

Prompt 1 — Connecting Multiple APIs for Unified Data

"You are an **API Integration Engineer** with expertise in REST (Representational State Transfer) and GraphQL APIs. I want to create a single automated workflow that combines data from Google Sheets, Shopify, and Google Analytics APIs.

Your task:

1. Authenticate each API using OAuth 2.0 and store tokens securely.
2. Pull product sales data from Shopify, website traffic data from Google Analytics, and inventory data from Google Sheets.
3. Merge the datasets on product ID and date fields for unified reporting.
4. Schedule the workflow to run daily at midnight using a cron job or cloud scheduler.
5. Include logging for errors and success status.

Output format: Python script with step-by-step API calls, merged dataset output as CSV, and instructions for deployment in a cloud environment (AWS Lambda or Google Cloud Functions)."

Prompt 2 — Automating Social Media Posting via API

"You are a **Social Media Automation Specialist** experienced with the Meta Graph API, LinkedIn API, and Twitter API (now X API). I manage 3 platforms and want to post the same content automatically at scheduled times.

Your task:

1. Authenticate all platform APIs and handle rate limits.
2. Create a reusable function that posts text, images, and videos from a single JSON file.
3. Add an option to customise captions per platform for optimal engagement.
4. Schedule posts using a job scheduler (like APScheduler in Python).
5. Log post IDs and engagement metrics for tracking.

Output format: Python automation script + setup instructions + sample JSON template for post content."

Prompt 3 — Automating Data Entry from Web Forms to CRM

"You are a **CRM Workflow Automation Engineer** skilled in HubSpot, Salesforce, and Zoho CRM APIs. I receive customer leads from a website form and want them automatically added to my CRM with tags for campaign tracking.

Your task:

1. Set up a webhook to receive form submissions in real time.
2. Transform form data into the CRM's required JSON format.
3. Use the CRM API to create a new lead record with appropriate tags (e.g., "WebForm2024").
4. Send a confirmation email to the lead using the CRM's email API.
5. Log each successful lead creation in a Google Sheet via API.

Output format: API workflow diagram + example webhook handler code + CRM API call scripts."

Prompt 4 — Email Marketing Automation with API

"You are an **Email Marketing Automation Expert** familiar with Mailchimp, SendGrid, and ActiveCampaign APIs. I want to send a weekly newsletter automatically using my Google Sheets contact list.

Your task:

1. Connect Google Sheets API to read subscriber list.
2. Connect the chosen email service API and authenticate securely.
3. Pull the email template from a stored HTML file.
4. Send emails in batches to avoid exceeding API rate limits.
5. Update the Google Sheet with a "last sent" timestamp for each contact.

Output format: Python/Node.js script + deployment guide + API keys & secrets handling instructions."

Prompt 5 — Automating File Backups to Cloud Storage

"You are a **Cloud Automation Engineer** skilled in AWS S3, Google Drive, and Dropbox APIs. I have a folder on my local machine containing financial reports that must be backed up daily to all three cloud platforms.

Your task:

1. Authenticate with all three cloud APIs.
2. Compress the local folder into a timestamped ZIP file.
3. Upload the ZIP file to each cloud platform.
4. Send an email notification with file URLs after successful upload.
5. Log the backup details in a CSV file for auditing.

Output format: Shell/Python script + setup instructions + automation scheduling plan."

Prompt 6 — Real-Time Stock Price Tracker with Alerts

"You are a **Financial Data Automation Specialist** experienced in Alpha Vantage, Yahoo Finance, and TradingView APIs. I want to track live stock prices for a watchlist of 10 Indian companies and receive alerts when prices change more than $\pm 3\%$ in a day.

Your task:

1. Connect to the stock price API with authentication.
2. Create a script to fetch and store real-time prices every 5 minutes.
3. Compare the current price with the opening price for percentage change.
4. Trigger an email/SMS alert when the $\pm 3\%$ threshold is crossed.
5. Store all intraday data in a CSV for end-of-day analysis.

Output format: Python script + CSV logging + alert system integration plan.

Input Files & Code Section:

- API Key file (api_keys.json) for Alpha Vantage/Yahoo Finance.
- watchlist.csv containing company ticker symbols.
- Placeholder for email/SMS sending function."

