

EasyMy Learning Pvt. Ltd.

Prompt Book 2: IT, Software, Data and AI

A comprehensive collection of expert-crafted AI prompts for IT professionals, developers, data engineers, and AI enthusiasts.

An Initiative by EasyMy Learning Pvt. Ltd. to empower Nepalese students with AI education and digital literacy.

© 2025 EasyMy Learning Pvt. Ltd. | All Rights Reserved

Coding & Debugging

1. Prompt 1 — Code Review & Optimisation Plan

Code Review & Optimisation Plan "You are a senior software engineer with expertise in [programming language/framework]. I have a codebase for a [type of application — e.g., e-commerce site, chatbot, mobile app] that works but runs slowly. Review the code for: Inefficient loops or redundant logic. Poor memory management. Opportunities to replace custom code with standard libraries. Security vulnerabilities (SQL injection, XSS). Best practices for scalability. Provide output in 2 sections: a table of issues (with line numbers & problem description) and an optimised code snippet for each fix."



2. Prompt 2 — Bug Reproduction & Fix Documentation

Bug Reproduction & Fix Documentation "You are a QA (Quality Assurance) automation tester and developer. I have a bug where [describe bug behaviour] in my [framework/app type]. Create a debugging report that includes: Exact steps to reproduce the bug. Screenshots or logs showing the issue. The suspected root cause in the code. The corrected code segment. Unit test cases to ensure the bug doesn't reappear. Output in debugging report format with code blocks."



3. Prompt 3 — Multi-Language Code Conversion

Multi-Language Code Conversion "You are a cross-platform developer. Convert my [programming language] code for [feature/function] into [target language], ensuring: Exact feature parity. Proper syntax & library usage in the target language. Equivalent performance or better. Inline comments explaining logic. A quick performance test script in the target language. Output in side-by-side original vs converted code format."



4. Prompt 4 — Algorithm Efficiency Upgrade

Algorithm Efficiency Upgrade "You are a competitive programming expert. My current [sorting/searching/matching/etc.] algorithm in [language] works but is slow on large datasets. Optimise it by: Suggesting a faster algorithm (e.g., replacing bubble sort with merge sort). Explaining time & space complexity differences. Providing optimised code. Writing performance benchmarks comparing both versions. Output in comparative performance report format."



5. Prompt 5 — Debugging API Integration Issues

Debugging API Integration Issues "You are an API integration specialist. My app in [language/framework] fails to fetch data from [API name] correctly. Diagnose the problem by: Reviewing my API request code. Checking authentication & endpoint issues. Suggesting correct request format with example. Adding error handling for network failures. Writing a test function to validate the fix. Output in debugging log + corrected code block."



6. Prompt 6 — Code Refactoring for Readability & Maintainability

Code Refactoring for Readability & Maintainability "You are a senior code architect. Refactor my [language/framework] code for [feature/function] so it is: Easier to read with proper indentation & naming conventions. Modularised into reusable functions or classes. Documented with meaningful comments. Compliant with [specific coding standard, e.g., PEP8 for Python]. Unit tested to ensure functionality remains unchanged. Output in before/after code comparison format with a change log."



7. Prompt 7 — Legacy Code Modernisation Plan

Legacy Code Modernisation Plan "You are a legacy systems upgrade specialist. My [old programming language or framework] code needs to be upgraded to [modern equivalent] while preserving functionality. **Include:** Identification of outdated functions/libraries. Recommended replacements & modern equivalents. Compatibility concerns with the new environment. Performance benefits after modernisation. Step-by-step migration plan with testing checkpoints. Output in legacy-to-modern migration report format."



8. Prompt 8 — Continuous Integration (CI) Debugging Setup

Continuous Integration (CI) Debugging Setup "You are a DevOps engineer. Configure a CI pipeline for my [language/framework] project to automatically run code linting, unit tests, and integration tests whenever code is pushed. **Include:** Recommended CI tool (GitHub Actions, Jenkins, GitLab CI). Pipeline configuration script. Test coverage reporting setup. Common CI errors & how to fix them. Notification integration (Slack/Email) for failed builds. Output in pipeline YAML file + setup guide."



9. Prompt 9 — Memory Leak Detection & Fix

Memory Leak Detection & Fix "You are a performance optimisation engineer. Analyse my [language/framework] application for memory leaks and: Identify possible causes from code structure. **Suggest** profiling tools (Valgrind, Perf, Chrome DevTools). **Provide** a step-by-step method to reproduce memory growth. Fix the leak with corrected code examples. **Suggest** long-term prevention strategies. Output in diagnostic report + corrected code samples."



