B.tech CSE (III YEAR – VI SEM) (2025-26)

DEPARTMENT OF COMPUTER ENGINEERING & APPLICATIONS

GLA University

17km Stone, NH-19, Mathura-Delhi Road P.O. Chaumuhan, Mathura – 281406 (Uttar Pradesh) India

Project Title:

Emotion Drift Analysis in Temporal Text Streams Using AI

Team Member 1 :

Sneha Pandit

Uni. Roll No.: 2315510206

Mentor :

Mr. Preshit Mangesh Desai

Project Synopsis :

Emotion Drift Analysis in Temporal Text Streams Using AI

---

0. Cover

Project Title : Emotion Drift Analysis in Temporal Text Streams Using AI

Team Name & ID: Sentia (T-92)

Institute / Course: GLA University, Mini-Project

Version: v1.0

Date: 05-Feb-2026

---

1. Overview

- **Problem statement:** Single-modality emotion detection systems fail to accurately capture human emotions due to ambiguity in text, lack of contextual understanding in facial expressions, and absence of temporal emotional analysis.
- **Goal:** Build a multimodal emotion intelligence system that combines **text-based emotion detection, facial emotion recognition, and emotion drift analysis** to improve accuracy and interpretability.
- **Non-goals:** Speech emotion recognition, clinical diagnosis, and production-scale deployment in v1.
- **Value proposition:** Improved emotion understanding through multimodal fusion, historical emotion logging, and drift detection over time using a modular FastAPI and React-based architecture.

2. Scope and Control

2.1 In-scope

- Text emotion detection
- Facial emotion recognition
- Multimodal fusion logic
- Emotion logging
- Timeline visualization
- Drift detection
- REST APIs
- Frontend dashboard

2.2 Out-of-scope

- Audio-based emotion detection
- Medical diagnosis
- Mobile applications
- Large-scale cloud deployment

2.3 Assumptions

- Users provide readable text input.
- Webcam access is available and permitted.
- Emotion categories are predefined.

2.4 Constraints

- Academic timeline constraints.
- Limited dataset size.
- Local compute and hardware limitations.

2.5 Dependencies

- Pretrained NLP emotion models
- Facial emotion recognition models
- Python ML ecosystem
- React visualization libraries.

2.6 Acceptance criteria and sign-off

- **GIVEN** a user submits text, **WHEN** the system processes the input, **THEN** an emotion label and confidence score are returned within 2 seconds.

- **GIVEN** sufficient historical data, **WHEN** emotion drift is analyzed, **THEN** significant emotional shifts are detected and logged.

Sign-off table

Stakeholder	Role	Decision area	Signature/Approval	Date
Mr. Preshit Desai	Mentor	Scope, final acceptance	Approved	05/02/26
Sneha Pandit	Project Lead	Release readiness	Approved	04/02/26

3. Stakeholders and RACI

Activity	Responsible (R)	Accountable (A)	Consulted (C)	Informed (I)
Requirements	Sneha Pandit	Sneha Pandit	Mentor	Team
Design	Sneha Pandit	Sneha Pandit	Mentor	Team
Implementation	Sneha Pandit	Sneha Pandit	—	Mentor
Testing	Sneha Pandit	Sneha Pandit	Mentor	Team
Release	Sneha Pandit	Sneha Pandit	Mentor	Dept

---

4. Team and Roles

Member	Role	Responsibilities	Key skills	Availability	Contact
Sneha Pandit	Project Lead	Architecture, ML, integration	Python, ML, NLP	10 hrs/wk	sneha.pandit_cs.aiml23@gla.ac.in
Sneha Pandit	Backend	APIs, database	FastAPI, SQL	8 hrs/wk	sneha.pandit_cs.aiml23@gla.ac.in
Sneha Pandit	Frontend	UI, charts	React	8 hrs/wk	sneha.pandit_cs.aiml23@gla.ac.in
Sneha Pandit	QA & Docs	Testing, documentation	Testing, writing	6 hrs/wk	sneha.pandit_cs.aiml23@gla.ac.in

---

5. Week-wise Plan and Assignments

Week	Dates	Milestones	Lead	Backend	Frontend	QA	Deliverables	Status
1	1 – 7 Feb	Requirements	Scope	API plan	Wireframes	Test plan	SRS	✅
2	8 – 14 Feb	Architecture	Review	DB schema	UI layout	Test data	ERD	Planned
3	15 – 21 Feb	Backend setup	Sync	Inference APIs	Auth UI	Smoke tests	Backend	Planned
4	22 – 28 Feb	Facial module	Review	Integration	Webcam UI	Checks	Facial module	Planned
5	29 Feb – 5 March	Fusion logic	Scope	Fusion	Visualization	Tests	Fusion demo	Planned
6	6 – 12 March	Drift analysis	KPIs	Drift logic	Charts	Perf tests	Drift demo	Planned
7	13 – 19 March	Hardening	Review	Bug fixes	Bug fixes	Regression	Test report	Planned
8	20 – 26 March	Release	Sign-off	Final APIs	UI polish	Docs	v1.0	Planned

6. Users and UX

- ## 6.1 Personas
    - **Analyst Alex:** Wants emotion trends and drift insights.
    - **User Uma:** Wants instant emotional feedback.
- ## 6.2 Top user journeys
    - **User:** Login / Sign in → Input text → Emotion detection → Facial capture → Fusion → Visualization.
    - **KPI:** Response ≤ 2 s, completion ≥ 90%.
- ## 6.3 User stories
    - As a user, I want my emotions analyzed using text and facial expressions for higher accuracy.
    - **GIVEN valid input WHEN prediction is triggered THEN emotion and confidence are displayed.**
- ## 6.4 Accessibility & localization
    - Keyboard-accessible UI
    - Contrast AA
    - Language: English.

