use case 1: Trend Analysis

- **scraping real-time trending data** from TikTok and X (create accounts ⇒ scrap data => easier doesn’t require api key)

- **ETL Pipeline** : Google Vertex AI Pipelines (For scalable AI workflow execution)

- **Trend Identification / Analysis** :
  - NLP Models (Sentiment + Topic Detection)
    - BERT / FinBERT / LLaMA / GPT-4 API (for trend classification & sentiment analysis)
    - Hugging Face Transformers (for pretrained models)

- **Automated Content Generation**:

- Text Generation
  - OpenAI GPT-4 / Google Gemini (for writing viral captions & tweets)
- Image Generation
  - Stable Diffusion / DALL·E (for AI-generated images)
- Video Generation
  - RunwayML / Synthesia (for AI-powered video creation)

- **AI Deployment :**

- Google Vertex AI (For training & deploying ML models)
- FastAPI / Flask (To expose AI models as APIs)

- **Automation & Scheduling :**

- N8N / Zapier (For workflow automation)
- Celery + Redis (For async task scheduling)

- **Frontend / Dashboard (for monitoring trends)**

- Streamlit / Gradio (If you need a simple UI for visualization)
- React + Next.js (For a full web dashboard)

Some examples for some tools : (you can also benefit from github student pack , there is so many free resources and alternatives that will help you through out the dev)

Source	Free Tools
TikTok Trends	`TikTok Unofficial API` ([Python Scraper])
Google Trends	`Tweepy` (Limited Free Access)
X (Twitter) Trends	`pytrends` (Python Google Trends API)
Web Scraping	`BeautifulSoup`, `Selenium`, `Scrapy`

Note: Combine **Tweepy + TikTok Scraper + Pytrends** for cost-free real-time data

Task	Open-Source Models
NLP Trend Analysis	`BERT`, `T5`, `Llama 2` (Hugging Face)

Sentiment Analysis	VADER, TextBlob, FinBERT
Time Series Forecasting	Facebook Prophet, ARIMA, LSTM

Type	
Text (Captions, Tweets)	GPT-2, Llama-2-7B (Meta),OpenAI GPT-4 API
Images (Memes, Thumbnails)	Stable Diffusion (Local)
Video (AI-Generated Content)	RunwayML, D-ID Free Plan

Some helpful tools:

Data Collection	TikTok Scraper, Tweepy, Pytrends
Storage	PostgreSQL (Supabase), MongoDB Atlas
Trend Detection	BERT, Prophet, TextBlob
AI Content Gen	Stable Diffusion, GPT-2, Llama 2
API Hosting	FastAPI on Hugging Face
Automation	N8N, Celery + Redis
Frontend	Streamlit, Gradio

some helpful links :

TikTok Scraper	🔗 TikTok-Api

Tweepy	🔗 Tweepy GitHub
Google Trends API	🔗 Pytrends GitHub

Stable Diffusion	Generate viral images/memes	🔗 SD GitHub
TTS & AI Voices	Convert text to viral video narration	🔗 TTS GitHub
RunwayML	AI-generated videos	🔗 RunwayML
DeepFaceLab	AI deepfake videos (for viral engagement)	🔗 DeepFaceLab GitHub

LangChain	Framework for AI agents using LLMs (GPT, Llama)	🔗 LangChain GitHub
AutoGPT	Self-improving AI agent	🔗 AutoGPT GitHub
GPT-Engineer	AI-driven code generation agent	🔗 GPT-Engineer GitHub
OpenAgent	Framework for building AI assistant	🔗 OpenAgent GitHub

use case 2: On-Chain Tracking

Develop blockchain-based analytics for user behavior or token performance and use AI to create charts and infoposts about your on chain analysis

Tools: Dione Protocol SDK, ETH Scan, SOL Scan, centralized exchange APIs and others as well as N8N and other automation tools

1. Data Extraction (Blockchain & CEX APIs)

- **Ethereum & EVM Chains:** Etherscan API, Infura, Alchemy, Dione Protocol SDK

- **Solana & Non-EVM Chains:** SolScan API, Solana RPC, The Graph
- **Cross-Chain & CEX Data:** Binance API, CoinGecko API, Chainlink, DefiLlama API