10. Prompt 10 — Multi-Threading Bug Resolution

Multi-Threading Bug Resolution "You are a concurrency programming specialist. My [language/framework] application faces race conditions and deadlocks. **Include:** Detailed explanation of the issue. Steps to identify which thread is causing the block. Corrected thread-safe code. Recommended locking or async patterns. Performance benchmarks after fix. Output in bug analysis + updated code format."



11. Prompt 11 — Automated Unit Test Generation

Automated Unit Test Generation "You are a test automation engineer. Write automated unit test scripts for my [language/framework] code covering [feature/function]. **Include:** Test cases for normal, boundary, and error conditions. Assertions for expected output. Mock data creation. Code coverage percentage target (e.g., >80%). How to run tests in CI/CD pipeline. Output in ready-to-run test script format."



12. Prompt 12 — SQL Query Debugging & Optimisation

SQL Query Debugging & Optimisation "You are a database performance engineer. Optimise my slow SQL queries for [database type: MySQL, PostgreSQL, etc.]. **Include:** Query execution plan analysis. Indexing strategy. Query rewriting for speed. Caching recommendations. Before/after execution time comparison. Output in query optimisation report with revised SQL statements."



13. Prompt 13 — Cross-Browser Bug Fix Plan

Cross-Browser Bug Fix Plan "You are a front-end debugging expert. My web app works on Chrome but fails in Firefox & Safari. **Include:** List of browser compatibility issues. Code fixes using cross-browser safe APIs. CSS vendor prefixing guide. Polyfill recommendations for unsupported features. Testing checklist for all major browsers. Output in browser compatibility report + fixed code snippets."



14. Prompt 14 — API Rate Limit Error Resolution

API Rate Limit Error Resolution "You are an API performance consultant. My app hits rate limits when fetching data from [API name]. **Include:** How to detect rate limit headers. Backoff strategies & caching techniques. Batch request examples. Code modifications for retry logic. Test scenarios to confirm the fix. Output in rate limit handling guide + updated code."



15. Prompt 15 — Deployment Bug Fix Checklist

Deployment Bug Fix Checklist "You are a deployment engineer. My application works locally but fails in production. **Include:** Environment variable checks. Dependency version mismatches. Server configuration issues. Build process verification. Automated rollback setup. Output in deployment debugging checklist format."



16. Prompt 16 — Version Control Conflict Resolution

Version Control Conflict Resolution "You are a Git (Version Control) expert. Resolve merge conflicts in my [repository name] while ensuring no functionality loss. **Include:** Step-by-step conflict resolution process. Commit best practices to avoid future issues. Branch management guidelines. Git command examples for common scenarios. Output in merge resolution guide with Git commands."



17. Prompt 17 — Security Vulnerability Patch Plan

Security Vulnerability Patch Plan "You are a cybersecurity code auditor. Review my [language/framework] code for vulnerabilities like SQL injection, XSS, and CSRF. **Include:** Vulnerability list with risk levels. Code patches with secure alternatives. OWASP (Open Web Application Security Project) best practice checklist. Security testing tools list. Output in vulnerability report + patched code samples."



18. Prompt 18 — Cloud Function Debugging Guide

Cloud Function Debugging Guide "You are a cloud application developer. Debug my [AWS Lambda / Google Cloud Function / Azure Function] which is failing intermittently. **Include:** Log analysis methods. Error pattern detection. Code changes for reliability. Testing with local emulators. Deployment steps after fix. Output in debugging flowchart + corrected function code."



19. Prompt 19 — Mobile App Crash Analysis

Mobile App Crash Analysis "You are a mobile app debugging expert. Analyse my [Android/iOS] app for crash reports related to [feature/function]. **Include:** Crash log interpretation. Root cause identification. Code corrections. Device-specific fixes. Testing plan for all OS versions. Output in crash report + updated code block."



20. Prompt 20 — Real-Time Error Monitoring Setup

Real-Time Error Monitoring Setup "You are a site reliability engineer. Set up a real-time error tracking system for my [language/framework] app. **Include:** Recommended monitoring tools (Sentry, New Relic, Datadog). Setup steps for integration. Error categorisation for alerts. Dashboard layout suggestions. Weekly reporting format. Output in tool setup guide + dashboard screenshot mockup."



21. Prompt 21 — Data Processing Script Debugging

Data Processing Script Debugging "You are a data engineer. My Python ETL (Extract, Transform, Load) script fails at the transformation stage. **Include:** Step-by-step debugging for data type mismatches. Handling null values & schema changes. Logging setup for error tracking. Optimised data transformation logic. Unit tests for validation. Output in debugging report + corrected script."