---

7. Market and Competitors

7.1 Competitor table

Competitor	Product	Target users	Key features	Strengths	Weaknesses	Our differentiator
Text sentiment tools	NLP	General users	Text emotion	Simple	No non-verbal cues	Multimodal fusion
Facial emotion tools	CV	Research	Face emotion	Visual cues	No context	Text + face
Chatbots	Conversational AI	Users	Interaction	Scale	No drift analysis	Temporal analysis

7.2 Positioning

- Unique angle: Multimodal emotion detection with drift analysis.
- Measurable delta: Improved accuracy vs single-modality systems.

---

8. Objectives and Success Metrics

The objective of the **Emotion Drift Analysis in Temporal Text Streams Using AI System** is to design and implement an accurate and interpretable emotion analysis platform by combining text-based emotion detection, facial emotion recognition, and temporal emotion drift analysis.

Objectives

O1: To develop a text-based emotion detection module using machine learning techniques to classify emotions from user-provided text.

O2: To integrate a facial emotion recognition module using real-time webcam input to capture non-verbal emotional cues.

O3: To implement a multimodal emotion fusion mechanism that combines text and facial emotion outputs for improved accuracy.

O4: To log detected emotions with timestamps for historical and longitudinal analysis.

O5: To detect emotion drift by analyzing changes in emotional patterns over time.

O6: To visualize emotion timelines and drift alerts through an interactive frontend dashboard.

O7: To ensure reliable system performance with acceptable response time and modular design.

Success Metrics

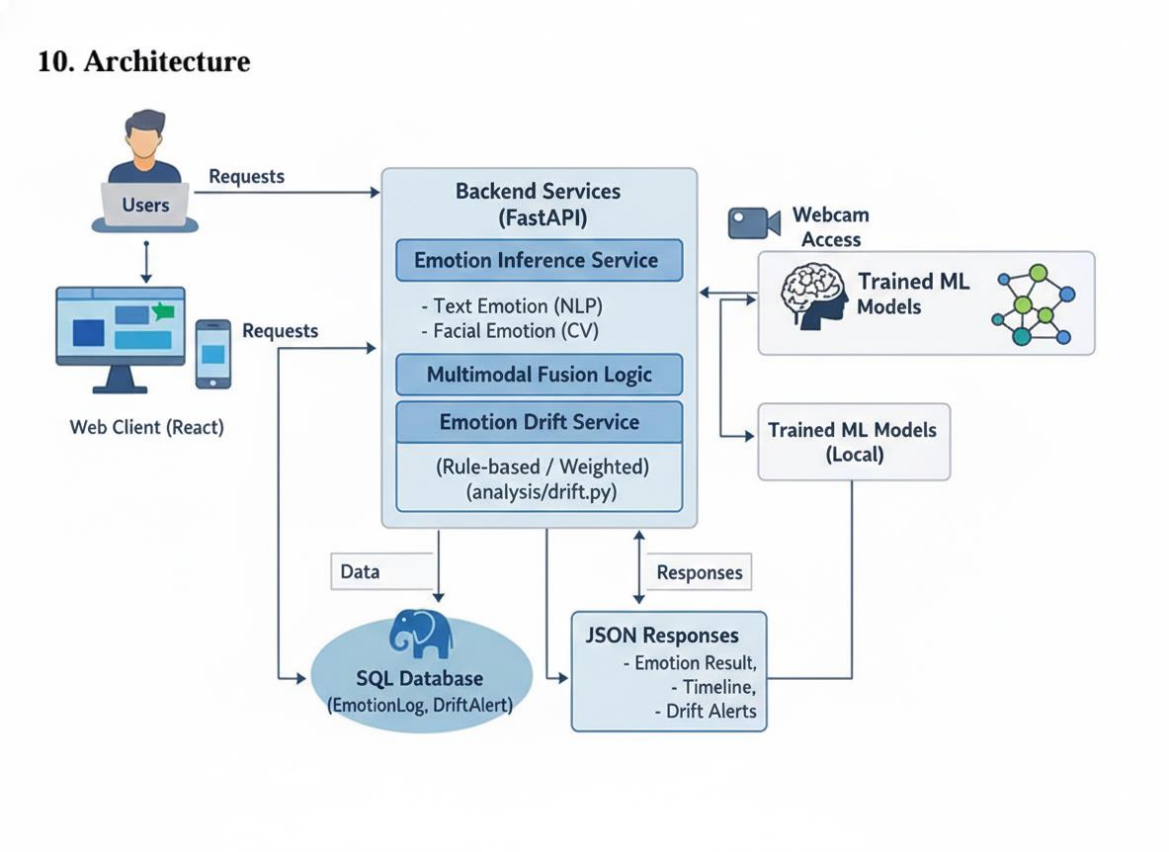
- Text emotion classification accuracy of **at least 75%** on test data.
- Successful facial emotion detection when webcam input is available, with graceful fallback to text-only analysis.
- Generation of a valid fused emotion output for every prediction request.
- **100% logging** of emotion predictions with accurate timestamps.
- Detection and recording of significant emotion drift patterns.
- Correct visualization of emotion timelines and drift alerts on the frontend.
- API response time of **less than 2 seconds** for core endpoints.