Prompt 7 — Automating PDF Invoice Creation from Sales Data

"You are a **Document Automation Engineer** skilled in ReportLab, wkhtmltopdf, and Google Docs API. I have daily sales data in CSV format and need automatically generated PDF invoices sent to customers.

Your task:

1. Read the CSV to fetch customer details, products, and prices.
2. Generate a branded PDF invoice for each customer.
3. Save the invoice locally and in Google Drive.
4. Email the invoice to the customer with a personalised message.
5. Log invoice status (sent, pending, failed) in a Google Sheet.

Output format: Python script + invoice PDF template + Google Drive integration guide.

Input Files & Code Section:

- sales_data.csv with customer and order details.
- invoice_template.html for branding.
- API credentials for Google Drive and Gmail."

Prompt 8 — Weather-Based Automation for Agriculture

"You are an **Agricultural IoT Automation Specialist** skilled in OpenWeatherMap API and smart irrigation systems. I want to automate irrigation based on real-time weather data.

Your task:

1. Connect to the OpenWeatherMap API to fetch daily forecasts.
2. If rainfall probability is >70%, delay irrigation by 24 hours.
3. If temperature >35°C, schedule an extra watering cycle.
4. Send an SMS to the farmer confirming the decision.
5. Log all actions in a daily report file.

Output format: IoT control script + weather API integration + action logging.

Input Files & Code Section:

- API key file for OpenWeatherMap.
- farm_config.json with field size, crop type, and irrigation limits.

Placeholder for SMS gateway integration code."

Prompt 9 — Automating YouTube Video Uploads

"You are a **YouTube API Automation Specialist**. I want to upload videos from a folder to YouTube with titles, descriptions, and tags automatically pulled from a CSV file.

Your task:

1. Authenticate using YouTube Data API v3 with OAuth 2.0.
2. Loop through a folder containing video files.

3. Read metadata from a CSV (title, description, tags, privacyStatus).
4. Upload each video with the corresponding metadata.
5. Log upload IDs and publish status.

Output format: Python script + CSV metadata mapping + OAuth setup guide.

Input Files & Code Section:

- video_metadata.csv with columns for each video.
- Folder path for video files.

client_secret.json for OAuth credentials."

Prompt 10 — Daily Currency Conversion Automation

"You are a **Currency Data Automation Specialist**. I want to fetch daily INR to USD, EUR, and GBP exchange rates and update them in my Google Sheet automatically.

Your task:

1. Connect to a currency exchange API (e.g., ExchangeRate-API).
2. Fetch latest conversion rates for INR to target currencies.
3. Write data to a specific Google Sheets cell range.
4. Include timestamp of last update.
5. Schedule script to run daily at 8 AM IST.

Output format: Python script + Google Sheets API integration + scheduler setup guide.

Input Files & Code Section:

- API key file for ExchangeRate-API.
- Google Sheets spreadsheet ID.

config.json for target currency list."

Prompt 11 — Automating Job Application Tracking

Backstory: You're a 28-year-old marketing professional applying to multiple companies at once. Keeping track of applications manually is messy — you often forget where you applied, the status, or the interview schedule. You want AI and APIs to track everything automatically.

Goal: Build an automation that pulls application data from job portals (LinkedIn, [Naukri.com](#)) and updates it into a single Google Sheet dashboard daily.

Prompt:

"You are a **Job Search Workflow Automation Engineer**. I want an automated job application tracker that consolidates applications from LinkedIn Jobs and [Naukri.com](#) using their APIs/webhooks.

Your task:

1. Authenticate with LinkedIn API and [Naukri.com](#)'s developer API (or scrape data if no API exists).

2. Fetch job title, company name, date applied, status (applied, shortlisted, interview scheduled), and job link.
3. Push this data into a Google Sheet in structured columns.
4. Highlight rows where the application has been idle for >14 days.
5. Send me a daily email digest of new application updates.

Output format: Google Sheet dashboard + email digest example + API scripts.

Input Files & Code Section:

- API credentials for LinkedIn and Naukri.com
- Google Sheet ID and credentials JSON
- Email SMTP settings for sending daily digest"

Prompt 12 — Automating Property Price Tracking for Investment

Backstory: You're a 35-year-old professional looking to invest in property in Bangalore. Prices change fast and manual tracking is too slow. You want a tool that automatically fetches and compares prices across multiple real estate portals.

Goal: Build a daily property price tracker with alerts for deals under your budget.