22. Prompt 22 — Infinite Loop Prevention in Code

Infinite Loop Prevention in Code "You are a software safety expert. Analyse my [language] code for infinite loop risks and fix them. **Include:** Detection of loops without termination conditions. Corrected loop conditions. Safeguards to prevent reoccurrence. Performance benchmarks after fix. Output in before/after code format."



23. Prompt 23 — Automated Code Documentation Generator

Automated Code Documentation Generator "**You are** a documentation automation consultant. Set up an auto-documentation system for my [language/framework] project. **Include:** Recommended tools (e.g., JSDoc, Sphinx). Integration into CI/CD pipeline. Style guide for consistent doc formatting. Sample generated documentation. Output in documentation setup guide + example output."

24. Prompt 24 — Debugging Scheduler & Cron Job Failures

Debugging Scheduler & Cron Job Failures "**You are** a backend engineer. My scheduled tasks in [language/framework] fail intermittently. **Include:** Log analysis to find failure points. Corrected cron expressions. Error handling logic. Monitoring alerts setup. Retry mechanism. Output in debugging report + fixed scheduling script."

25. Prompt 25 — Codebase Technical Debt Reduction Plan

"**You are** a senior software architect. Analyse my [language/framework] codebase for technical debt and create a 3-month cleanup roadmap. **Include:** List of outdated dependencies. Code smells & fixes. Testing coverage improvement. Refactoring priorities. Risk mitigation plan. Output in technical debt report + phased action plan."

Data Analysis & Visualization

26. Prompt 1 — Full Exploratory Data Analysis (EDA) with Actionable Insights

Full Exploratory Data Analysis (EDA) with Actionable Insights "**You are** a Senior Data Analyst with expertise in Python (Pandas, NumPy, Matplotlib, Seaborn) and data storytelling. I have a dataset in CSV format containing sales data for an Indian retail chain (2018–2024) with 50,000 rows and 12 columns (date, location, category, units sold, price, discount, etc.). Your task: Import and inspect the dataset to understand structure, data types, and column meanings. Generate summary statistics for both numerical and categorical columns, highlighting anomalies. Detect missing values, quantify them column-wise, and suggest domain-specific imputation or removal strategies. Identify outliers using both statistical (Z-score, IQR) and visual (boxplots) methods, explaining potential business causes. **Create** a correlation heatmap for numerical features and explain the top 5 strongest relationships in simple business terms. **Provide** at least 3 actionable business insights and possible next analytical steps. Output format: A Jupyter Notebook with fully commented code, inline plots, and a concluding Markdown section explaining insights in non-technical language."

27. Prompt 2 — Interactive KPI Dashboard for Decision-Makers

Interactive KPI Dashboard for Decision-Makers "**You are** a Business Intelligence (BI) Dashboard Expert skilled in Power BI, Tableau, and Google Data Studio. I have quarterly sales data for multiple product categories in India for the last 5 years. Your task: Identify 5–7 key performance

indicators (KPIs) relevant to retail business health (e.g., total revenue, gross margin, average order value, conversion rate). **Design** an interactive dashboard layout showing KPIs as cards at the top, trend charts in the middle, and filters (by time, category, location) on the side. **Include** drill-down capability so a user can click on a KPI and view detailed breakdowns by category, region, or month. **Add** a geographic heatmap showing sales distribution across states, with hover tooltips. **Include** an export-to-PDF function for monthly reporting. Output format: **Provide** a step-by-step dashboard build guide (tool-agnostic), a mockup diagram of the dashboard, and sample formulas for KPI calculations."



28. Prompt 3 — Predictive Sales Forecasting with Model Comparison

Predictive Sales Forecasting with Model Comparison "**You are** a Data Scientist specialising in forecasting. I have monthly sales data for an e-commerce platform from January 2018 to December 2024. Your task: Perform time-series decomposition to analyse trend, seasonality, and residual components. Build at least two forecasting models (ARIMA/SARIMA and Facebook Prophet). Compare model performance using RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error). Plot actual vs predicted sales for both models and highlight differences. **Provide** recommendations for which model to deploy, along with a 12-month sales forecast. Output format: A Python Jupyter Notebook with all code, plots, and a Markdown cell comparing models with business-friendly explanations."



