Project report
on
ZOOM EMOTION TRACKER

Project report submitted in partial fulfilment of the requirement for Software Engineering Lab

Submitted By

G.KUMAR	**[R210564]**
A.GEETHA SREE	**[R210780]**
K.USHARANI	**[R210554]**
S.UJJEEF	**[R210610]**

Under the Esteemed Guidance of

Ms.SUSMITHA ,Assistant Professor

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES(AP IIIT)

R.K Valley, Vempalli, Kadapa(Dist) – 516 330 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING 2025-2026

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES (AP IIIT)

R.K Valley, Vempalli(M), Kadapa(Dist) – 516330 DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2025-2026

CERTIFICATE

This is to certify that the project report entitled “**ZET TOOL**” being submitted by **G.Kumar, A.Geetha Sree, K.UshaRani, S.Ujjeef** under my guidance and supervision and is submitted to **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** in partial fulfilment of requirements for SoftwareEngineering Lab during semester – I of academic year 2025 - 2026 and it has been found worthy of Acceptance According to the requirements of the Department.

Signature of Internal Guide	**Signature of HOD**
Ms.Susmitha	**Dr.Ch.Ratna Kumari**
Assistant Professor	**Assistant Professor**
Department of CSE	**Department of CSE**

Signature of External Examiner

Table of Contents

List Of Figures

WBS DOCUMENT

About Company: -

Insight Zoom provides technical and IT solutions to clients. We specialize in consulting, software development, and system integration services.

Our services include web application development, database management systems, and enterprise-level solutions that ensure efficiency, transparency, and accessibility. We focus on developing reliable, secure, and high-performing applications tailored to client requirements.

Project Name: -ZOOM EMOTION TRACKER (ZET)

Objective (Aim of the Project): -

This project is about virtual meetings.It’s main aim is to build a system that can detect and display emotions of participants during a zoom meeting using real time video analysis.By leveraging facial expression recognition,the system helps presenters,teachers or team leads understand audience engagement better.

Users of the Product: -

- Teachers or hosts
- students
- Administartors

Technologies: -

- **Django,SQLite(SQLAlchemy ORM) –** Database for reports of emotions and user information
- **Express.js** – Server-side framework for building APIs
- **React.js** – Frontend framework for dynamic user interface
- **Node.js** – Backend runtime for executing server logic
- **RESTful APIs,JWT tokens** – For communication between frontend and backend
- **Integration**– Zoom SDK/API
- **Backend** -Python,openCV, Flask,TenserFlow
- **Emotion bot -**Emotion detection

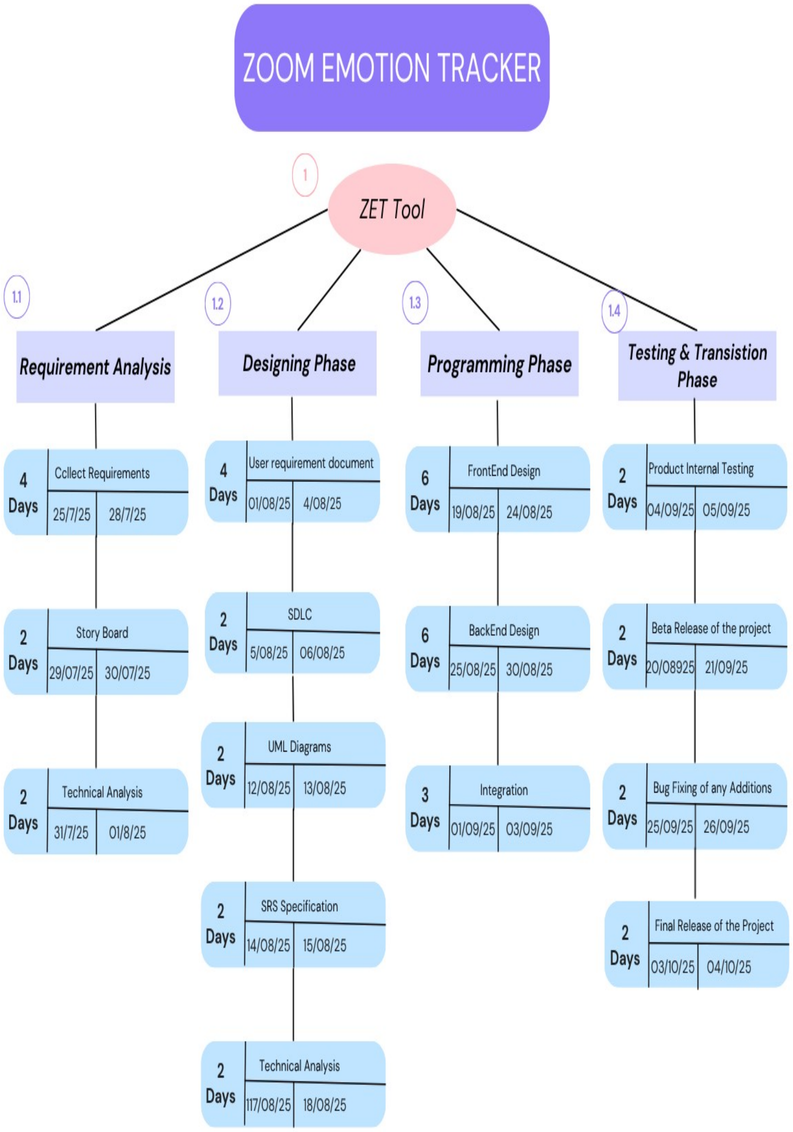
There are many departments involved in this project. The project manager has to divide the project roles and responsibilities to various departments. Here we have Work break down structure model for dividing roles. As per the WBS the project manager divides the project. Each level of this structure breaks the project deliverables or objectives of more specific and measurable units.

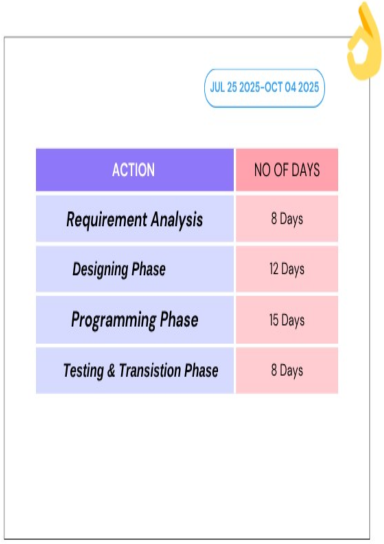
WBS will help with assigning responsibilities resource allocation, monitory the project and controlling the project. The WBS makes the deliverables more precise and concrete so that the project team knows exactly what has to be accomplished with in each deliverable**.**

Work Assignment to Team Insight Zoom:

Work Name	Assigned Person(s)
Requirements Gathering	Entire Team
Interface Design (Front-End UI)	Geetha , UshaRani
Storyboard / Flow Design	Geetha Sree, UshaRani, Ujjeef, Kumar
Work Breakdown Structure (WBS)	Usha Rani
URD (User Requirement Document)	Entire Team
SRS (Software Requirement Specification)	Entire Team
Risk Analysis	Entire Team
Front-End Development (ReactJS)	Geetha Sree, Usha Rani
Backend Design (Emotion bot,Socket.io,Python (Flask),TenserFlow Logic)	Kumar
Database Design(Django,SQLAlchemy) Code Integration	Ujjeef,Kumar Geetha Sree,Usha Rani,Kumar
Testing	Entire Team
Final Demonstration	Entire Team

WORK BREAK DOWN STRUCTURE(WBS)




JUL 25 2025-OCT 04 2025

ACTION	NO OF DAYS
Requirement Analysis	8 Days
Designing Phase	12 Days
Programming Phase	15 Days
Testing & Transistion Phase	8 Days

STORY BOARD

Introduction

Purpose

To detect emotions in virtual meetings and help speakers, teachers, trainers, or presenters adapt their communication based on **real-time audience feedback**.

