High Level Design Document

This High Level Design (HLD) document outlines the architecture and core design of **MediaPulse - Trending Content Analyzer**, a beginner-friendly Python tool for analyzing and visualizing trending media topics via keywords or hashtags.

Project Name

MediaPulse - Trending Content Analyzer

Purpose:
Enable users to input keywords/hashtags and view interactive charts and analytics summarizing their popularity trends in media content, with a modern, futuristic UI.

1. System Architecture Overview

Architecture Summary:
MediaPulse is a modular Python application structured as a local web or desktop app. It consists of a user interface, data handling logic, analytics, and visualization modules.

Main System Modules

Module	Role
User Interface (UI)	Accepts user input, displays charts and analytics, applies UI theme
Data Fetcher	Retrieves or generates sample trend data (mocked or public API)
Data Processor	Cleans, aggregates, and prepares data using pandas
Analytics Engine	Computes basic stats: peak, average, trend direction
Visualization Engine	Renders interactive line/bar charts (matplotlib/plotly)

2. Component Interactions

Interaction Sequence:

1. **User** enters a keyword/hashtag in the UI.
2. **UI** sends input to **Data Fetcher**.
3. **Data Fetcher** retrieves/generates time-series data.
4. **Data Processor** cleans and aggregates data.
5. **Analytics Engine** computes summary statistics.
6. **Visualization Engine** generates trend charts.
7. **UI** displays charts and analytics summary.

From	To	Purpose

User	UI	Input keyword/hashtag
UI	Data Fetcher	Request data for input
Data Fetcher	Data Processor	Provide raw data
Data Processor	Analytics Engine	Provide processed data
Data Processor	Visualization Engine	Provide processed data
Analytics/Vis.	UI	Display results

3. Data Flow Overview

Step	Data Item	Source	Destination	Description
1	Keyword/hashtag input	User	UI	User initiates analysis
2	Raw trend data	Data Fetcher	Data Processor	Fetch/generate time-series data
3	Cleaned/aggregated data	Data Processor	Analytics/Vis. Eng.	Prepare data for analytics and visualization
4	Analytics summary	Analytics Eng.	UI	Peak, average, trend direction
5	Trend chart	Vis. Engine	UI	Interactive line/bar chart

4. Technology Stack

Layer	Technology/Frameworks
Language	Python 3.x
Data Handling	pandas
Visualization	matplotlib or plotly
UI Framework	Streamlit or Flask
Data Source	Mocked CSV/JSON or public API
UI Theme	Futuristic (dark mode, neon accents)

5. Scalability & Reliability

- **Scalability:** Designed for small datasets and local use; efficient for beginner-level analytics.
- **Reliability:** Minimal dependencies, robust error handling for user input and data loading.
- **Security:** No sensitive data handled; no authentication required.
- **Maintainability:** Beginner-friendly, well-documented codebase for easy setup and extension.

End of High Level Design Document