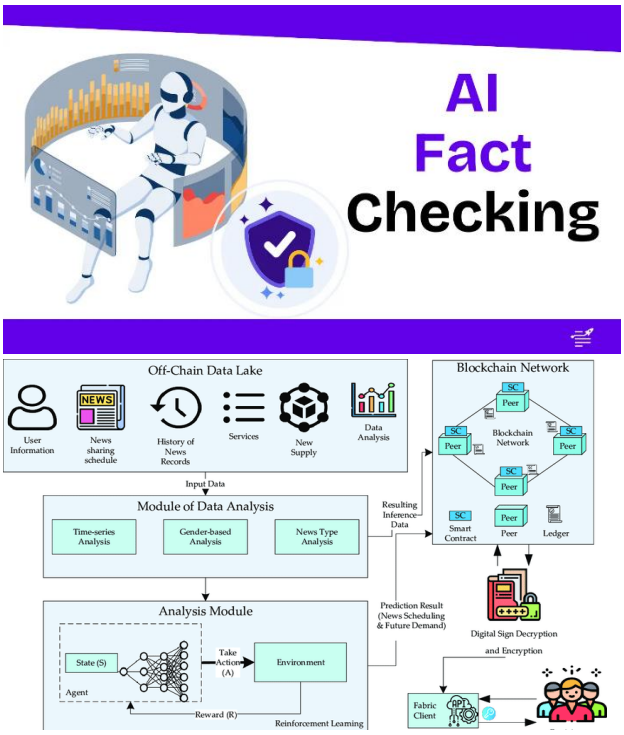
FactSeeker: MultiAgent, Multimodal, Autonomous Fact Verification AI

Fact-checking reimagined: protecting minds, empowering truth, in real-time.

Overview:
FactSeeker is an advanced, autonomous misinformation detection and verification system designed to address the growing problem of false information circulating across multiple platforms. The system combines state-of-the-art technologies including **Large Language Models (LLMs), Generative AI, Retrieval-Augmented Generation (RAG), multimodal analysis, and multi-agent autonomous frameworks** to provide real-time detection, verification, and public alerting.

The core mission of FactSeeker is to **protect society from misinformation**, empower individuals with accurate information, and promote informed decision-making. By leveraging agentic AI, FactSeeker does not just analyze data; it takes **proactive actions autonomously**, ensuring critical information reaches the public efficiently.



Key Components and Features

1. **Multi-Agent Autonomous Architecture:**
    - FactSeeker employs multiple specialized AI agents that independently handle **data ingestion, misinformation detection, fact verification, and public alerting**.
    - Each agent is responsible for a specific task but communicates with others in a coordinated, autonomous workflow.
2. **Multimodal Analysis:**
    - The system analyzes **text, images, videos, and audio** to detect potential misinformation across social media, news outlets, messaging platforms, and public forums.
    - Deep learning models, including fine-tuned **BERT, RoBERTa, and multimodal Transformers**, are used for classification, object recognition, and semantic analysis.
3. **Large Language Model Integration:**
    - LLMs are utilized for **context-aware understanding** of content, summarizing complex information, generating human-readable explanations, and assisting in zero-shot classification tasks.
4. **Retrieval-Augmented Generation (RAG) Module:**
    - FactSeeker retrieves verified information from **trusted knowledge bases** such as government portals, WHO guidelines, established fact-checking websites, and peer-reviewed data.
    - RAG ensures that every claim flagged is compared with authentic sources to enhance accuracy and reliability.
5. **Generative AI for Public Alerts:**
    - Generative AI is used to **rewrite verified information** into simple, clear, and actionable alerts suitable for the general public.
    - Supports multiple languages to increase accessibility and reach.
6. **Real-Time Alerts and Notifications:**
    - FactSeeker autonomously sends **instant notifications** to users via **WhatsApp, Telegram, web dashboards, and mobile applications**, enabling proactive dissemination of accurate information.