The system identifies emotions such as:
Happy, Sad, Angry, Bored, Confused, Neutral, Surprised — all **in real time** by analyzing the participant’s webcam feed.

STORYBOARD SCENE – 1: The Communication Gap

Scene Description:

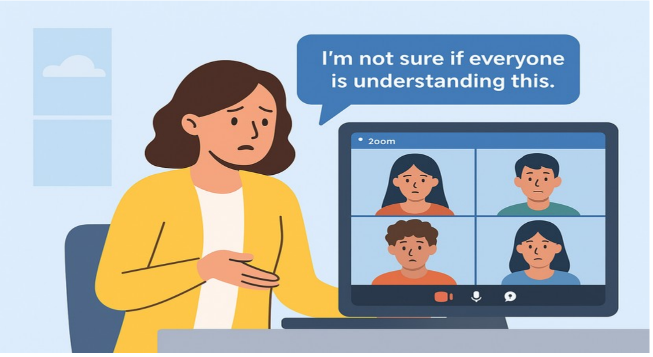
A **teacher** is conducting an online class on Zoom. She is explaining a topic but cannot understand whether the students are engaged or confused because their microphones are muted and only a few cameras are on.

Visual / Action

- The teacher looks unsure.
- The students appear passive or expressionless on the screen.
- The teacher says: *“I’m not sure if everyone is understanding this.”*

Purpose

To show the **problem** — in virtual meetings or classes, it is hard for speakers to understand audience emotions because **non-verbal cues are limited**.
This highlights the need for an automated **real-time emotion detection system.**



STORYBOARD SCENE – 2: Activating the Emotion Tracker

Scene Description

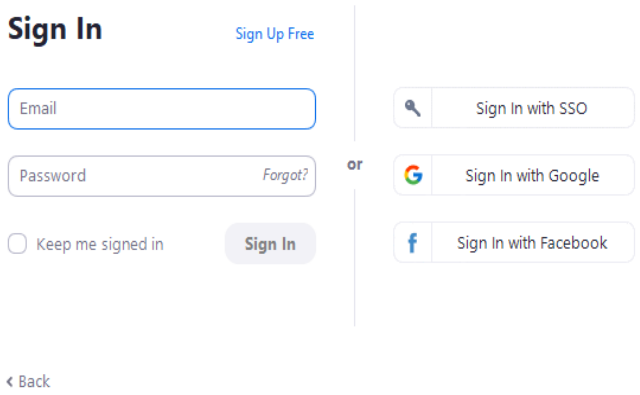
Before starting the session, the speaker enables the **Zoom Emotion Tracker** tool integrated into their Zoom interface.

Visual / Action

- A notification appears:
  “Emotion Tracker has started analyzing the session.”
- The system begins scanning active video feeds.
- Small indicators appear at the corners of each participant’s video tile.

Purpose

To show the **starting point** of the AI system.
This step represents the transition from traditional teaching or speaking to an **AI-enhanced environment** that captures emotional patterns in real time.



shutterstock.com · 2590084439

STORYBOARD SCENE – 3: AI-Based Emotion Detection

Scene Description

As the class progresses, the AI analyzes each participant's facial expressions continuously.

Visual / Action

- Faces are detected using rectangular bounding boxes.
- Real-time emotion labels appear above each face: *"Neutral", "Confused", "Happy", "Bored", "Angry"*
- Confidence scores appear below each label (e.g., *Happy – 82%*).

- The AI updates these labels every second based on facial changes.

Purpose

To show the **core function** of the solution — **machine learning-based emotion recognition** that replaces manual monitoring.

This scene explains how the technology processes data and produces meaningful emotional insights instantly.

STORYBOARD SCENE – 4: Live Emotion Dashboard

Scene Description

The teacher opens the integrated **Emotion Dashboard** that shows aggregated emotional data of all meeting participants.

Visual / Action

- A **pie chart** shows emotion distribution:
  - Happy – 35%
  - Confused – 28%
  - Bored – 20%

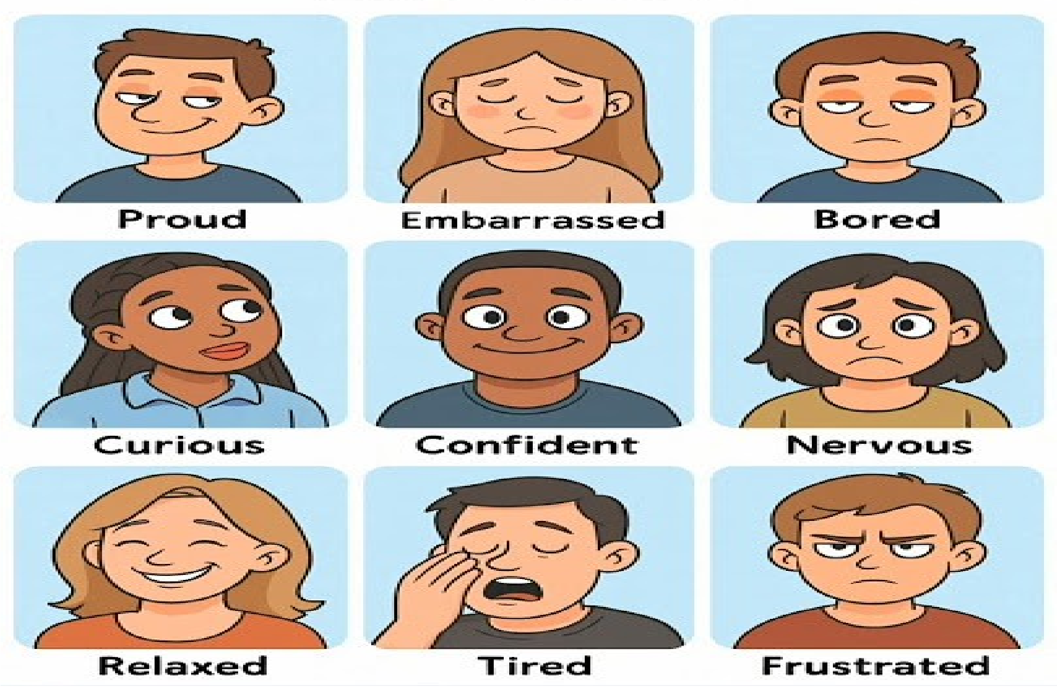
    - Neutral – 12%
    - Angry – 5%
- A **time-series graph** shows how emotions changed since the class started.
- Alerts appear:
  “High levels of confusion detected!”
  “Engagement decreasing.”

Purpose

To demonstrate how the system provides **real-time analytical insights**, allowing the speaker to:

- Monitor engagement
- Understand emotional reactions
- Adjust teaching speed or content delivery immediately

This dashboard empowers the speaker with **data-driven decision-making**.



STORYBOARD SCENE – 5: Speaker Adjusts Communication

Scene Description

The speaker modifies their teaching strategy based on dashboard feedback.

Visual / Action

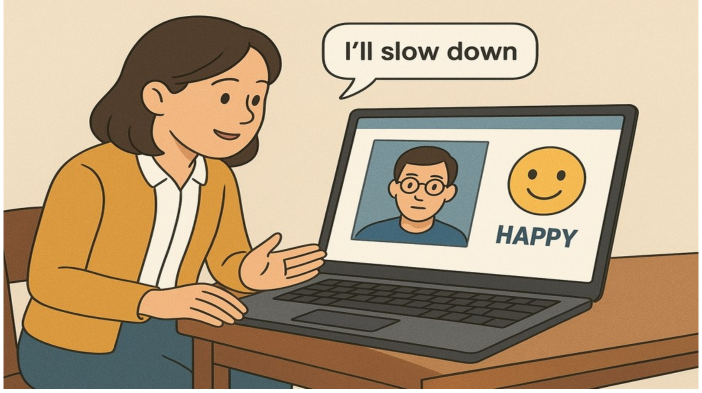
- The teacher sees increased confusion and says:
  “Let’s revisit the previous concept with another example.”
- Students start showing emotions like **Happy** and **Neutral**.
- Dashboard updates reflect improved engagement.

