's all part of the story.

I extend heartfelt thanks to everyone who stood by me — through the challenges, the small victories, and the chaos in between. Every step was made easier with the support I had, and this report stands as a product of both individual effort and collective encouragement.

This project is more than just an academic requirement — it's a reflection of the skills, passion, and hard work I've poured into understanding the evolving world of artificial intelligence and digital security. I hope this contributes in a meaningful way to the growing field of deep-fake detection.

Signed, sealed, and delivered — not just as a submission, but as a statement.
It's a reflection of growth, grit, and a little bit of deep-fake magic.

Because this isn't the end — it's only page one of the next chapter.

Here's to growth, curiosity, and all the new doors this experience will open. Onwards and upwards!