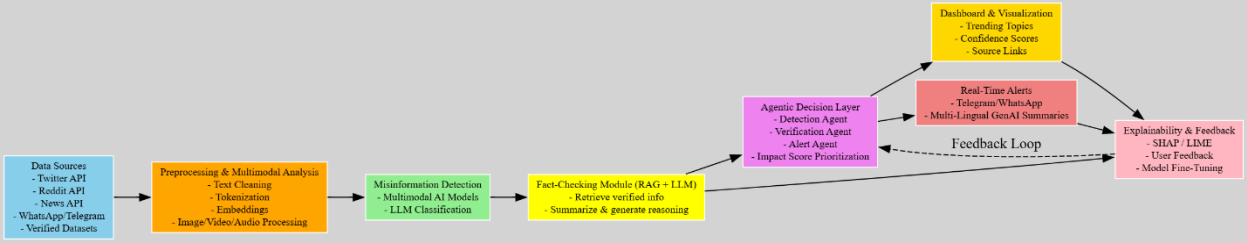
- Alerts include summaries, source references, confidence scores, and recommended actions.

7. **Explainable AI:**
   - Each flagged content item is accompanied by an **explainability layer** that shows why it was classified as misinformation, what sources were referenced, and the reasoning behind public alerts.
   - Enhances user trust and allows moderators or officials to review AI decisions efficiently.
8. **Impact Scoring and Prioritization:**
   - FactSeeker calculates an **Impact Score** for every misinformation item based on virality potential, source credibility, and severity of misinformation.
   - High-risk misinformation is prioritized for faster verification and alerting.
9. **Predictive Trend Analysis:**
   - The system leverages social network analysis and AI-based trend prediction to **anticipate potential misinformation** before it becomes viral.
   - Enables proactive mitigation and early awareness campaigns.
10. **Community Engagement:**
    - Users can **interact with the system** by providing feedback on alerts, contributing to model fine-tuning, and reporting new sources of misinformation.
    - This real-time feedback loop strengthens AI learning and improves detection accuracy.
11. **Use Cases:**
    - **Health Crises:** Detect false claims about vaccines, treatments, or epidemics and send verified advice to the public.
    - **Political Events:** Identify trending rumors or misleading narratives during elections or conflicts to prevent panic and misinformation spread.
    - **Public Safety:** Monitor social media during festivals, transport disruptions, or disasters and alert citizens with verified instructions.
12. **Technical Stack:**

- **Data Ingestion:** Twitter API, Reddit API, NewsAPI, RSS feeds, public messaging platforms
- **Backend / AI Models:** Python, PyTorch, TensorFlow, HuggingFace Transformers
- **RAG Vector Database:** FAISS, Pinecone, or Weaviate
- **Agentic Framework:** LangChain or AutoGPT style autonomous agents
- **Frontend / Dashboard:** React.js or Streamlit
- **Notification Channels:** Telegram Bot API, WhatsApp Business API, Web Push Notifications

13. **Differentiating Factors / Innovation:**
    - Full **autonomous operation**, not reliant on manual triggers.
    - **Multimodal intelligence**—handles text, images, videos, and audio.
    - **Explainable AI + LLM summaries** for transparency and trust.
    - **Real-time impact scoring and prioritization**, ensuring critical misinformation is addressed first.
    - **Multi-agent coordination** to parallelize detection, verification, and alerting tasks.
    - Predictive modeling for **emerging misinformation trends**.
14. **Impact and Societal Relevance:**
    - Studies show that in India, over **70% of adults receive news primarily through social media and messaging apps**, where misinformation spreads rapidly.
    - FactSeeker reduces the **risk of harm from false information**, increases public awareness, and promotes **responsible information sharing**.
    - Example: During the COVID-19 pandemic, misleading treatment suggestions caused significant health risks. FactSeeker would have autonomously flagged and corrected these narratives in real-time, preventing misinformation propagation.