9. Key Features

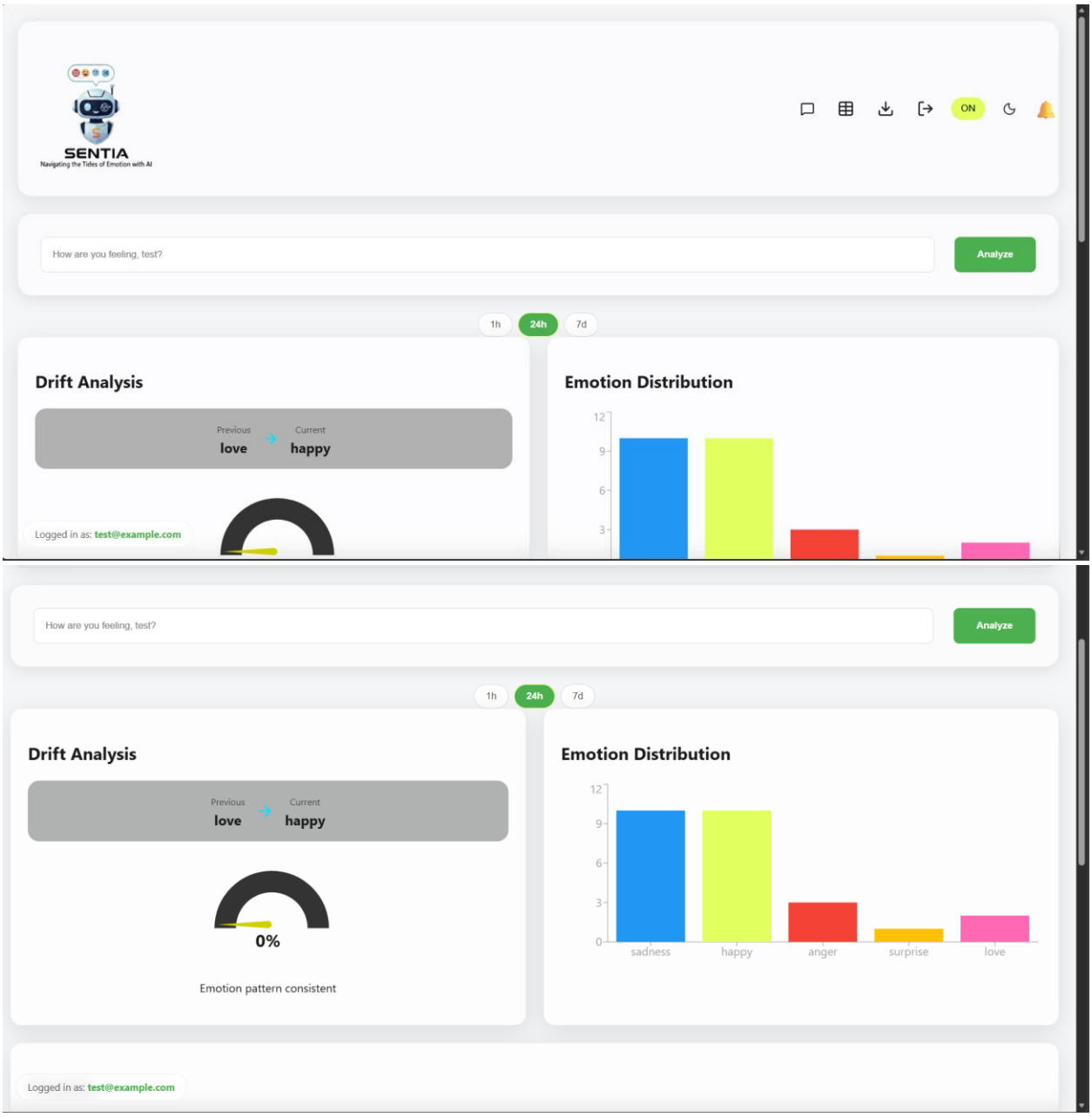
Feature	Description	Priority	Dependencies	Acceptance criteria
Text-Based Emotion Detection	Analyzes user-provided textual input using an NLP-based machine learning model to classify emotions and generate confidence scores.	Must	NLP emotion model, FastAPI backend	GIVEN valid text input WHEN prediction is requested THEN the system returns an emotion label and confidence score within 2 seconds
Facial Emotion Recognition	Detects emotional states from facial expressions captured through real-time webcam input using computer vision techniques.	Must	Webcam access, facial emotion model	GIVEN webcam access WHEN facial input is captured THEN a facial emotion label is detected without storing raw images
Multimodal Emotion Fusion	Combines text-based and facial emotion outputs using rule-based logic to generate a final emotion prediction.	Must	Text emotion output, facial emotion output	GIVEN outputs from both modalities WHEN fusion is applied THEN a single final emotion is produced with adjusted confidence
Emotion Timeline Retrieval	Retrieves historical emotion data for visualization and analysis over time.	Must	EmotionLog database, `/timeline` API	GIVEN a valid user ID WHEN timeline data is requested THEN the system returns ordered emotion history
Emotion Drift Detection	Identifies significant changes in emotional patterns across defined time windows using historical data.	Should	Emotion logs, drift analysis logic	GIVEN sufficient historical data WHEN drift analysis is triggered THEN significant emotional shifts are detected

