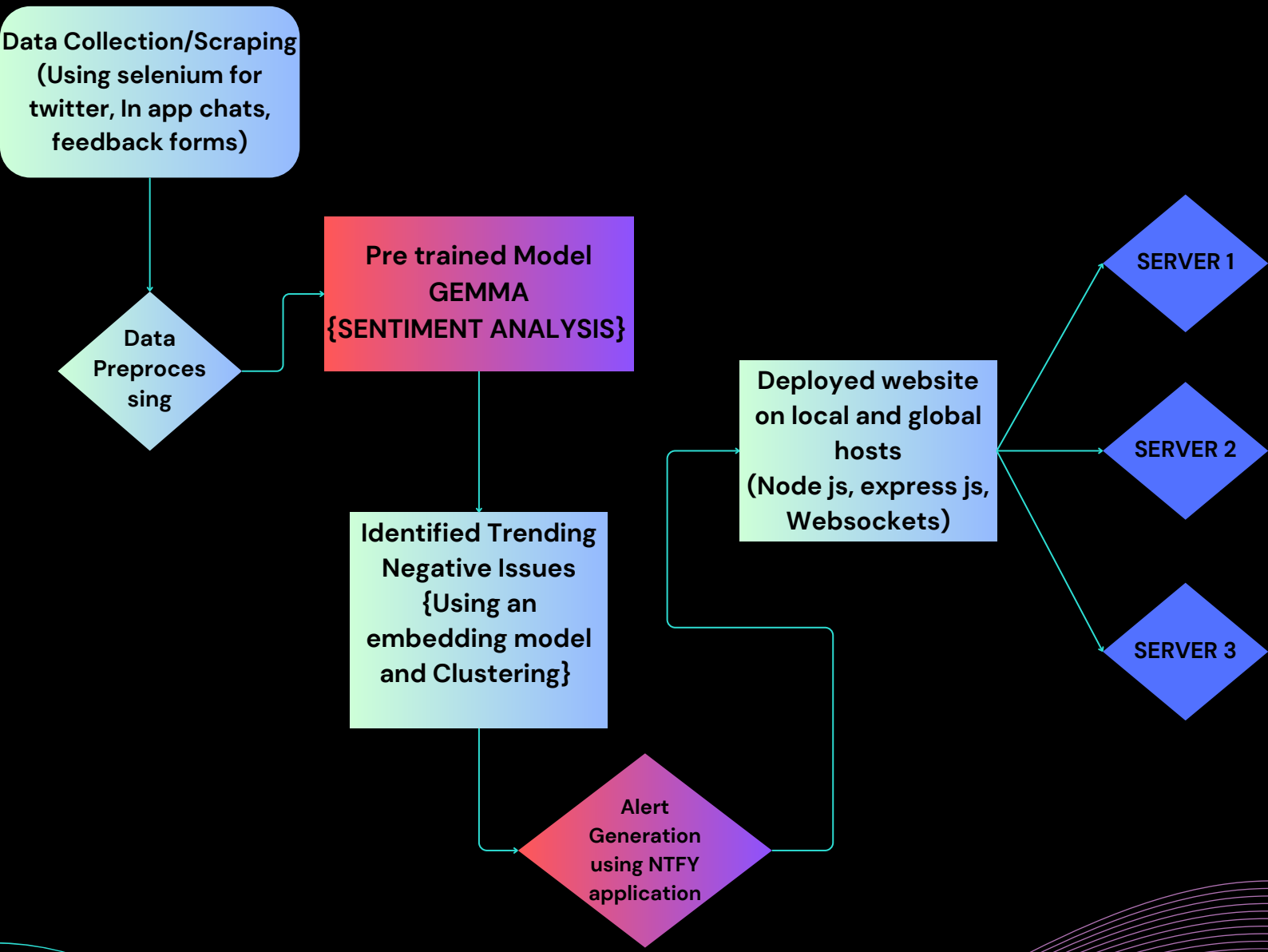
HACKFEST'25

AI POWERED SENTIMENT & ISSUE DETECTION PLATFORM

Prepared by :
Team A3VH

Workflow Diagram

Feedback is collected from multiple channels, processed through NLP for sentiment and issue detection, triggering real-time alerts and updating a live insights dashboard.



CHALLENGES FACED AND MAJOR COMMENTS

1. Data Scraping

Our main goal was data collection, (say from n sources) and then run sentiment analysis on it.
Didn't use twitter and other social media apps API's for scraping API as they were paid.

2. Data Preprocessing

- Removed the hashtags, mentions, punctuation marks, numeric digits and any extra whitespaces.
- Replaced newline characters with space

3. NTFY(Notification)

NTFY is infinitely flexible, and 100% free software.

4. Streamlit & FAST Api

Chose Streamlit for quick UI and real-time visualization, faster to prototype than React.
Preferred FASTAPI for async support and speed, better real-time handling than Flask.

CHALLENGES FACED AND MAJOR COMMENTS

5.Model Selection (Gemma 3 IB param)

Evaluation Criteria

- Macro F1-Score: For balanced class-wise performance

$$\text{Macro-F1} = \frac{F1_{\text{positive}} + F1_{\text{neutral}} + F1_{\text{negative}}}{3}$$

- Sarcasm Handling: Accuracy on manually tagged sarcastic samples
- Inference Latency: Average time per prediction on CPU
- Scalability & Integration: Ease of deployment in real-time systems

Comparative Model Performance

Model	Accuracy (%)	Remarks
Naive Bayes	76	Simple and fast but lacks context awareness
SVM	79	Handles patterns better but misses nuance
VADER	66	
VADER + BERT (Hybrid)	84	Good blend of speed and context
BERT-base	80	Context-aware but weaker on sarcasm
Gemma 3 (IB)	94	High accuracy, low latency, sarcasm-aware

CHALLENGES FACED AND MAJOR COMMENTS

6.Core ML Implementation

- Sentiment Classification with Gemma 3

We used the Gemma 3 (1B parameter) model to classify feedback into Positive, Neutral, and Negative categories. It was selected for:

- High performance on nuanced feedback (including sarcasm)
- Fast inference on CPU for real-time usage

- Embedding Generation and Clustering

Feedback messages were converted into semantic embeddings using sentence encoders. These embeddings were clustered using unsupervised algorithms (e.g., KMeans or DBSCAN) to identify groups of similar negative feedback.
Clusters with:

- < 4 entries → ignored as noise
- ≥ 4 entries → flagged as frequent issues

Each valid cluster's representative phrase was shown on the dashboard.

7.Real-time Alerts

When significant clusters are detected, NTFY alerts are triggered. The system updates clusters and alerts dynamically with new incoming feedback.