Prompt:

"You are a **Real Estate Data Automation Specialist** skilled in integrating housing.com, magicbricks.com, and 99acres.com APIs.

Your task:

1. Fetch property listings for specified locations (e.g., Whitefield, Indiranagar) within a budget range.
2. Extract details — price, size (sqft), price per sqft, location link.
3. Store data in a Google Sheet with a "lowest price this week" column.
4. Trigger an SMS alert when a property price drops more than 5% from last week.
5. Generate a weekly PDF market trend report.

Output format: Google Sheet tracker + automated PDF report + SMS alert script.

Input Files & Code Section:

- API credentials or scraping script for property portals
- property_config.json with budget, preferred locations, size range

Google Sheets & Twilio SMS API credentials"

Prompt 13 — Automating Invoice Payment Reminders

Backstory: You run a small design agency. Clients often delay payments, and manually sending reminders eats up your evenings. You want an automated reminder system that sends polite follow-ups.

Goal: Build an API-based automation that sends reminders at 7, 14, and 21 days after invoice due date.

Prompt:

"You are a **Business Workflow Automation Specialist**. I want to automate client payment reminders using QuickBooks API and Gmail API.

Your task:

1. Pull unpaid invoice data from QuickBooks API with due dates.
2. Identify invoices past due by 7, 14, or 21 days.
3. Send a customised reminder email based on how late the payment is.
4. Log all sent reminders in a Google Sheet.
5. Mark the invoice in QuickBooks with “reminder sent” status.

Output format: Automated reminder script + email template files + logging spreadsheet.

Input Files & Code Section:

- QuickBooks API credentials
- Google API credentials for Gmail & Sheets



email_templates/ folder with HTML templates for 7, 14, 21 days"

Prompt 14 — Automating Resume Screening for Recruitment

Backstory: You're an HR manager for a startup. Hundreds of resumes arrive daily. Manually screening them for skills is impossible. You need an API workflow that filters CVs based on required skills.

Goal: Automatically screen resumes and send shortlisted profiles to a hiring manager's email.

Prompt:

"You are a **Recruitment Automation Specialist**. I want to integrate Google Drive API and an NLP model to process incoming resumes.

Your task:



1. Monitor a Google Drive folder for new resumes.
2. Extract text from PDFs/DOCs using an OCR/NLP API.
3. Match candidate skills with a given job description using keyword matching and semantic similarity.
4. Move shortlisted resumes to a “Shortlisted” folder.
5. Email a daily summary to the hiring manager with names and matched skills.

Output format: Resume screening script + summary email template + candidate matching report.

Input Files & Code Section:

- Google Drive API credentials
- Job description text file
- API key for NLP/OCR service (e.g., Google Cloud Vision, OpenAI)"

Prompt 15 — Automating YouTube Comment Sentiment Analysis

Backstory: You're a content creator with 500K subscribers. It's impossible to read every comment and spot trends in audience sentiment.

Goal: Build an API workflow that pulls all new comments, runs sentiment analysis, and gives you a weekly trend report.

Prompt:

"You are a **Social Media Analytics Automation Engineer**. I want a system that fetches my YouTube video comments weekly, analyses sentiment, and creates a dashboard.

Your task:

1. Connect to YouTube Data API to fetch comments for all videos from the last 7 days.
2. Run sentiment analysis using a pre-trained model (e.g., VADER, BERT).
3. Categorise comments as positive, negative, or neutral.
4. Create visualisations showing weekly sentiment trends.
5. Generate a PDF report and store it in Google Drive.

Output format: Sentiment analysis notebook + dashboard + weekly PDF.

Input Files & Code Section:

- YouTube API credentials
- Sentiment analysis model file or package requirements
- Google Drive API credentials"

Prompt 16 — Automating E-commerce Inventory Updates Across Platforms

Backstory: You sell products on Amazon, Flipkart, and your own Shopify store. Inventory changes fast, but updating each platform manually wastes hours and risks overselling.

Goal: Build an API automation that updates inventory levels across all platforms from a single source.

Prompt:

"You are an **E-commerce API Integration Specialist**. I want a single source of truth for my inventory, updated across Amazon, Flipkart, and Shopify in real time.