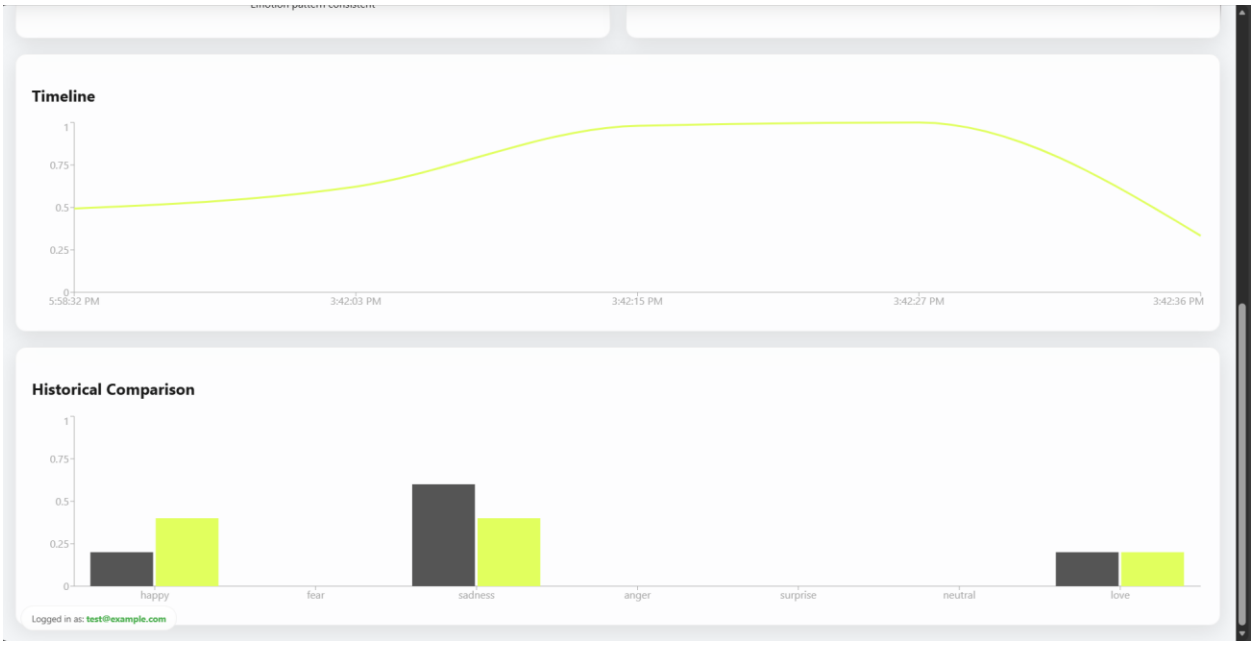
Drift Alert Generation	Generates and stores alerts when emotion drift is identified.	Should	Drift detection module, database	GIVEN detected drift WHEN thresholds are crossed THEN a drift alert is generated and stored
Frontend Visualization Dashboard	Displays emotion predictions, timelines, and drift alerts through an interactive user interface.	Should	React frontend, charting libraries	GIVEN valid API responses WHEN data is rendered THEN charts and alerts display correctl
RESTful API Communication	Enables structured interaction between frontend and backend components.	Must	FastAPI routes, Axios	GIVEN an API request WHEN processed THEN the correct JSON response is returned with appropriate status code
Privacy-Aware Data Handling	Ensures ethical handling of user data by avoiding storage of raw facial images and limiting data retention.	Must	Backend design, database policies	GIVEN facial emotion detection WHEN processing occurs THEN no raw image data is stored