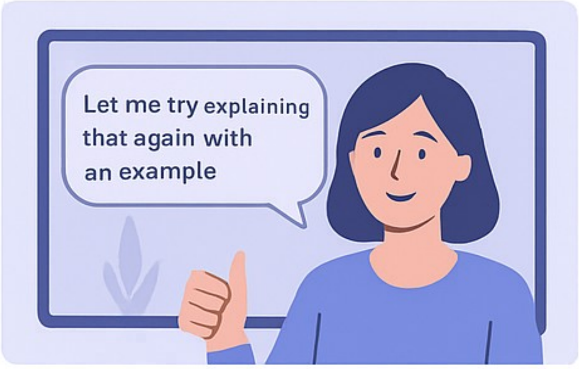
Purpose

To show the **impact** of using the Emotion Tracker.
By understanding audience emotions, speakers can:

- Re-explain difficult topics
- Increase interaction
- Change tone or pace
- Maintain higher engagement levels

This scene shows how the tool leads to **instant actionable improvement**.





STORYBOARD SCENE –6: Post-Session Emotion Summary

Scene Description

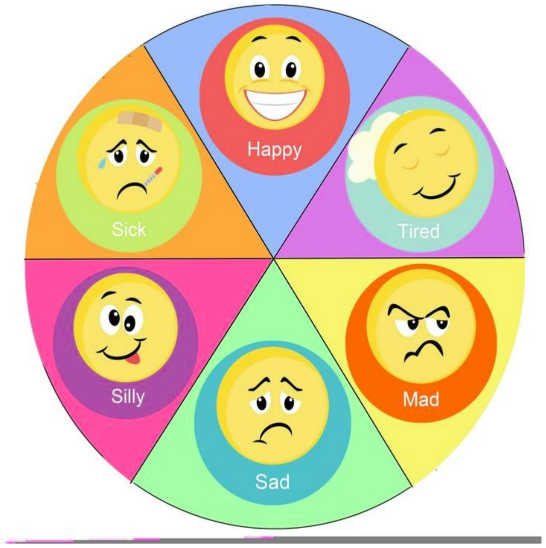
After the meeting ends, the speaker receives a detailed emotion analysis report for the entire session.

Visual / Action

- Summary includes:
  - Emotion percentage totals
  - Timeline charts
  - Engagement peaks and drops
  - Most common emotion
  - Suggestions for improvement
- The report is downloadable as a PDF.

Purpose

To highlight how the system continues to add value **after the session**, helping users analyze performance and improve future presentations or classes.



Conclusion :

The Zoom Emotion Tracker transforms virtual communication by bridging the gap created by lack of physical presence.
By analyzing facial expressions in real time, it empowers speakers to:

- Understand audience emotions
- Modify teaching or speaking strategies immediately
- Increase engagement and attention
- Deliver clearer and more effective sessions

The system brings **human connection and emotional understanding** back into online environments, making digital interactions more meaningful, responsive, and productive.

.

Key Highlights :

- **Real-time emotion detection** using AI
- Supports multiple emotions: Happy, Sad, Angry, Confused, Neutral, Bored, Surprised
- Live dashboard for instant audience feedback
- Alerts when engagement drops or confusion rises
- Helps teachers, trainers, and speakers adapt their communication
- Post-meeting emotion summary for performance improvement
- Reduces guesswork and improves clarity in virtual interactions
- Works with multiple participants simultaneously
- Enhances the overall online learning and meeting experience

USER REQUIREMENT DOCUMENT(URD)

Introduction

The rise of online education through platforms like Zoom has highlighted the need for better engagement monitoring.Traditional classrooms allow teachers to observe student behavior directly, but virtual classes lack this capability.

Purpose:

The purpose of this document is to gather the requirements that are needed for implementing the Zoom empotion tracker. It also Emotion recognitionbridges this gap by using AI to analyze facial expressions and engagement levels,helping teachers adapt their methods in real time.

Intended Audience:

The Audience who will conduct meetings in Zoom can use this App.

Stakeholders:

Client: Zoom Emotion Tracker

Users: individuals who teach classes or conduct meetings

Product Vision:

Vision Statement:

The product vision is to develop a ZET Tool, which is user friendly and easily accessible. This ZET Tool helps to provide to build bridge betweem teacher and students through emotion detection in online class.

Technologies:

Face Detection: OpenCV for detecting faces in video streams.

Backend: Django API, JWT for processing,Python(Flask), TenserFlow.

Frontend:Dahboard-React.

ML Models:Emotion detection(CNN,Transformer-based).

Database:SQLite(SQLAlchemy) For reports.

Integration:Zoom SDK/API

System in Context:

The system integrates directly with Zoom’s SDK to capture participant video streams.Each frame is analyzed locally (or on cloud) to detect emotions such as happy, sad, neutral, confused, bored.

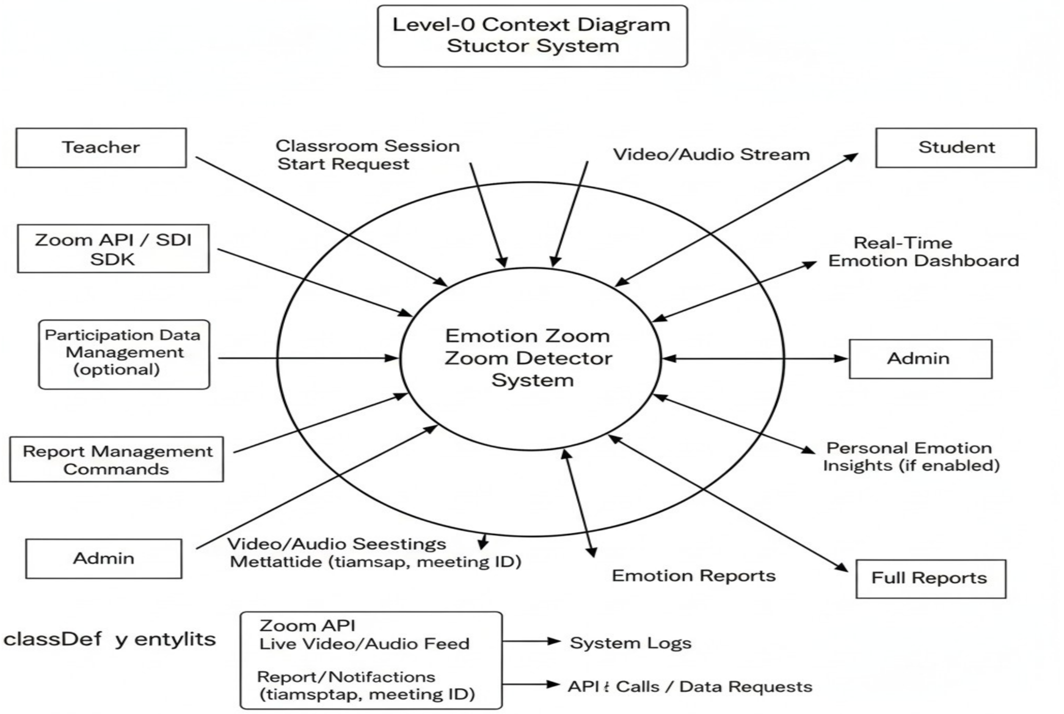
The processed results are sent to a teacher dashboard that aggregates all students emotions and provides trends during the session.

Input: Student video feed

Processing: Emotion recognition model

Output: Real-time analytics dashboard + post-class engagement

CONTEXT DIAGRAM:-



User Characteristics:

Teacher: Conduct engaged online classes and Require easy to read visualizations(graphs,alerts).