Technical Implementation and Architecture



1. Overall Architecture Overview

Key layers:

1. **Data Ingestion Layer – Collects data from multiple sources**
2. **Preprocessing & Multimodal Analysis Layer – Cleans and converts data into analyzable formats**
3. **Detection & Verification Layer – Misinformation detection via AI models + RAG fact-checking**
4. **Agentic Decision Layer – Autonomous action planning for alerts**
5. **Notification & Dashboard Layer – Sends alerts and visualizes results for users and admins**
6. **Explainability & Feedback Layer – Provides reasoning and incorporates user feedback**

---

2. Data Ingestion Layer

- **Sources:**
  - **Social media APIs: Twitter, Reddit**
  - **News APIs: NewsAPI, RSS feeds**
  - **Messaging apps: WhatsApp Business API (optional for hackathon demo)**
  - **Pre-verified datasets: Kaggle misinformation datasets for model training**
- **Implementation:**
  - **Python scripts with async API calls for real-time streaming**
  - **Data stored in NoSQL database (MongoDB) for flexibility**

- **Metadata collection: timestamp, source, author, likes/shares, content type**

---

3. Preprocessing & Multimodal Analysis

- **Text Processing:**
  - **Tokenization, stopword removal, normalization**
  - **LLM embedding generation (BERT, RoBERTa, or Sentence-BERT)**
- **Image Processing:**
  - **CNN or Vision Transformer models for image verification and anomaly detection**
- **Video/Audio:**
  - **Extract frames, transcribe speech using Whisper/other speech-to-text models**
  - **Analyze using multimodal transformers**
- **Outcome: Clean, vectorized multimodal representations ready for detection**

---

4. Detection & Verification Layer

4.1 Misinformation Detection (AI Models)

- **Text Classifier: Fine-tuned RoBERTa / BERT or zero-shot GPT-based model**
- **Image/Video Classifier: Vision Transformer / CNN model**
- **Multimodal Fusion: Combine text + visual embeddings for final classification**
- **Output: Content flagged as "Potential Misinformation" with confidence score**

4.2 Fact-Checking (RAG + LLM)

- **RAG Module:**
  - **Vector database (FAISS, Pinecone) with verified sources: WHO, Govt sites, Fact-check databases**
  - **Retrieve top relevant sources for flagged content**

- **LLM Reasoning:**
    - **GPT / LLaMA generates human-readable verification summary**
    - **Assigns final confidence score for misinformation**

---

5. Agentic Decision Layer

- **Autonomous Actions:**
    - **Decide which flagged content requires urgent public alert**
    - **Prioritize based on Impact Score = virality potential × misinformation severity × source credibility**
- **Multi-Agent Coordination:**
    - **Detection Agent: Continuously scans and flags content**
    - **Verification Agent: Performs RAG + LLM-based fact checking**
    - **Alert Agent: Sends notifications and updates dashboard**
- **Implementation: Python with LangChain / AutoGPT framework for autonomous decision-making**
- **Logging: Each action logged for explainability and auditing**

---

6. Notification & Dashboard Layer

- **Dashboard:**
    - **Streamlit / React-based frontend**
    - **Displays flagged content, impact score, source links, confidence scores**
    - **Visualizations: trending misinformation topics, geographical spread**
- **Alerts:**
    - **Telegram bot / WhatsApp integration**
    - **Multi-lingual alerts using GenAI summaries**
    - **Users receive actionable, readable notifications**

---

7. Explainability & Feedback Layer

- **Explainability:**
  - **SHAP / LIME or LLM-generated rationale for each flagged item**
  - **Shows which signals led to detection: keywords, source reliability, sentiment, etc.**
- **Feedback Loop:**
  - **Users can confirm or dispute flagged content**
  - **Feedback used to fine-tune detection models dynamically**

---

8. Implementation Flow (Step-by-Step)

1. **Data Streaming: Collect posts/articles/videos in real-time**
2. **Preprocessing: Convert text/images/videos into embeddings**
3. **Detection: Multimodal AI classifies content as true/potential misinformation**
4. **Fact-Checking: RAG + LLM verifies flagged content against reliable sources**
5. **Impact Assessment: Calculate Impact Score and prioritize**
6. **Autonomous Alerting: Multi-agent system sends notifications + updates dashboard**
7. **Explainability: Provide reasoning and source references to users/admins**
8. **Feedback Incorporation: Fine-tune models with user feedback**