Your task:

1. Connect to all three platform APIs using secure authentication (API keys or OAuth).
2. Fetch the latest inventory count from my central warehouse database or Google Sheet.
3. Update product stock levels on each platform.
4. Send me an email if a product's stock falls below a reorder threshold.
5. Log all updates with timestamp, product ID, and before/after quantities.

Output format: Inventory sync script + alert email template + update log file.

Input Files & Code Section:

- API credentials for Amazon, Flipkart, and Shopify
- inventory.csv or database connection details

Email SMTP settings for low-stock alerts"

Prompt 17 — Automating Customer Support Ticket Categorisation

Backstory: Your startup gets 200+ support emails daily. Agents waste time reading and assigning tickets manually.

Goal: Use APIs and AI to automatically categorise tickets and assign them to the right team.

Prompt:

"You are a **Customer Service Workflow Automation Engineer**. I want to integrate Gmail API, NLP (Natural Language Processing), and a ticketing system API (like Zendesk).

Your task:

1. Fetch new support emails via Gmail API.
2. Run NLP classification to detect category (Billing, Technical Issue, General Query, Complaint).
3. Create a ticket in Zendesk with the detected category.
4. Assign tickets to the relevant department queue.
5. Send an auto-response email to the customer with an estimated resolution time.

Output format: Categorisation script + Zendesk integration + auto-reply email templates.

Input Files & Code Section:

- Gmail API credentials
- NLP model or keyword mapping file
- Zendesk API credentials"

Prompt 18 — Automating Daily Stock Market Newsletter

Backstory: You run a Telegram channel for stock market updates. Manually collecting news, stock prices, and analysis every morning is slow.

Goal: Generate and send a daily market summary via email and Telegram using APIs.

Prompt:

"You are a **Financial Automation Developer**. I want a daily 7:30 AM IST newsletter combining stock prices, market news, and a short AI-generated analysis.

Your task:

1. Connect to Yahoo Finance API for NIFTY 50, SENSEX, and top 10 stocks data.
2. Pull top 5 market news headlines from News API.
3. Use GPT API to generate a 150-word market analysis.
4. Send the report via Gmail API and post to a Telegram channel via Telegram Bot API.
5. Store all reports in a Google Drive folder for archiving.

Output format: Automated newsletter script + Telegram bot setup + daily report template.

Input Files & Code Section:

- Yahoo Finance API key
- News API key
- OpenAI GPT API key

Gmail API and Telegram Bot credentials"

Prompt 19 — Automating Attendance Tracking with Face Recognition

Backstory: Your office wants to replace manual attendance sheets with automated facial recognition connected to HR software.

Goal: Build a system that captures attendance via webcam and updates HR records automatically.

Prompt:

"You are an **AI-Driven HR Automation Specialist**. I want a face recognition attendance tracker that integrates with Zoho People API.

Your task:

1. Connect a webcam to capture employee images at check-in/check-out.
2. Run face recognition using an API like AWS Rekognition or OpenCV.
3. Match recognised faces to employee IDs.
4. Update attendance in Zoho People API.
5. Send a daily attendance summary to HR.

Output format: Attendance capture script + Zoho API integration + HR report template.

Input Files & Code Section:

- Zoho People API credentials
- Employee ID to face mapping database
- Webcam access permissions and recognition API credentials"

Prompt 20 — Automating Podcast Transcription and Upload

Backstory: You run a podcast and need transcripts for SEO and accessibility. Doing it manually takes hours.

Goal: Use APIs to transcribe each new episode and upload the text to your blog automatically.

Prompt:

"You are a **Content Automation Engineer**. I want an automation that listens for new podcast episodes, transcribes them, and publishes to my WordPress blog.

Your task:

1. Monitor an RSS feed for new podcast episodes.
2. Download the audio file.

3. Use AssemblyAI or Google Speech-to-Text API for transcription.
4. Format the transcript into a blog-friendly HTML format.
5. Upload it as a new blog post via WordPress REST API.

Output format: End-to-end transcription and upload script + blog post HTML template.

Input Files & Code Section:

- Podcast RSS feed URL
- Transcription API key

WordPress API credentials"

Prompt 21 — Automating Business KPI Dashboard Updates

Backstory: You manage a startup and track sales, expenses, and customer data. You want your KPI dashboard updated automatically every morning.



Goal: Build an API workflow that pulls data from CRM, accounting software, and marketing tools into a BI dashboard.