Students: Passive users, no additional action required.

Administrators: Configure app settings,manage reports.

Constraints:

Must comply with privacy regulation(GDPR,FERPA)

Limited bandwidth in Online classes.

Accuracy depends on Camera quality and student visibity.

System-Wide Requirements (Received):

Actors:

Teacher, Student, Admin.

Student – Passive participant, emotion tracked automatically

Teacher – Views real-time dashboard, receives alerts

Admin – Manages system, generates reports, ensures compliance

Emotion bot(AI Engine) – Detects emotions, processes data

Events:

The Zoom Emotion Tracker is a system which conducts engaged online meetings or classes.

The most critical events are:

1. Teacher/host conduct online class or meeting

2. Emotion bot joins with teacher given meetingid, password and start detecting facial emotions.

3. Students join classes

4. Emotion bot use ML models to detect emotions(bored, happy, confused, engaged, neutral, sad etc)

6. Provide real time feedback to teacherdashboard about overall emotional states of class.

7. Teacher get session past and present reports summaries and view emotional trends of students.

8. Admin manage reports,configure settings, and ensure participants privacy through secure data handling.

The below table provides a set of user visible events that define the functionalities that are in ZET Tool

Actor	**Action**	**Object**	**Frequency**	**Arrival Pattern**	**System Response**
Teacher	Opens the ZET web or desktop app	Login page	1/class session	Episodic	Displays login interface for authentication
Teacher	Creates a new online class session	Create Meeting option	1/class	Episodic	System generates unique Meeting ID and link
Student	Opens the ZET app and joins class	Meeting join interface	1/class	Episodic	System connects student video stream to session
Emotion Bot (AI System)	Starts emotion detection	Student video feed	Continuous	Periodic	Detects faces and classifies emotions
Teacher	Views real-time emotions dashboard	Emotion Dashboard window	Multiple times/class	Continuous	Displays emotion distribution
Student	Participates in the online class	Camera and mic interface	Continuous	Episodic	System monitors engagement

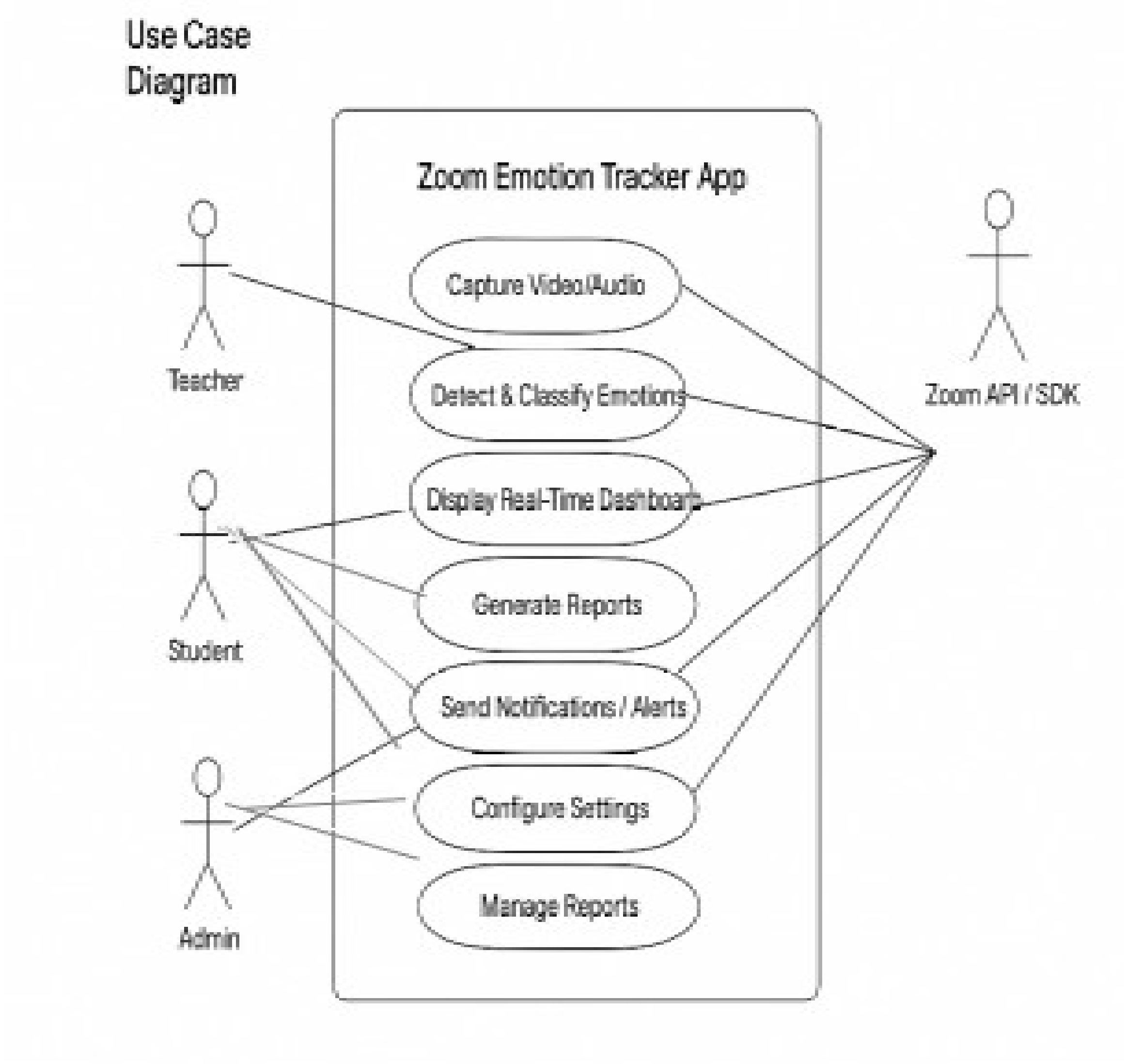
Teacher	Ends class session	End Meeting button	1/class	Episodic	Stops detection & generates report
System (ZET)	Generates emotion summary report	Database & analytics engine	1/class	Automated	Creates emotion trends and charts
Student	Submits feedback on emotion accuracy	Feedback form	1/class	Episodic	Stores feedback for model improvement
Admin	Updates AI emotion model	Admin control panel	As required	Episodic	Validates and activates new model
Teacher	Views previous emotion reports	Reports page	1/day	Episodic	Displays stored session reports
System (ZET)	Alerts teacher on emotional trends	Real-time notification	Continuous	Event-driven	Sends alert on anomalies

Functional Requirements:

Use case overview:

Use-Case ID	**Use-Case Name**	**Actors**	**Priority**	**Stability**	**Verifiability**
UC-ZET-CR	Create/Join Class Meeting	Teacher, Student	Very High	Stable	Verifiable
UC-ZET-FD	Face Detection and Emotion Tracking	Emotion Bot (AI System)	High	Stable	Verifiable
UC-ZET-TR	Teacher View Real-Time Emotions	Teacher	High	Stable	Verifiable
UC-ZET-ER	Emotion Report Generation	Teacher, System	Medium	Stable	Verifiable
UC-ZET-ST	Student Emotion Feedback	Student	Low-Medium	Stable	Verifiable
UC-ZET-AI	AI Model Update	System Administrator	Low	Stable	Verifiable

USE CASE DIAGRAM:-



Use Case Specifications:

Use Case 1: UC-ZET-CR — Create/Join Class Meeting

Use-Case ID: UC-ZET-CR	**Use-case Name:** Create/Join Class Meeting
Actor	Teacher / Student
Description:	Enables the teacher to create a class meeting and students to join using meeting ID.
Pre-conditions:	Teacher and students must be registered and logged in.
Success guarantee (Post-conditions):	Meeting successfully created/joined, and emotion tracking session initialized.
Frequency of use:	Very High
Main Success Scenario (Basic Flow):	1. Teacher logs in and clicks “Create Meeting”. 2. System generates Meeting ID and link. 3. Students enter the ID to join. 4. System starts live emotion detection via webcam. 5. Meeting dashboard becomes visible to teacher and students.
Extensions (Alternate Flows):	If meeting ID invalid → Show “Invalid Meeting ID”.
Frequency of occurance	Very HIgh