---

9. Tech Stack Summary

Layer	Technology
Backend & Orchestration	**Python, FastAPI / Flask, LangChain / AutoGPT**
Data Storage	**MongoDB (NoSQL), FAISS/Pinecone (Vector DB)**
NLP Models	**BERT, RoBERTa, GPT-4.5, LLaMA, Sentence-BERT**

Layer	Technology
Multimodal Models	**Vision Transformer, CNN, Whisper (audio)**
Frontend & Dashboard	**Streamlit / React.js, D3.js (visualizations)**
Notifications	**Telegram Bot API, WhatsApp Business API, Web Push Notifications**
Explainability	**SHAP, LIME, LLM-generated rationales**

Implementation Workflow

1. **Data Ingestion:** Collect real-time posts, articles, and multimedia content.
2. **Preprocessing:** Clean, tokenize, embed, and convert all modalities into analyzable format.
3. **Detection:** Multimodal AI classifies content as true or potential misinformation.
4. **Fact-Checking:** RAG + LLM verifies flagged content using verified sources.
5. **Impact Assessment:** Calculates Impact Score to prioritize alerts.
6. **Agentic Action:** Multi-agent system autonomously sends alerts and updates dashboard.
7. **Explainability:** Provides reasoning for flagged content with sources.
8. **Feedback Loop:** Users confirm/dispute alerts; model dynamically fine-tunes.

Detailed Implementation Plan

1. Project Setup

- **Repository Structure**

```
factseeker/
  ├── backend/           # API, agent orchestration
  ├── models/            # Pre-trained AI models
  ├── data/              # Raw and processed datasets
  ├── preprocessing/     # Scripts for cleaning & embeddings
  ├── frontend/          # Dashboard / Web App
  ├── notifications/     # Telegram / WhatsApp integration
  ├── explainability/    # SHAP/LIME + LLM reasoning
  └── docs/              # Diagrams, technical docs
```

- **Environment Setup:**
  - Python 3.10+
  - Virtual environment: venv or conda
  - Install dependencies: transformers, torch, sentence-transformers, langchain, fastapi, pydantic, faiss, streamlit, shap, lime, opencv-python, whisper, requests

---

2. Data Ingestion Layer

- **Objective:** Collect posts, articles, multimedia in real-time.
- **Sources:** Twitter API, Reddit API, NewsAPI, WhatsApp (optional for MVP), pre-verified datasets.
- **Implementation:**

```python
import tweepy

client = tweepy.Client(bearer_token='YOUR_TOKEN')
query = "covid misinformation -is:retweet lang:en"
tweets = client.search_recent_tweets(query=query, max_results=100)
```

- **Storage:** MongoDB (NoSQL) for flexibility. Each document contains: content, timestamp, source, media links, metadata.

---

3. Preprocessing & Multimodal Embedding

- **Text:** Tokenization, stopword removal, lowercasing, special char removal.
- **Embeddings:** Generate using **Sentence-BERT** or **OpenAI embeddings**.
- **Images:** Preprocess with OpenCV, then embed using **Vision Transformer (ViT)**.
- **Videos/Audio:** Extract frames, transcribe speech using **Whisper**, then embed text + visual features.
- **Output:** Unified embeddings for each content piece.

```
from sentence_transformers import SentenceTransformer
model = SentenceTransformer('all-MiniLM-L6-v2')
text_embedding = model.encode("Sample text content")
```

---

4. Detection Layer – Multimodal AI

- **Text Classification:** Fine-tuned **BERT/RoBERTa** model for misinformation detection.
- **Image/Video Classification: Vision Transformer / CNN** detects misleading images or deepfakes.
- **Multimodal Fusion:** Concatenate embeddings (text + image + audio) and feed into **MLP or transformer-based classifier**.
- **Output:** Flagged content with confidence score.

---

5. Fact-Checking Layer – RAG + LLM

- **Step 1: Retrieve verified sources** from vector database (FAISS / Pinecone).
- **Step 2: LLM Summarization & Reasoning**

```
from langchain.llms import OpenAI
from langchain.chains import RetrievalQA

qa = RetrievalQA.from_chain_type(llm=OpenAI(), retriever=my_vector_db.as_retriever())
answer = qa.run("Is this tweet about covid vaccine accurate?")
```

- **Step 3:** Generate human-readable verification summary, include source links and confidence.

