

Project Proposal: AI-Powered Dashboard for Social Media Insights

1. Introduction

In the evolving creator economy, digital content producers need deeper tools to understand their audience beyond basic metrics like likes and views. Comment sections, especially on platforms like YouTube, hold valuable but unstructured data that often go underutilized. This project proposes the development of an AI-powered analytics platform that extracts, processes, and interprets social feedback at scale. Our platform will evolve into a centralized dashboard for cross-platform social media intelligence, serving creators, brands, and analysts alike.

2. Project Goals

Phase 1: YouTube Comment Analyzer MVP

The initial module will support YouTube video URL input and deliver three AI-backed tools: summarization, semantic search, and a conversational agent for querying comment content.

- Summarization will use LLMs like Gemini 2 Flash to generate concise summaries of video comment sections.
- Semantic search will embed comments using models like all-MiniLM-L6-v2 and retrieve top-*k* matches based on user input.
- An AI chatbot, powered by a Retrieval-Augmented Generation (RAG) pipeline, will allow users to ask natural language questions like “What did people say about the product?” or “What issues were raised in comments?”

Comments will be pulled via the YouTube Data API and processed through a FastAPI backend. Vector embeddings will be stored using Chroma or FAISS, and frontend components will be built using Next.js. Early development used Colab and Pyngrok for tunneling; later iterations moved to a local FastAPI server for stability.

Phase 2: Engagement & Preference Insight

This stage introduces correlation between comment sentiment and video performance indicators (likes, views, watch time, click-through rates). Features will include time-series sentiment tracking, keyword clustering using BERTopic, and viewer preference modeling based on video metadata like topic, duration, or upload time.

Task automation will be enabled via Celery and Redis. Visualization will be handled through Plotly or D3.js to render interactive trend lines and comparative heatmaps.

Phase 3: Personalized Creator Dashboard

A full-stack, account-based system will be deployed with OAuth2 integration for secure access to a creator’s own data. Users will receive personalized content strategy suggestions, upload timing insights, evolving sentiment graphs, and category-wise engagement summaries. JWT-based user authentication and a modular dashboard UI will support multi-channel tracking and user-specific analytics.

AutoML hooks may be introduced to let power users fine-tune their own summarizers or toxic comment detectors.

Phase 4: Cross-Platform Social Media Hub

The project will extend to support Instagram, Twitter (X), Reddit, and TikTok by integrating their respective APIs. The goal is to unify comment analysis, mention tracking, and messaging insights across all platforms. Centralized fanbase clustering and demographic-level trend detection will allow creators to compare what resonates best per platform.

Stream ingestion will scale via Kafka; custom cross-platform ML pipelines using TensorFlow or PyTorch will handle different data formats. Inter-platform similarity models will enable generalized sentiment and topic comparison across content.

Phase 5: SaaS Productization and Monetization

The platform will transition into a commercial SaaS offering with tiered pricing—freemium for individual creators, premium for agencies. It will be containerized using Docker, deployed over Kubernetes, and scaled on cloud providers like AWS or GCP. Stripe will handle payments, and the stack will support continuous integration via GitHub Actions. Metrics will be monitored with Prometheus and Grafana.

3. Future Directions

More features to be included might be:

- AI-based response suggestion tools for creators.
- Content calendar optimizers based on historical performance.
- Thumbnail and title optimizers via generative models.
- Integration with video editing platforms for content feedback loops.

Project Proposal: AI-Powered Resume Analyzer and Enhancement Platform

1. Introduction

Resumes are often the first touchpoint between a candidate and a recruiter, yet many applicants fail to present their skills effectively due to poor formatting, vague descriptions, or lack of alignment with job roles. This project proposes the development of an intelligent, AI-driven resume analyzer and enhancer that not only reviews uploaded resumes but also suggests detailed improvements, highlights inconsistencies, and actively refactors content to increase clarity and impact. The tool will cater to students, job seekers, and professionals looking to optimize their resumes for modern recruitment systems.