Use Case 2: UC-ZET-FD — Face Detection and Emotion Tracking

Use-Case ID: UC-ZET-FD	**Use-case Name:** Face Detection and Emotion Tracking
Actor	Emotion Bot (AI system)
Description:	Detects students' faces and emotions (happy, sad, confused, attentive, etc.) in real-time during class.
Pre-conditions:	Camera permission enabled, student present in meeting.
Success guarantee (Post-conditions):	Emotion data stored and visualized for teacher.
Frequency of use:	High
Main Success Scenario (Basic Flow):	1. Emotion Bot scans student faces every few seconds. 2. Facial features processed using AI emotion detection model. 3. Emotion labels assigned to each student. 4. System sends emotion summary to teacher's dashboard.
Extensions (Alternate Flows):	If face not detected → display "No Face Detected" icon.
Frequency of occurance	High

Use Case 3: UC-ZET-TR — Teacher View Real-Time Emotions

Use-Case ID: UC-ZET-TR	**Use-case Name:** Teacher View Real-Time Emotions
Actor	Teacher
Description:	Teacher views live emotional states of all students during ongoing class
Pre-conditions:	Active class meeting with connected students.
Success guarantee (Post-conditions):	Emotion statistics displayed continuously.
Frequency of use:	High
Main Success Scenario (Basic Flow):	1. Teacher opens “Emotion Dashboard”. 2. System displays emotion status (e.g., happy: 10, bored: 5). 3. Teacher monitors and adjusts teaching method accordingly.
Extensions (Alternate Flows):	*If data delay occurs → system shows “Refreshing Emotion Data...” message.*
Frequency of occurance	*HIgh*

Use Case 4: UC-ZET-ER — Emotion Report Generation

Use-Case ID: UC-ZET-ER	**Use-case Name:** Emotion Report Generation
Actor	Teacher / System
Description:	System generates post-class emotion report summarizing each student’s engagement and emotional trends.
Pre-conditions:	Class session completed successfully.
Success guarantee (Post-conditions):	Report saved in database and accessible by teacher.
Frequency of use:	Medium
Main Success Scenario (Basic Flow):	1. After class ends, system compiles emotion data. 2. Generates visual graphs and summaries. 3. Teacher views or downloads report.
Extensions (Alternate Flows):	If emotion data incomplete → display “Partial Data Available” warning.
Frequency of occurance	Medium

Use Case 5: UC-ZET-ST — Student Emotion Feedback

Use-Case ID: UC-ZET-ST	**Use-case Name:** Student Emotion Feedback
Actor	Student
Description:	Allows students to provide manual feedback or emotion correction for accuracy improvement.
Pre-conditions:	Student must have attended the session.
Success guarantee (Post-conditions):	Emotion dataset updated for future AI improvement.
Frequency of use:	Low-Medium
Main Success Scenario (Basic Flow):	1. Student opens feedback form after class. 2. Enters optional emotion feedback. 3. System stores feedback in emotion log.
Extensions (Alternate Flows):	*If student skips feedback → system closes session automatically.*
Frequency of occurance	*Low-Medium*

Use Case 6: UC-ZET-AI — AI Model Update

Use-Case ID: UC-ZET-AI	**Use-case Name:** AI Model Update
Actor	System Administrator
Description:	Updates the emotion detection model periodically to improve accuracy.
Pre-conditions:	Admin access and new model file available.
Success guarantee (Post-conditions):	New AI model deployed successfully.
Frequency of use:	Low
Main Success Scenario (Basic Flow):	1. Admin uploads new model. 2. System validates and installs model. 3. Admin receives confirmation of update.
Extensions (Alternate Flows):	If model upload fails → rollback to previous version.
Frequency of occurance	Low

NON-FUNCTIONAL REQUIREMENTS

Reliability

- The system should always work correctly without errors.

Usability

- Simple dashboard for non technical teachers.

Availability

- The system should be available 24x7, allowing to conduct meetings or zoom online classes anytime.
- Administrators and staff should be able to manage reports, security handling at any time.

Accessibility

- The system should support multi-user access from different devices and locations.
- Students, teachers, and admins should be able to log in and use the tool simultaneously.

Performance

- Emotion detection latency < 2 seconds.

Security

- The system should protect personal data (student details, payment info) using authentication and encryption.
- Only authorized users (students, teachers, admin,) should access their respective functions.

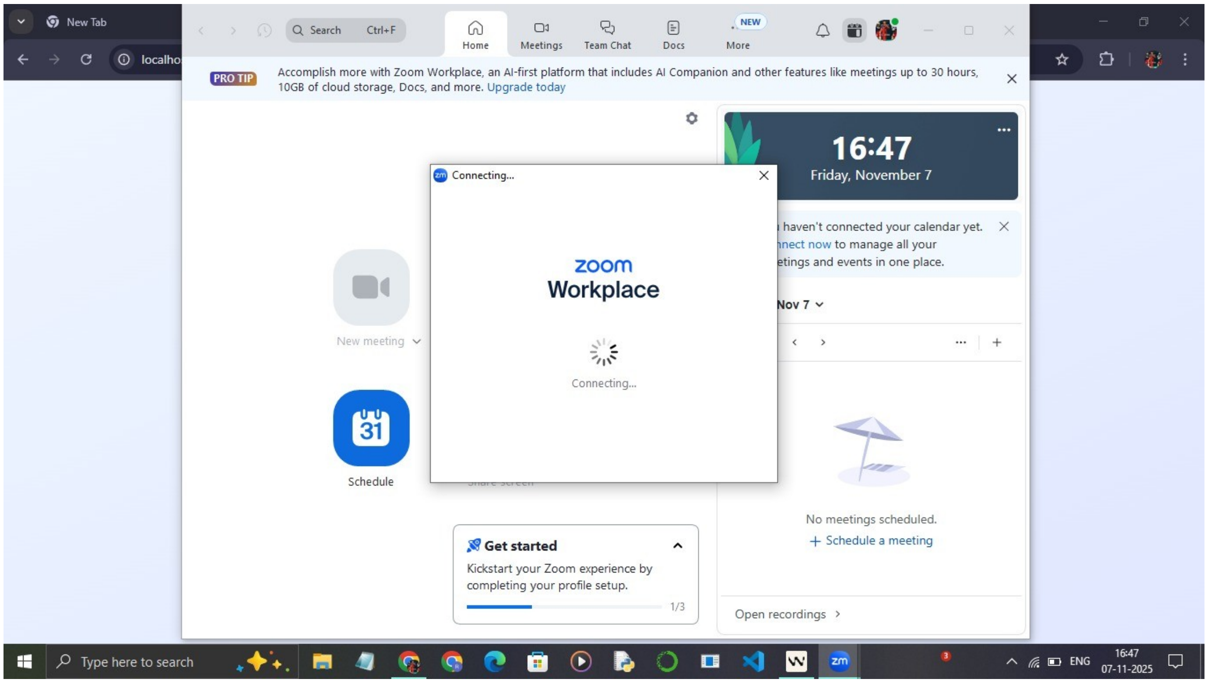
Platform Compatibility

- The system should run on operating system (Windows) and work across browsers (Chrome, Firefox, Edge).

- It should also be mobile-friendly for student access.

OUTPUT IMAGES OF PROJECT:

ZOOM WORKPLACE



PRO TIP Accomplish more with Zoom Workplace, an AI-first platform that includes AI Companion and other features like meetings up to 30 hours, 10GB of cloud storage, Docs, and more. Upgrade today

Connecting...

zoom Workplace

Connecting...