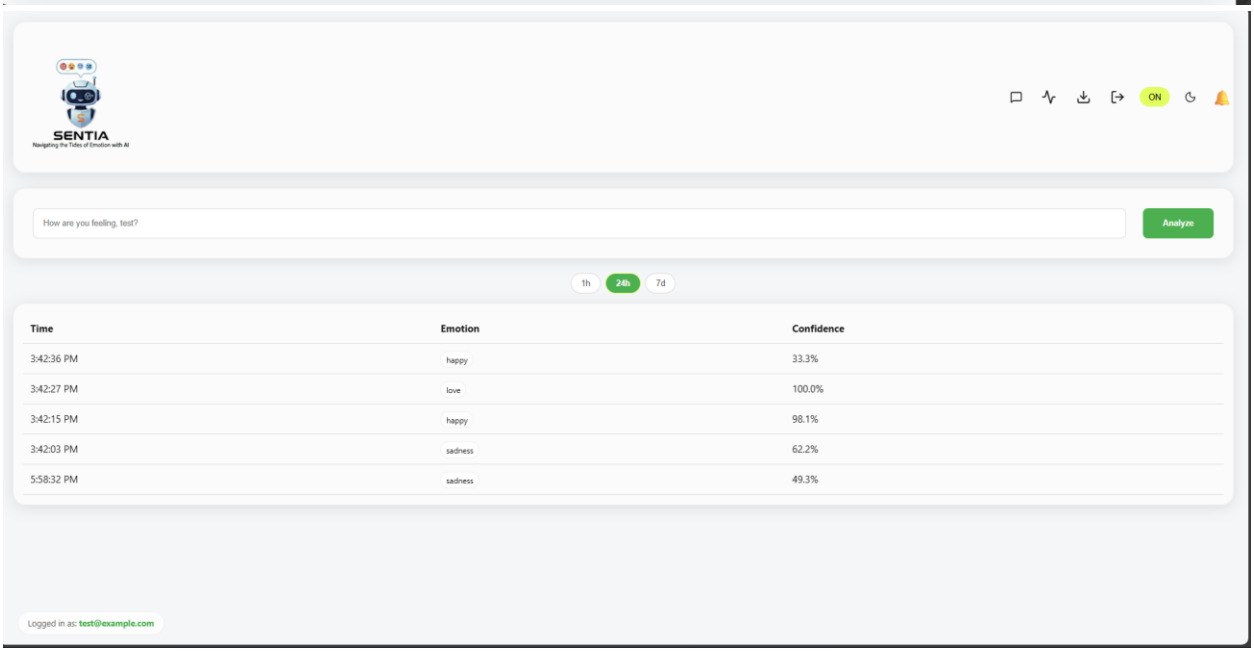
10. Architecture

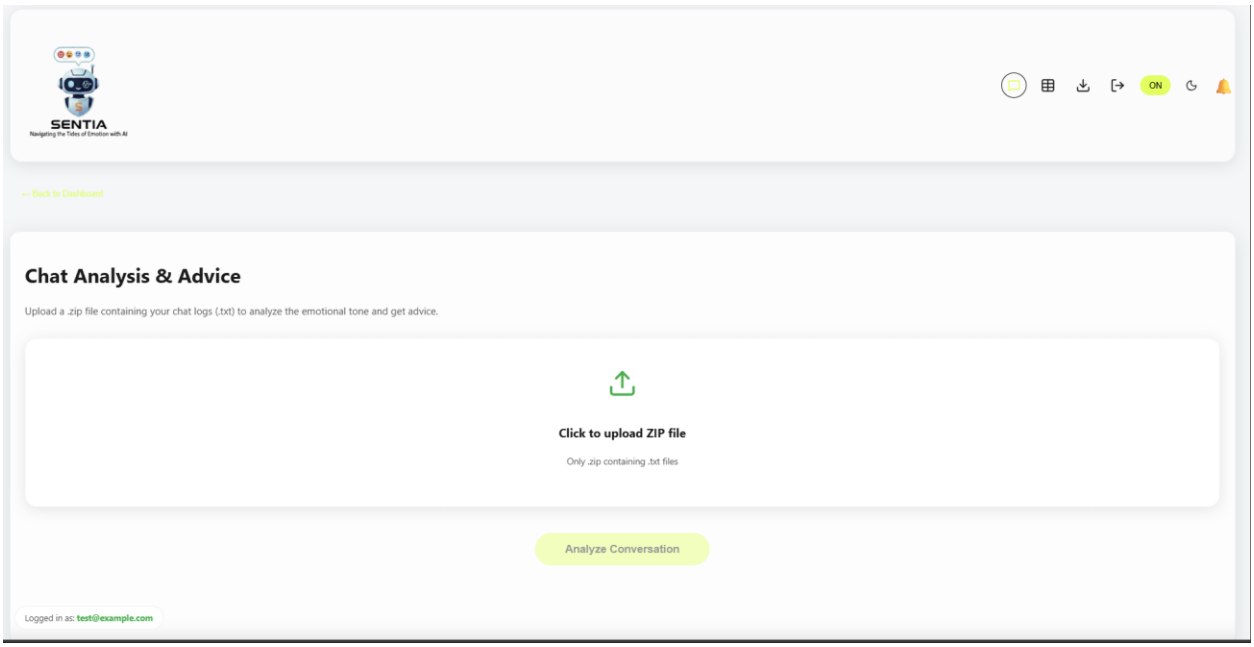


SnapShots :


SENTIA
Navigating the Tides of Emotion with AI
ON
How are you feeling, test?
Analyze
1h
24h
7d
Drift Analysis
Previous
love
Current
happy
Emotion Distribution
12
9
6
3
Logged in as: test@example.com
How are you feeling, test?
Analyze
1h
24h
7d
Drift Analysis
Previous
love
Current
happy
0%
Emotion pattern consistent
Emotion Distribution
12
9
6
3
0
sadness
happy
anger
surprise
love
Logged in as: test@example.com


Timeline
1
0.75
0.5
0.25
0
5:58:32 PM
3:42:03 PM
3:42:15 PM
3:42:27 PM
3:42:36 PM
Historical Comparison
1
0.75
0.5
0.25
0
happy
fear
sadness
anger
surprise
neutral
love
Logged in as: test@example.com