---

6. Agentic Decision Layer

- **Agents:**
    1. **Detection Agent:** Continuously scans data sources.
    2. **Verification Agent:** Performs RAG + LLM fact-checking autonomously.
    3. **Alert Agent:** Sends notifications based on Impact Score.
- **Impact Score Calculation:**

  Impact Score = Virality Potential × Misinformation Severity × Source Credibility

```
Impact Score = Virality Potential × Misinformation Severity × Source Credibility
```

- **Implementation:** LangChain / AutoGPT framework for autonomous multi-agent orchestration.

---

7. Notification & Dashboard Layer

- **Dashboard:** Streamlit or React.js web app. Shows:
    - Flagged content
    - Confidence score
    - Source links
    - Trending misinformation topics
- **Notifications:**
    - Telegram bot or WhatsApp Business API integration.
    - Multi-lingual GenAI summaries for accessibility.

```
import telegram
bot = telegram.Bot(token="YOUR_TELEGRAM_TOKEN")
bot.send_message(chat_id="@mychannel", text="⚠️ FactCheck Alert: ...")
```

---

8. Explainability & Feedback Layer

- **Explainability:** Use SHAP/LIME + LLM reasoning for each flagged content.
- **Feedback Loop:** Users can confirm/dispute alerts, feedback used to **fine-tune models dynamically**.

How FactSeeker Fits into the Misinformation Track

FactSeeker directly addresses the challenge of real-time detection, verification, and mitigation of misinformation. Using a multi-agent, multimodal, and autonomous AI system, it continuously scans multiple sources—including social media, news outlets, and messaging platforms—identifies misleading or false content, and verifies it against trusted knowledge bases using Retrieval-Augmented Generation (RAG) and LLMs.

The system doesn't just flag misinformation; it acts proactively by sending clear, human-friendly alerts to the public, ensuring accurate information reaches users before false narratives spread. By combining GenAI summarization, explainable AI, and autonomous decision-making, FactSeeker aligns perfectly with the Misinformation track's goal of leveraging agentic AI to manage and reduce misinformation in real-world scenarios.

In short, FactSeeker embodies the track's vision by providing a scalable, real-time, autonomous solution that protects society from the harmful effects of false information.

FactSeeker is not just an AI system; it's a **guardian of truth**, designed to combat the pervasive spread of misinformation that threatens public health, safety, and trust. In a country like India, where misinformation spreads rapidly through platforms like WhatsApp and social media, the consequences can be dire.

Real-World Impact:

- **Health Misinformation:** During the COVID-19 pandemic, false information about treatments and vaccines led to widespread panic and delayed medical interventions. PMC

- **Political Misinformation:** In the 2025 India–Pakistan conflict, both nations experienced a surge in misinformation campaigns, particularly on social media platforms, leading to heightened tensions and public unrest. Wikipedia
- **Public Safety:** Misleading videos about railway operations in Mumbai during festivals caused confusion among passengers, prompting authorities to take action against social media accounts spreading such content. The Times of India

FactSeeker's Role:

- **Multi-Agent System:** Employs multiple AI agents to autonomously detect, verify, and alert users about misinformation in real-time.
- **Multimodal Analysis:** Analyzes text, images, and videos across various platforms to identify and debunk false information.
- **Real-Time Alerts:** Sends immediate notifications to users, empowering them to make informed decisions and share accurate information.
- **Public Health Protection:** By combating health-related misinformation, FactSeeker helps prevent the spread of harmful myths and promotes timely medical interventions.
- **Community Trust:** Restores public confidence by providing reliable information and reducing the impact of fake news.

Conclusion:
FactSeeker is a **cutting-edge, autonomous, multi-agent AI system** that aligns perfectly with the **Misinformation track**. It combines **emerging technologies, multimodal capabilities, LLM understanding, and RAG verification** to deliver a real-time, trustworthy, and scalable solution. The project is designed not only to detect misinformation but to **act proactively**, educate the public, and protect communities, embodying both technological innovation and societal impact.