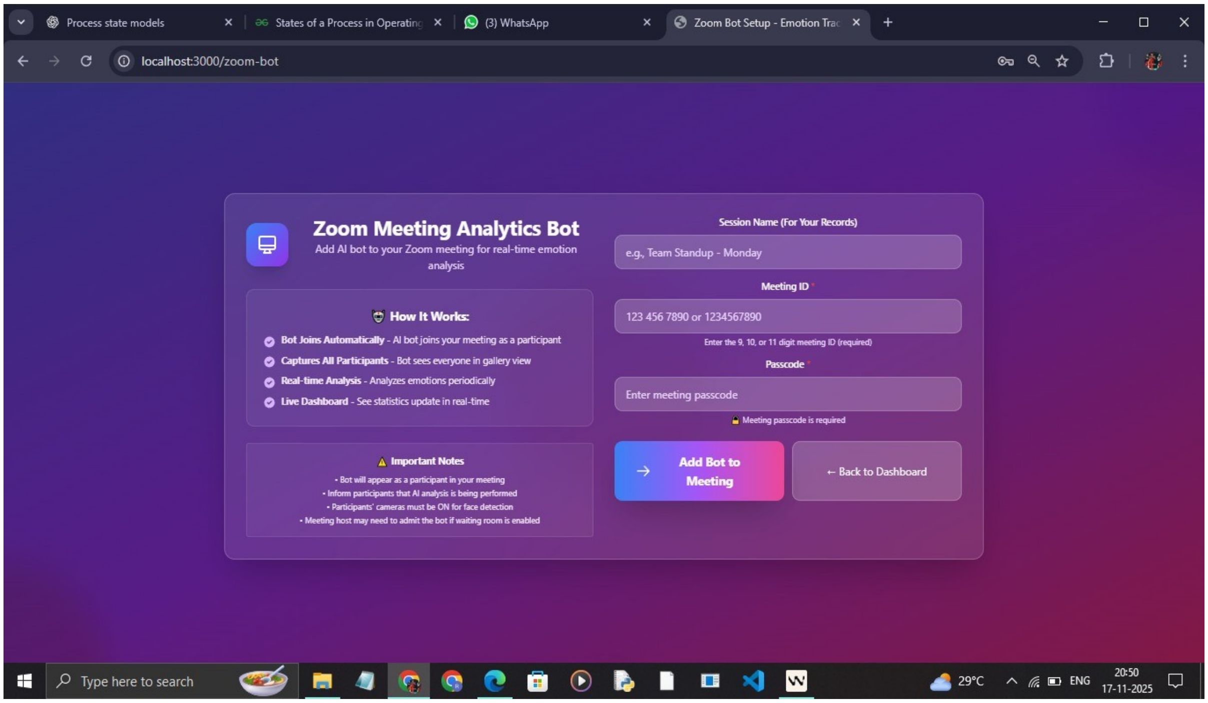
16:47
Friday, November 7

haven't connected your calendar yet. nect now to manage all your etings and events in one place.

No meetings scheduled.
+ Schedule a meeting

Get started
Kickstart your Zoom experience by completing your profile setup.
1/3

EMOTION BOT JOINS



Process state models
States of a Process in Operating
(3) WhatsApp
Zoom Bot Setup - Emotion Trac
localhost:3000/zoom-bot

Zoom Meeting Analytics Bot

Add AI bot to your Zoom meeting for real-time emotion analysis

How It Works:

- **Bot Joins Automatically** - AI bot joins your meeting as a participant
- **Captures All Participants** - Bot sees everyone in gallery view
- **Real-time Analysis** - Analyzes emotions periodically
- **Live Dashboard** - See statistics update in real-time

⚠ **Important Notes**

- Bot will appear as a participant in your meeting
- Inform participants that AI analysis is being performed
- Participants' cameras must be ON for face detection
- Meeting host may need to admit the bot if waiting room is enabled

Session Name (For Your Records): e.g., Team Standup - Monday

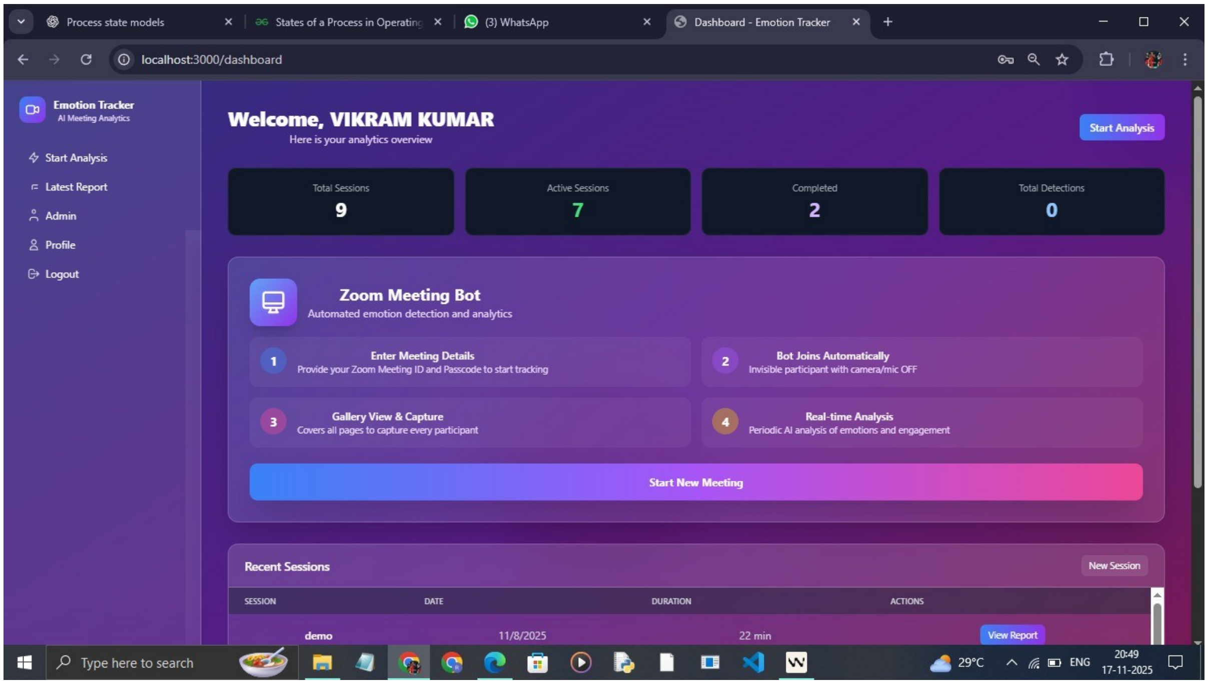
Meeting ID *: 123 456 7890 or 1234567890

Enter the 9, 10, or 11 digit meeting ID (required)