SENTIA
Navigating the Tides of Emotion with AI
ON
How are you feeling, test?
Analyze
1h
24h
7d
Time
Emotion
Confidence
3:42:36 PM
happy
33.3%
3:42:27 PM
love
100.0%
3:42:15 PM
happy
98.1%
3:42:03 PM
sadness
62.2%
5:58:32 PM
sadness
49.3%
Logged in as: test@example.com



Emotion Drift API

0.1.0 OAS 3.1

/openapi.json

Authorize

Reports

- POST /reports/generate Generate Report Api
- GET /reports/download/{report_id} Download Report

auth

- POST /auth/google Google Login
- POST /auth/signup Signup
- POST /auth/token Login For Access Token
- POST /auth/forgot-password Forgot Password
- POST /auth/reset-password Reset Password

Analysis

- POST /analyze/chat Analyze Chat Upload
- GET /analyze/chat/status/{job_id} Get Chat Analysis Status

default

- GET / Health
- POST /predict Predict
- GET /visualization/timeline Timeline
- GET /visualization/distribution Distribution
- GET /drift Drift
- GET /alerts Get Alerts
- GET /compare Compare

10.1 High-level

The Emotion Intelligence System follows a **client–server architecture** consisting of a frontend layer, backend services layer, and database layer. The system is designed to ensure modularity, scalability, and clear separation of concerns.

1. Frontend Layer (React)

The frontend is implemented as a **React Single Page Application (SPA)** responsible for user interaction and visualization.

Key Components:

- **Text Input UI:**
  Allows users to enter textual input for emotion analysis.
- **Webcam Capture UI:**
  Captures facial expressions using the user's webcam for facial emotion recognition.
- **Emotion Visualization:**
  Displays detected emotions along with confidence scores.
- **Emotion Timeline and Drift Alerts:**
  Visualizes historical emotion trends and displays alerts when emotion drift is detected.

Mapping:
Frontend components communicate with backend services using **Axios** for HTTP-based API calls.

Maps to:
React components + Axios

2. Backend Services (FastAPI)

The backend is implemented using **FastAPI** and is responsible for emotion inference, fusion logic, drift analysis, and data persistence.

Backend Service Mapping

Service Box	Description	Code Mapping
Emotion Inference Service	Performs text-based emotion detection using ML models	`ml/inference.py`
Multimodal Fusion Logic	Combines text and facial emotion outputs using rule-based logic	Backend processing layer
Emotion Drift Service	Analyzes historical emotion data to detect emotional shifts	`analysis/drift.py`
API Layer	Exposes RESTful endpoints for prediction, timeline, and drift analysis	`routes/report_routes.py`
Application Entry Point	Initializes FastAPI application and middleware	`api/main.py`

The backend ensures efficient request handling, modular logic separation, and clean API design.

3. Database Layer

The database layer stores emotion predictions and drift alerts for historical and analytical purposes.

Database Entities and Mapping

Entity	Description	File Mapping
EmotionLog	Stores detected emotions, confidence scores, user IDs, and timestamps	`db/models.py`
DriftAlert	Stores detected emotion drift alerts	`db/models.py`
DB Session	Manages database connections and sessions	`db/database.py`

A lightweight SQL-based database is used, suitable for academic-scale deployment.

10.2 API Specification Snapshot

The backend exposes RESTful APIs to support emotion prediction, historical analysis, and drift detection.

Endpoint	Method	Auth	Purpose	Request Schema	Response Schema	HTTP Codes
`/predict`	POST	—	Perform emotion inference	`text`, `user_id`	`emotion`, `confidence`	200
`/timeline`	GET	—	Retrieve emotion history	`user_id`	`logs[]`	200
`/drift`	GET	—	Detect emotion drift	`user_id`	`alerts[]`	200

Each endpoint is designed to return structured JSON responses with minimal latency and consistent behavior.

10.3 Configuration and Secrets Management

The system uses environment-based configuration to ensure security and flexibility.

- **Configuration Details:**
    - Database connection strings stored in environment variables.
    - ML model paths configured using environment variables.
    - Application settings loaded at runtime.
- **Secrets Management:**
    - Sensitive credentials are stored in environment variables.
    - Configuration files containing secrets are **Git-ignored**.
    - No hard-coded secrets are present in the codebase.
    - This approach ensures secure handling of configuration data while supporting easy deployment in academic environments.

---

11. Data Design

11.1 Core Entities

- Emotional Log: Stores individual emotion predictions generated by the system
- Drift Alert: Stores alerts generated when significant changes in emotional patterns are detected over time.
- Data Characteristics: Stores only derived emotion data; no raw text or facial images persisted

11.2 Data dictionary

Entity	Field	Type	Null?	Source	Notes
EmotionLog	id	UUID	No	System	Primary Key
EmotionLog	emotion	String	No	ML Model	Detected emotion label
EmotionLog	confidence	Float	No	ML Model	Confidence score of prediction
EmotionLog	timestamp	DateTime	No	System	Time of emotion detection

11.3 Schemas and migrations

- The database schema is managed using **versioned migrations** to ensure consistency and traceability of structural changes.
- Each schema update is tracked using migration versions.
- Migrations are applied incrementally to avoid data inconsistency.
- Rollback procedures are tested to ensure schema recovery in case of deployment or update failures.
- Schema design supports future extensibility without impacting existing data.
- This approach ensures database reliability and maintainability throughout the project lifecycle.