2. Project Goals

Phase 1: Resume Analyzer MVP

The first milestone is a working prototype that accepts a resume upload and produces a structured analysis report.

The phase will focus on enabling users to upload resumes in `.pdf` and `.docx` formats and receive structured feedback. This requires a reliable document parsing module built using libraries such as `pdfminer`, `PyMuPDF`, and `python-docx`, combined with regex-based extraction logic for common resume sections—Education, Experience, Skills, Projects, etc. Parsed text will be classified using NLP models (like `spaCy` pipelines or zero-shot classification via BART) to identify content type and location. The system will flag missing or weak sections, grammar issues, and language inconsistencies using tools like `LanguageTool` or `Grammarly API`. A rule-based scoring engine will initially provide users with section-wise scores and improvement tips.

Phase 2: AI-Powered Rewrite and Suggestions

In this phase, the platform will implement AI-driven rewrites of resume content. Bullet points in sections such as Work Experience or Projects will be analyzed and reframed using pretrained LLMs such as `GPT-4`, `Mistral-7B`, or lightweight instruct models depending on latency needs. Each bullet will be assessed for clarity, action-result structure, use of strong verbs, and measurability. If a bullet lacks impact, the system will suggest improved versions with better phrasing, relevant metrics, and stronger vocabulary. A dual-view interface will allow users to compare the original and AI-enhanced versions. Additionally, weak or generic phrases like “worked on” or “involved in” will be automatically flagged and rewritten with more assertive alternatives. Embedding-based similarity tools like `SBERT` will help identify redundant or repetitive content.

Phase 3: Tailoring and Role-Based Targeting

This phase introduces semantic alignment between the resume and a target job description. Users will upload both documents, and the system will use sentence embeddings (via `sentence-transformers`) to perform pairwise similarity analysis between resume points and job requirements. The platform will identify missing keywords, skill gaps, or mismatches and highlight them within the UI. Suggestions will be generated to rephrase or add bullets that improve alignment with the role. The system will also provide a “Match Score” to quantify how well the resume fits the job. This mechanism will allow users to iteratively tailor their resumes for each application, making them more ATS-friendly and customized. Attention will be paid to avoiding keyword stuffing by ensuring semantic appropriateness in each insertion.

Phase 4: Formatting, Visual Cohesion, and Templates

In this phase, the system will expand from textual to visual improvements. It will detect font inconsistencies, misaligned section headers, uneven spacing, and poor typographic hierarchy using `pdfplumber` and layout parsing techniques. It will also enforce standardization of bullet spacing, date alignment, and indentation across sections. Using templates rendered in LaTeX or CSS-based HTML resumes, users will be able to apply polished formats with minimal effort. The backend will offer automatic reformatting services with role-based themes — such as clean and modern formats for tech jobs or academic-style CVs for research positions. AI will assist in suggesting an ideal section order based on industry and career level, optimizing for readability and scannability by recruiters.

Phase 5: Personal Career Dashboard and SaaS Model

The final phase involves transitioning the platform into a secure, scalable SaaS product with account-based resume tracking. Users will be able to save multiple versions, access version history, and download customized resumes. The backend will be developed using `FastAPI` or `Django`, with authentication via `OAuth2` and role-based user access. A resume improvement history dashboard will visualize changes over time, allowing users to view progress and track which suggestions were implemented. Payment integration through Stripe will enable a tiered model: free users can access basic analysis, while premium users receive unlimited AI rewrites, JD tailoring, and exportable templates. The system will be containerized using Docker and orchestrated with Kubernetes for scalable deployment on cloud platforms like AWS or GCP. Monitoring and observability will be implemented using Prometheus and Grafana to ensure service reliability.