Passcode *: Enter meeting passcode

🔒 Meeting passcode is required

DASHBOARD OF TEACHER



Welcome, VIKRAM KUMAR

Here is your analytics overview

Total Sessions	Active Sessions	Completed	Total Detections
9	7	2	0

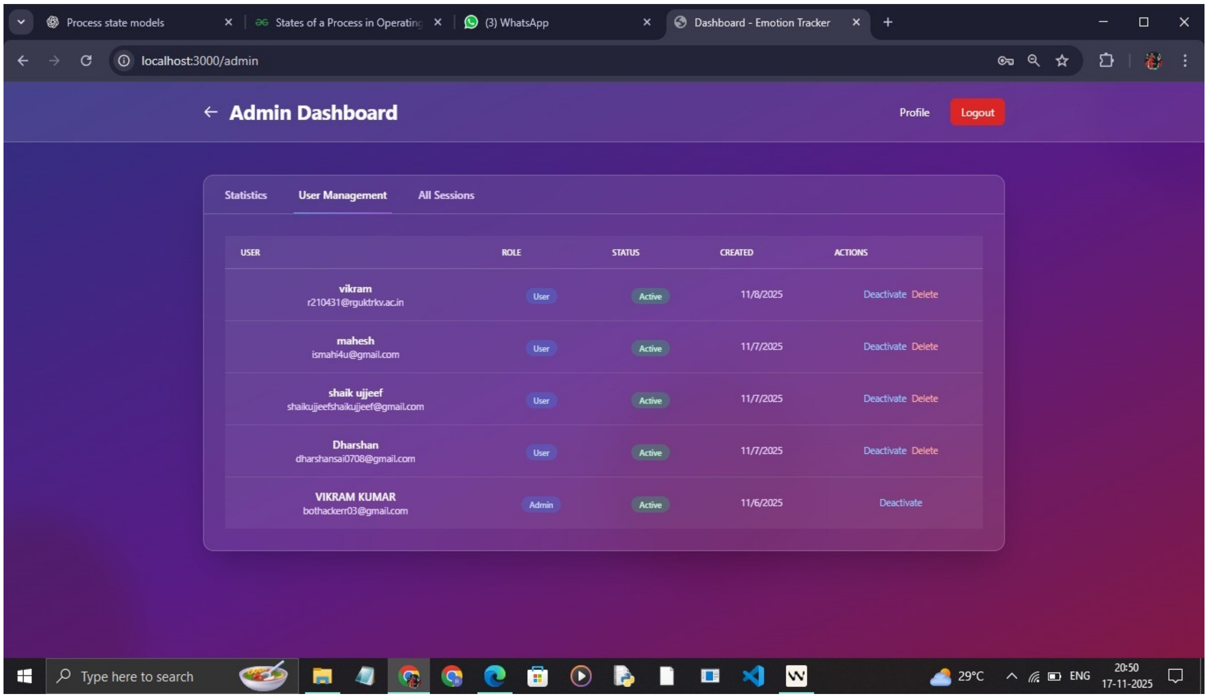
Zoom Meeting Bot

Automated emotion detection and analytics

1. **Enter Meeting Details** Provide your Zoom Meeting ID and Passcode to start tracking
2. **Bot Joins Automatically** Invisible participant with camera/mic OFF
3. **Gallery View & Capture** Covers all pages to capture every participant
4. **Real-time Analysis** Periodic AI analysis of emotions and engagement

Recent Sessions

SESSION	DATE	DURATION	ACTIONS
demo	11/8/2025	22 min	View Report

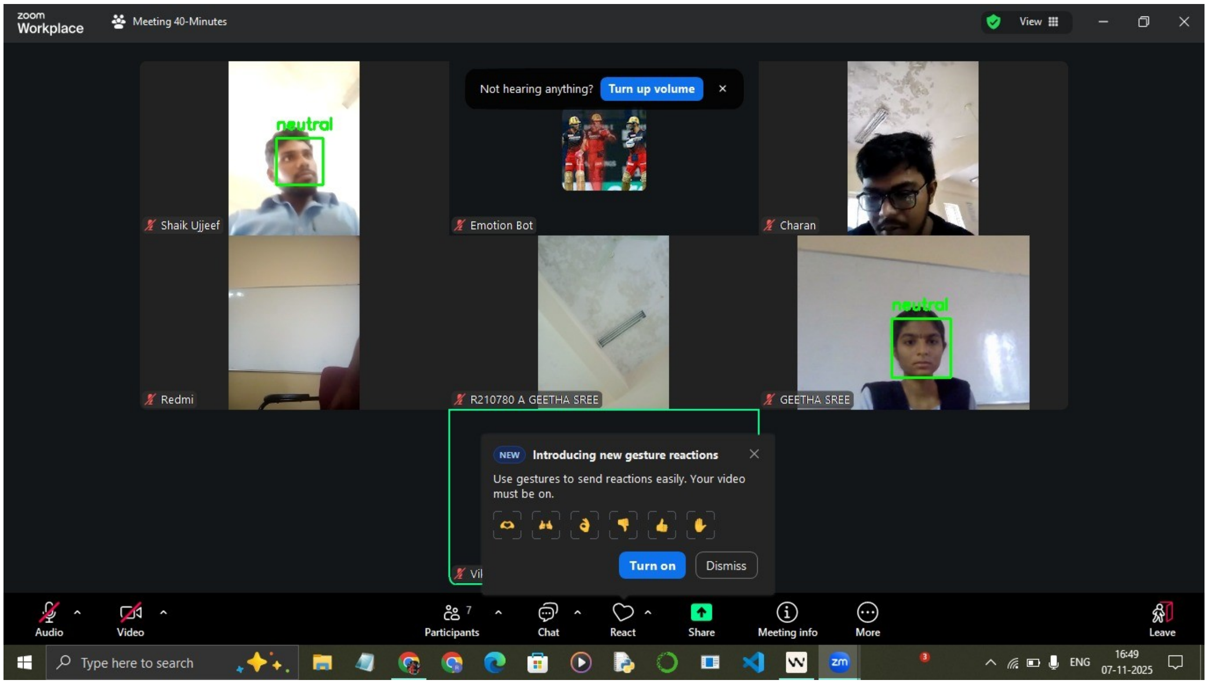


Admin Dashboard

Statistics | User Management | All Sessions

USER	ROLE	STATUS	CREATED	ACTIONS
vikram r210431@rguktrkv.ac.in	User	Active	11/8/2025	Deactivate Delete
mahesh ismahi4u@gmail.com	User	Active	11/7/2025	Deactivate Delete
shaik ujjeef shaikujjeefshaikujjeef@gmail.com	User	Active	11/7/2025	Deactivate Delete
Dharshan dharshansai0708@gmail.com	User	Active	11/7/2025	Deactivate Delete
VIKRAM KUMAR bothackerr03@gmail.com	Admin	Active	11/6/2025	Deactivate