Prompt:

"You are a **Business Intelligence Automation Specialist**. I want an automated data pipeline feeding my Power BI dashboard daily.

A SECTION 8 – NON-PROFIT ORGANIZATION REGISTERED UNDER THE COMPANIES ACT, 2013

Your task:

1. Fetch sales data from CRM API (HubSpot or Salesforce).
2. Fetch expenses from accounting API (QuickBooks or Zoho Books).
3. Fetch campaign performance from Google Ads API.
4. Push all data to a Power BI dataset via REST API.
5. Refresh dashboard daily at 7 AM IST.

Output format: ETL (Extract, Transform, Load) script + Power BI dataset refresh automation.

Input Files & Code Section:

- CRM API credentials
- Accounting API credentials
- Google Ads API credentials
- Power BI API token"

Prompt 22 — Automating Legal Document Generation

Backstory: You're a lawyer preparing NDAs, contracts, and agreements for clients. Filling them manually is slow.

Goal: Build an API automation that fills in legal document templates from client data.

Prompt:

"You are a **Legal Tech Automation Specialist**. I want an API-based system that populates legal document templates from a client database.

Your task:

1. Store client details (name, address, contract terms) in a Google Sheet or database.
2. Pull data via API and inject into pre-defined Word/PDF templates.
3. Save final documents in Google Drive and send via Gmail API.
4. Track sent documents in a log sheet.
5. Allow re-generation if client data changes.

Output format: Document automation script + legal template folder + logging sheet.

Input Files & Code Section:

- Document templates (Word/PDF)
- Google Sheets API credentials



Google Drive & Gmail API credentials"

Prompt 23 — Automating Food Delivery Order Processing

Backstory: You run a cloud kitchen. Orders from Zomato, Swiggy, and your own website come separately, causing delays.

Goal: Build an API integration that merges all orders into one system.

Prompt:

"You are a **Food Tech API Integration Specialist**. I want a centralised order management system pulling data from Zomato, Swiggy, and my website.

Your task:



1. Connect to all order APIs with authentication.
2. Merge incoming orders into one dashboard view.
3. Send order confirmation to customers via SMS API.
4. Trigger kitchen ticket printing via printer API.
5. Store all order data for monthly analysis.

Output format: Order aggregation script + kitchen display dashboard + SMS integration.

Input Files & Code Section:

- API keys for Zomato, Swiggy, website
- SMS API credentials

Database connection for order storage"

Prompt 24 — Automating Social Media Comment Replies

Backstory: You run a brand page with thousands of comments daily. Replying manually takes too long.

Goal: Build a system that auto-replies to comments based on sentiment and keywords.

Prompt:

"You are a **Social Media Engagement Automation Engineer**. I want to use Instagram Graph API and NLP to auto-reply to comments.

Your task:

1. Fetch new comments via Instagram API.
2. Run keyword & sentiment analysis to classify the comment.
3. Use a pre-defined reply template for each sentiment type.
4. Post the reply via API.
5. Log all replied comments in Google Sheets.



Output format: Comment reply automation script + sentiment keyword mapping + logging sheet.

Input Files & Code Section:

- Instagram Graph API credentials
- Keyword mapping CSV
- Google Sheets API credentials"

Prompt 25 — Automating YouTube to Instagram Clip Conversion

Backstory: You want to post highlights of your YouTube videos on Instagram Reels automatically.

Goal: Build an API workflow that trims, captions, and uploads clips from YouTube to Instagram.

Prompt:

"You are a **Video Content Automation Specialist**. I want to pull my latest YouTube videos, create 60-second highlights, auto-caption them, and upload to Instagram.

Your task:

1. Fetch video from YouTube Data API.
2. Trim to highlight section based on timestamps from a CSV.
3. Add captions using an API like [Rev.ai](#).
4. Upload to Instagram via Instagram Graph API.
5. Store uploaded video link in a Google Sheet.

Output format: Video processing script + Instagram upload automation + logging system.

Input Files & Code Section:

- YouTube API credentials
- Instagram Graph API credentials
- clip_timestamps.csv with video ID and time ranges"



TECH LEGAL AWARENESS FORUM

A SECTION 8 – NON-PROFIT ORGANIZATION REGISTERED UNDER THE COMPANIES ACT, 2013

SCAM **FREE INDIA**
CYBER ALERT