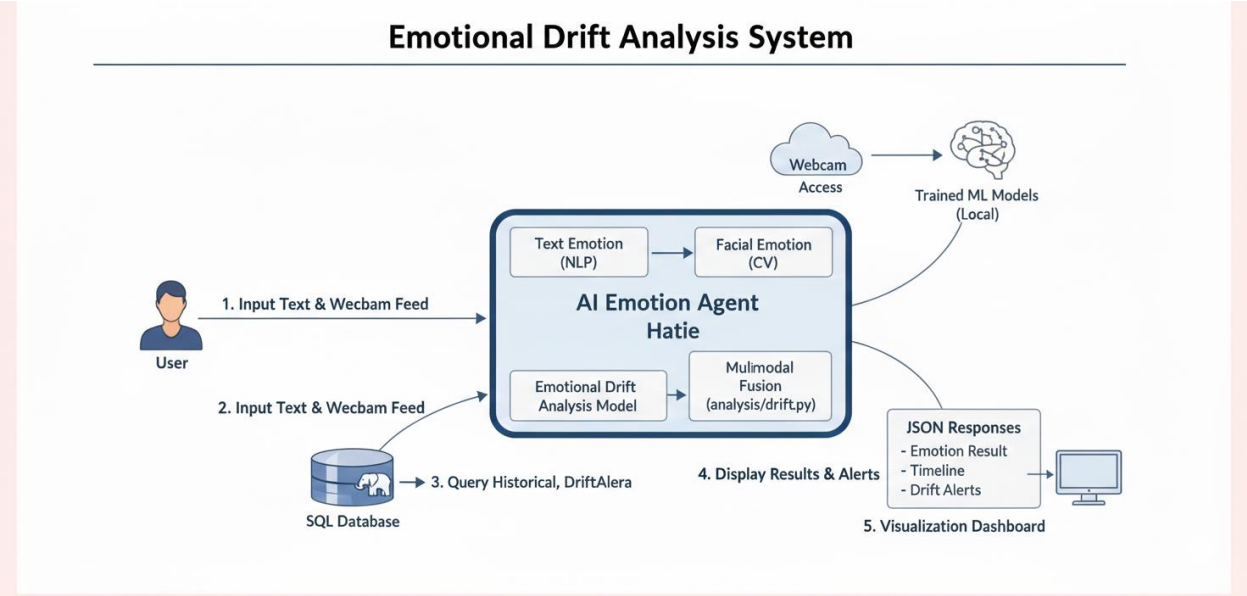
11.4 Privacy, retention, backup/DR

- **Privacy:**
  - No raw facial images or video frames are stored in the database.
  - Only derived emotion labels and confidence scores persisted.
  - User data is limited to identifiers required for analysis purposes.
- **Data Retention:**
  - Emotion logs are retained only for analytical and academic evaluation.
  - Historical data is used exclusively for emotion drift detection and visualization.
  - No third-party data sharing is performed.
- **Backup and Disaster Recovery (DR):**
  - Periodic database backups are maintained to prevent data loss.
  - Backup mechanisms support recovery of emotion logs in case of system failure.
  - The lightweight database design ensures quick restoration suitable for academic environments.

---

12. Technical Workflow Diagram

1. State Transition Diagram
2. Sequence Diagram
3. Use Case Diagram
4. Data Flow Diagram
5. ER Diagram
6. System Architecture Diagram



13. Quality: NFRs and Testing

13.1 Non-functional requirements

Metric	SLI	Target (SLO)	Measurement Method
Latency	p95 response time	≤ 2 seconds	API response time measured through backend logs
Availability	Uptime	≥ 99%	System availability monitoring during testing sessions
Error Rate	HTTP 5xx errors	≤ 1%	Backend error logs and API responses

13.2 Test plan

Area	Test Type	Tools Used	Coverage Target	Exit Criteria
Backend	Unit Testing	PyTest	≥ 70%	No Priority-1 defects
API	Integration Testing	Postman	All functional flows	All tests pass
User Interface	Manual Testing	Web Browser	Core user flows	All critical flows pass

13.3 Environments

The system is deployed and validated across multiple environments to ensure consistency and reliability.

Environment Flow:

Local → Test → Demo

- **Local Environment:**
  Used for development and debugging. Includes local FastAPI server, local database, and React development server.
- **Test Environment:**
  Used for functional validation, integration testing, and bug fixing.
- **Demo Environment:**
  Used for final project demonstration and evaluation.

---

14. Security and Compliance

14.1 Threat model (STRIDE)

Asset	Threat	Impact	Mitigation
Emotion data	Tampering	High	Access control and controlled database operations
API endpoints	Abuse	Medium	Rate limiting and request validation

- **Emotion Data Tampering:**
  Unauthorized modification of stored emotion logs could affect analysis accuracy. This risk is mitigated by restricting database access to backend services only.
- **API Abuse:**
  Excessive or malformed API requests could degrade system performance. Rate limiting and input validation reduce the likelihood of misuse.