2. AI Analytics & Visualization

- **Processing:** Python (Pandas, NumPy), Google Colab, TensorFlow, OpenAI API
- **Visualization:** Matplotlib, Seaborn, Plotly, D3.js, Chart.js
- **Automation:** N8N, Zapier

3. Smart Contract Interaction

- **Languages:** Solidity (Hardhat), Rust (Anchor for Solana)
- **APIs & SDKs:** Web3.js, Ethers.js, Solana Web3.js, The Graph

4. Frontend (Optional Analytics Dashboard)

- **Tech:** React.js, Next.js, TailwindCSS, Material UI

5. Deployment

- **Cloud & Hosting:** AWS, Google Cloud, Vercel
- **Decentralized Storage:** IPFS, Arweave

use case 3:Sustaiability AI Agent

Scraping

- **Scrapy / BeautifulSoup** (Python) – Web scraping
- **Selenium / Puppeteer** – JavaScript-rendered content scraping
- **APIs** – GraphQL for Ethereum-based projects, ESG databases
- **IPFS/Arweave Access** – If projects store documents on decentralized storage

Natural Language Processing (NLP) Analysis

- **Purpose**: Identify and evaluate sustainability claims, keywords, and sentiment in extracted data.
- **Technologies**:
  - **DeepSeek, ChatGPT, or Llama-based models** – NLP for text understanding

    - **spaCy & NLTK –** Text preprocessing (tokenization, named entity recognition)
    - **BERT/RoBERTa –** ESG keyword classification
    - **FinBERT** – Sentiment analysis for financial sustainability statements

ESG Scoring Model

- **Purpose**: Evaluate sustainability metrics based on predefined ESG (Environmental, Social, Governance) criteria.
- **Key ESG Metrics for Crypto Projects**:
    - **Energy Consumption –** Blockchain energy usage (Ethereum vs. Layer 2)
    - **Carbon Offsets** – Use of carbon credits, partnerships (KlimaDAO)
    - **Governance** – Decentralization score, DAO governance structure
    - **Transparency** – Frequency of sustainability updates, partnerships
- **Technologies**:
    - **AI-driven ESG Scoring Models –** Custom ML models trained on sustainability reports
    - **Explainable AI (XAI)** – SHAP, LIME for ESG decision transparency
    - **Data Sources**: Carbon tracking APIs (e.g., Greenly, KlimaDAO)

Scoring System & Report Generation

- **Purpose**: Generate sustainability scores and insights for Ethereum-based projects.
- **Technologies**:
    - **Scikit-learn / XGBoost** – For ESG score modeling
    - **Power BI / Tableau** – Data visualization dashboards
    - **Matplotlib / Plotly** – Visualizing ESG score trends
    - **Markdown/PDF Report Generation** – Automated ESG reports

Automated Alerts & Tracking System

- **Purpose**: Monitor and alert on changes in sustainability statements, ESG policies, and project updates.
- **Technologies**:
    - **Cron Jobs / Cloud Functions** – Scheduled re-scraping and analysis
    - **Webhook Integrations** – Alerts via Telegram, Slack, or Email
    - **Vector Databases (FAISS, Pinecone)** – Tracking document changes over time

Web Scraping & Data Extraction

Web Scraping	Scrapy/BeautifulSoup/	

JavaScript Scraping	Puppeteer	Headless browser for automated scraping
Web Automation	Selenium	Automates browser interactions
API Data	GraphQL Explorer	Fetch blockchain & Web3 project data
IPFS Access	IPFS.js / Py-IPFS	Access decentralized storage

Automated Alerts & Tracking

Automation	Schedule automated ESG scrapers//Cron Jobs / Airflow	Apache Airflow
Notification APIs	Telegram Bot API/Slack API	
Document Change Tracking	Pinecone / FAISS	Vector databases for ESG text updates FAISS

Scraping webpages of different L1 blockchains - go