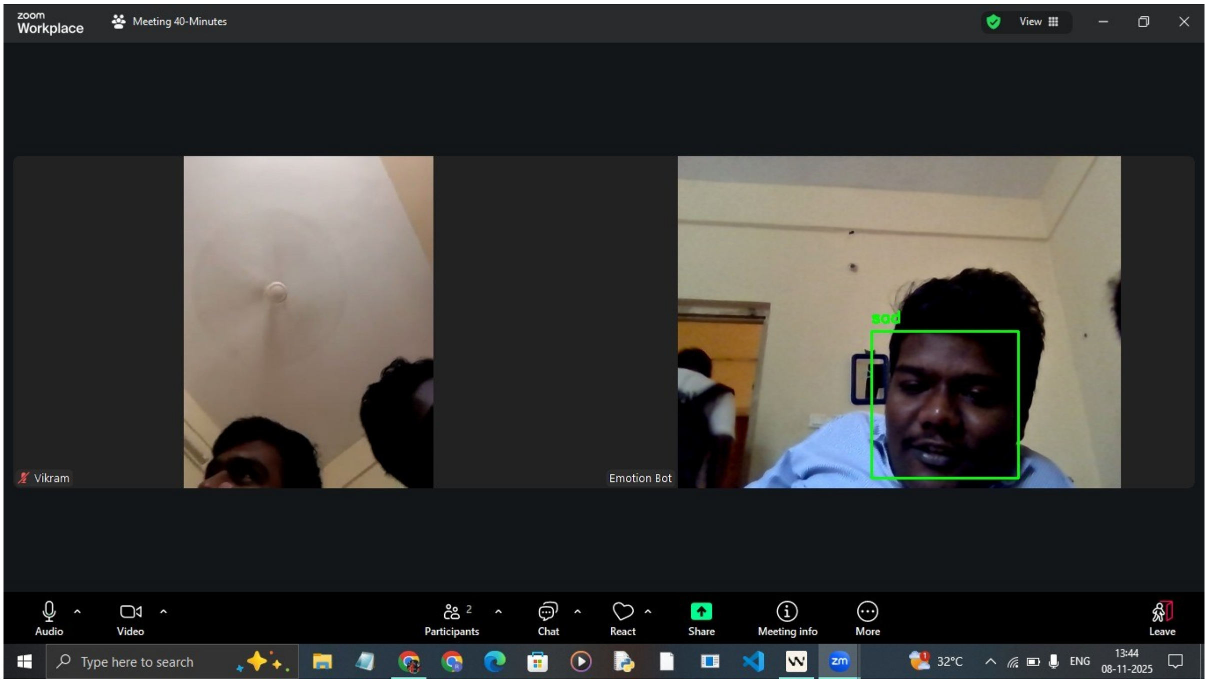
facial detection



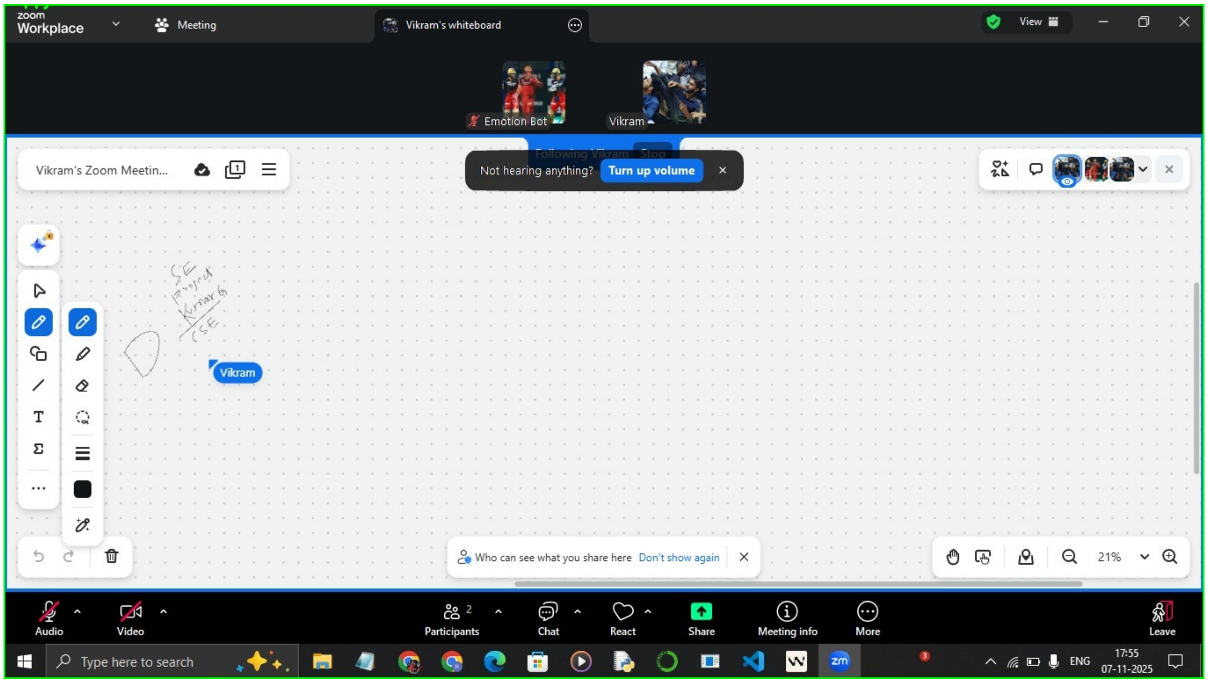
Not hearing anything? Turn up volume

NEW Introducing new gesture reactions

Use gestures to send reactions easily. Your video must be on.



screen sharing



DATABASE CODE:

```
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
from werkzeug.security import generate_password_hash,
    check_password_hash
import secrets

db = SQLAlchemy()

class User(db.Model):
    """Model for user authentication and profile"""
    __tablename__ = 'users'

    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), unique=True,
    nullable=False, index=True)
    username = db.Column(db.String(80), unique=True,
    nullable=False, index=True)
    password_hash = db.Column(db.String(255), nullable=False)
    full_name = db.Column(db.String(200))
    role = db.Column(db.String(20), default='user',
    nullable=False)  # user, admin
```

```
is_active = db.Column(db.Boolean, default=True, nullable=False)
is_verified = db.Column(db.Boolean, default=False, nullable=False)
reset_token = db.Column(db.String(100), unique=True, nullable=True)
reset_token_expiry = db.Column(db.DateTime, nullable=True)
created_at = db.Column(db.DateTime, default=datetime.utcnow, nullable=False)
last_login = db.Column(db.DateTime, nullable=True)

# Relationships
sessions = db.relationship('Session', backref='user', lazy=True)

def set_password(self, password):
    """Hash and set the user's password"""
    self.password_hash = generate_password_hash(password)

def check_password(self, password):
    """Check if the provided password matches the hash"""
    return check_password_hash(self.password_hash, password)
```

```
def generate_reset_token(self):
    """Generate a password reset token"""
    self.reset_token = secrets.token_urlsafe(32)
    self.reset_token_expiry = datetime.utcnow() +
datetime.timedelta(hours=24)
    return self.reset_token

def is_admin(self):
    """Check if user has admin role"""
    return self.role == 'admin'

def to_dict(self, include_sensitive=False):
    """Convert user to dictionary"""
    data = {
        'id': self.id,
        'email': self.email,
        'username': self.username,
        'full_name': self.full_name,
        'role': self.role,
        'is_active': self.is_active,
        'is_verified': self.is_verified,
        'created_at': self.created_at.isoformat() if
self.created_at else None,
            'last_login': self.last_login.isoformat() if self.last_login
    else None
        }
        if include_sensitive:
            data['reset_token'] = self.reset_token
            data['reset_token_expiry'] =
    self.reset_token_expiry.isoformat() if self.reset_token_expiry
    else None
        return data

    def __repr__(self):
        return f'<User {self.username}>'


class Session(db.Model):
    """Model for tracking session information"""
    __tablename__ = 'sessions'

    id = db.Column(db.Integer, primary_key=True)
    session_id = db.Column(db.String(100), unique=True,
    nullable=False)
    session_name = db.Column(db.String(200), nullable=False)
    user_id = db.Column(db.Integer, db.ForeignKey('users.id'),
    nullable=True)  # Link to user
    start_time = db.Column(db.DateTime, nullable=False,
    default=datetime.utcnow)

    end_time = db.Column(db.DateTime, nullable=True)

    summary = db.Column(db.Text, nullable=True)

    created_at = db.Column(db.DateTime,
    default=datetime.utcnow)


    # Relationship with emotion logs

    emotion_logs = db.relationship('EmotionLog',
    backref='session', lazy=True, cascade='all, delete-orphan')


    def __repr__(self):

        return f'<Session {self.session_name}>'


class EmotionLog(db.Model):

    """Model for logging individual emotion detections"""

    __tablename__ = 'emotion_logs'


    id = db.Column(db.Integer, primary_key=True)

    session_id = db.Column(db.String(100),
    db.ForeignKey('sessions.session_id'), nullable=False)

    participant_id = db.Column(db.String(100), nullable=False)

    emotion = db.Column(db.String(50), nullable=False)

    confidence = db.Column(db.Float, nullable=False)
```

```
timestamp = db.Column(db.DateTime, nullable=False,
default=datetime.utcnow)


def __repr__(self):
    return f'<EmotionLog {self.participant_id} -
{self.emotion}>'


def to_dict(self):
    return {
        'id': self.id,
        'session_id': self.session_id,
        'participant_id': self.participant_id,
        'emotion': self.emotion,
        'confidence': self.confidence,
        'timestamp': self.timestamp.isoformat()
    }
```