14.2 AuthN/AuthZ

- Each request includes a user identifier to ensure correct data mapping.
- Authorization checks ensure users can access only their own emotion data.
- Role-based authorization is not implemented, as the system is designed for academic demonstration purposes.

14.3 Audit and logging

- To ensure traceability and system observability, the following events are logged:
  - Emotion prediction requests and results
  - Emotion drift detection events
  - API-level errors and exceptions
- Logs are used for:
  - Debugging and testing
  - Verifying correct system behavior
  - Supporting academic evaluation and demonstrations

14.4 Compliance

- Designed strictly for **academic and research purposes**.
- No sharing of user data with third-party services.
- No storage of raw facial images or video frames.
- Data collection and usage are limited to emotion analysis and evaluation.

---

15. Delivery and Operations

15.1 Release plan

- v1.0 academic demo.
- The release includes:
    - Fully functional FastAPI backend
    - React-based frontend dashboard
    - Text and facial emotion detection
    - Multimodal fusion and emotion drift analysis

15.2 CI/CD and rollback

- Code integration is performed manually using version control.
- Manual testing is conducted before each demo or submission.
- Deployment involves running the backend and frontend locally or in a controlled demo environment.
- In case of deployment or runtime issues, rollback is performed by reverting to the last stable version of the codebase.

15.3 Monitoring and alerting

- **Latency Monitoring:** API response times are observed through backend logs to ensure they remain within acceptable limits.
- **Error Monitoring:** Backend exceptions and HTTP error responses are logged and reviewed.

15.4 Runbooks

- **Inference Failure:**
  If facial emotion recognition fails due to webcam issues or access denial, the system automatically falls back to **text-only emotion detection**.
- **API Failure:**
  If an endpoint fails, the request is logged, and the system continues operation without crashing the frontend.

15.5 Communication plan

- Weekly progress updates are shared with the project mentor.
- Issues and risks are discussed during regular review sessions.
- A final project demonstration is conducted at the end of the development cycle.

---

16. Risks and Mitigations

Risk	Probability	Impact	Mitigation
Low emotion detection accuracy	Medium	High	Model tuning, threshold adjustment, and validation using test inputs
Webcam failure or access denial	Medium	Medium	Automatic fallback to text-based emotion detection
Time constraints	Medium	High	Strict scope control and prioritization of core features

---

17. Research and Evaluation

17.1 Research

- A review of existing emotion detection approaches was conducted to understand current techniques and limitations. The study focused on:
  - Text-based emotion detection using natural language processing models.
  - Facial emotion recognition using computer vision techniques.
  - Limitations of single-modality systems in capturing complete emotional context.
  - Existing research highlighting the importance of temporal emotion analysis.

17.2 Evaluation

- The system was evaluated using qualitative and quantitative methods:
  - **Accuracy Evaluation:**
    Emotion prediction results were compared against expected emotion labels for sample inputs to assess classification accuracy.
  - **Drift Analysis Evaluation:**
    Emotion logs were analyzed across multiple time windows to verify correct detection of significant emotional pattern changes.

- **Functional Evaluation:**
  End-to-end testing ensured correct operation of emotion prediction, logging, visualization, and drift alert generation.

---

18. Appendices

18.1 Glossary

- **NLP (Natural Language Processing):**
  A field of artificial intelligence that focuses on enabling machines to understand, interpret, and process human language.
- **CNN (Convolutional Neural Network):**
  A deep learning model commonly used for image and visual pattern recognition, applied in this project for facial emotion recognition.
- **Emotion Drift:**
  A significant change in emotional patterns over time, identified by analyzing historical emotion data rather than isolated predictions.
- **Multimodal AI:**
  An artificial intelligence approach that combines multiple data modalities, such as text and visual inputs, to improve analysis accuracy.

18.2 References

- Research papers on text-based and facial emotion detection techniques.
  - Ekman, P.
    Basic Emotions Theory
    https://www.paulekman.com/universal-emotions/
  - Calvo, R. A., & D'Mello, S.
    Affect Detection: An Interdisciplinary Review
    https://ieeexplore.ieee.org/document/6475935
  - Poria, S., et al.
    Multimodal Sentiment Analysis: Addressing Key Issues
    https://ieeexplore.ieee.org/document/8269807
  - Zeng, Z., et al.
    A Survey of Affect Recognition Methods
    https://ieeexplore.ieee.org/document/5206841

- Hugging Face – Emotion Models
  https://huggingface.co/models?pipeline_tag=text-classification
- FER-2013 Dataset (Kaggle)
  https://www.kaggle.com/datasets/msambare/fer2013

- Official FastAPI documentation for backend API development.
  - FastAPI Official Documentation
    https://fastapi.tiangolo.com/
  - SQLAlchemy ORM Documentation
    https://docs.sqlalchemy.org/

- Official React documentation for frontend user interface development.
  - React Official Documentation
    https://react.dev/
  - Axios HTTP Client
    https://axios-http.com/
  - Recharts Documentation
    https://recharts.org/en-US/
  - Scikit-learn Documentation
    https://scikit-learn.org/stable/
  - PyTorch Documentation
    https://pytorch.org/docs/stable/index.html