3. Future Enhancements

Planned upgrades might include:

LinkedIn sync for automatic resume generation

Portfolio and project auto-linking from GitHub or Behance

Soft skill identification and tone tuning based on target company culture

Interview Q&A generator based on resume content

Project Proposal: AI-Powered Client Intelligence System

1. Introduction

In sales, consulting, placement, or outreach roles—the quality of preparation before a call often determines its success. Despite having access to data, employees waste considerable time manually searching through LinkedIn profiles, company pages, news articles, and social media to understand their clients or candidates. This project proposes an AI-based assistant that automatically scrapes and analyzes publicly available web data—including LinkedIn—and generates a compact, goal-oriented briefing before each client interaction. The vision is to build a lightweight intelligence platform that acts like a digital research intern. Employees will receive a summarized profile of the client, including work history, company background, recent updates, notable publications or activities, and suggestions for how to steer the conversation toward a desired goal. This tool will streamline meeting preparation and raise the quality of B2B or career-related conversations through actionable intelligence.

2. Project Goals

Phase 1:

The system's backend will include a modular scraping layer capable of extracting relevant data from:

- LinkedIn profiles (via Selenium/Puppeteer with session authentication)
- Google Search results (scraped top N snippets for relevance)
- Company websites and employee directories
- News aggregators like Google News and NewsAPI
- GitHub, Medium, and X/Twitter (when relevant)
- Email/calendar invite content for client name/URL extraction
-

Each scraper will perform structured parsing using BeautifulSoup and/or Regex to isolate semantically meaningful content blocks. Scraped content will be tokenized and passed to preprocessing pipelines for filtering and ranking. Since LinkedIn scraping can involve dynamic loading and authentication, an internal cookie/session manager will maintain safe, throttled sessions for scraping under user-logged-in context.

Phase 2:

Once raw textual data is gathered, the AI layer will summarize, rephrase, and organize it into the following components:

- Personal Profile Summary (designation, location, education, previous roles)
- Professional Background (company, recent projects, notable mentions)
- Contextual Insights (articles authored, posts interacted with, event appearances)
- Detected Interests and Keywords (derived from profile and content)
- Recent public statements, tweets, or posts (time-filtered relevance)
-

The summarization and NER (Named Entity Recognition) will use Hugging Face transformers—BART or T5 for general summarization, spaCy for structure extraction, and SBERT for semantic ranking of key snippets. Context clustering will be implemented using BERTopic to group related text chunks for clarity.

Phase 3:

One of the key modules will be a pitch assistant that lets users upload their own documents—like resumes, project summaries, or a call goal—and generates smart, personalized conversation starters. For example:

- If a user uploads a product brochure, the system can craft relevant ways to introduce it in context to the client's needs or interests.
- If a student uploads a resume and chooses “networking for internship” as the intent, the tool generates suggested opening questions and talking points tailored to the target person's profile.

This phase will use OpenAI's GPT-like architecture or Gemini APIs via RAG (Retrieval-Augmented Generation), enabling document-grounded response generation. A ranking system will score conversation options on how goal-aligned and non-generic they are.

Phase 4

The platform will be delivered as:

- A web dashboard (React-based) where users can enter names or URLs and upload documents
- A browser extension that auto-generates summaries when hovering over LinkedIn profiles or Gmail invite names
- Optional Slack/Gmail plugin that responds with precall summaries based on calendar invite parsing

Backend APIs can be built using FastAPI or Flask. Async workers via Celery will perform background scraping. Data will be temporarily cached in MongoDB or Redis for fast repeated access.

3. Additional Features and Future Expansion

- Resume-to-call optimizer: AI will highlight shared topics or career links between client and user, suggesting bridges
- Font/formatting fixer for resumes and documents before upload
- AI “Improve” button to rewrite questions, talking points, or pitch lines in more persuasive ways
- Timeline overview of a person's professional life, combining multiple sources into a chronological view
- Call recording integration (with consent) for post-call analysis and refinement suggestions

4. Use Cases

- Sales teams prepping for cold calls or demos
- Placement cells/recruiters scheduling candidate interviews
- Founders/marketers targeting potential collaborators or clients
- Students preparing to network at conferences or LinkedIn events