29. Prompt 4 — Automated Data Cleaning and Preprocessing Script

Automated Data Cleaning and Preprocessing Script "**You are** a Data Preprocessing Automation Specialist skilled in Python and Pandas. I have a CSV file containing a mix of numerical, categorical, and datetime fields, with missing values and inconsistent formats. Your task: Write a reusable Python script to detect and handle missing values using mean/median/mode or forward-fill/backward-fill depending on the column type. Remove duplicate rows and flag near-duplicates for manual review. Normalise numerical columns (min-max or z-score scaling) and encode categorical columns (label or one-hot encoding as appropriate). Convert date columns to proper datetime format and extract features (day, month, year, day-of-week). Save the cleaned dataset to a new CSV file with a timestamped filename. Output format: A fully commented Python script that can be reused for different datasets, with clear function definitions."



30. Prompt 5 — Business Data Storytelling for Stakeholder Reports

Business Data Storytelling for Stakeholder Reports "**You are** a Business Data Storyteller with experience in creating executive summaries from analytical results. I have analysed customer purchase behaviour for my online store and want to present findings to the leadership team. Your task: Frame the analysis as a story — starting with the problem, key findings, and implications. Select only the 5–7 most impactful visuals from the analysis, ensuring they are simple and easy to understand. **Explain** each chart in 1–2 sentences highlighting what matters for the business. Conclude with 3 actionable recommendations, each tied to a business outcome (e.g., revenue growth, cost saving). **Create** a 2-slide PowerPoint layout that can be used in a leadership meeting. Output format: A concise text storyboard + slide content that a non-technical executive can understand at a glance."

31. Prompt 6 — Real-Time Data Monitoring and Alerts

Real-Time Data Monitoring and Alerts "**You are** a Real-Time Data Monitoring Specialist skilled in tools like Grafana, Kibana, and Power BI Streaming Dataflows. I operate a logistics company with live GPS and delivery data flowing in every 15 seconds. Your task: **Design** a real-time dashboard that displays vehicle location, delivery status, and delays in near real-time. Implement colour-coded alerts for deliveries delayed beyond SLA (Service Level Agreement) thresholds. **Add** trend visualisations for daily delivery count, average delivery time, and % on-time rate. Integrate automated alerts via email and SMS for key managers when KPIs cross thresholds. Ensure the system can handle data spikes (e.g., festival season). Output format: A visual architecture diagram + tool integration plan + example SQL queries for alert generation."

32. Prompt 7 — Sentiment Analysis of Customer Feedback

Sentiment Analysis of Customer Feedback "**You are** an NLP (Natural Language Processing) Specialist with expertise in Python libraries like NLTK, SpaCy, and Transformers. I have 50,000 customer reviews collected over 2 years. Your task: Clean and preprocess the text (remove stopwords, lemmatise, handle emojis). Classify sentiment into positive, neutral, and negative categories using a pre-trained BERT model. **Create** visualisations: Sentiment distribution pie chart. Monthly sentiment trend line chart. Word cloud for each sentiment category. Identify top 5 positive and top 5 negative themes with example reviews. **Provide** actionable recommendations for product/service improvement based on sentiment patterns. Output format: A Jupyter Notebook with code, charts, and a Markdown insights summary."

33. Prompt 8 — Comparative Category Performance Report

Comparative Category Performance Report "**You are** a Business Performance Analyst. I have category-wise sales data for 10 product categories over the last 3 years. Your task: Calculate YoY (Year-over-Year) and MoM (Month-over-Month) growth rates for each category. Rank categories based on revenue, profit margin, and units sold. **Create** a dashboard view showing category trends side-by-side. Highlight top 3 performing categories and bottom 3 lagging categories. **Suggest** category-level actions to boost sales and margins for underperformers. Output format: A comparative analysis table + dashboard layout mockup + 1-page action plan."

34. Prompt 9 — Correlation and Causation Testing

Correlation and Causation Testing "**You are** a Data Scientist with a focus on statistical inference. I have a dataset on marketing spend (TV, social media, influencer, print) and corresponding sales figures. Your task: Calculate correlation coefficients for each marketing channel vs sales. Perform hypothesis testing to check statistical significance (p-values). Run a multiple regression analysis to see which channels predict sales best. Visualise results using scatter plots and regression lines. **Provide** a plain-language explanation of findings for non-technical stakeholders. Output format: Jupyter Notebook with plots + regression output table + simplified insights brief."

35. Prompt 10 — Customer Churn Prediction and Retention Strategy

Customer Churn Prediction and Retention Strategy "**You are** a Customer Analytics Expert specialising in churn modelling. I have SaaS customer data including sign-up date, usage frequency, support tickets, and payment history. Your task: Define churn for my business context (e.g., inactive for 60 days). Engineer predictive features from usage and payment history. Build a classification model (Logistic Regression, Random Forest, or XGBoost) to predict churn probability. Evaluate using accuracy, precision, recall, and ROC-AUC. **Suggest** retention strategies for the top 20% at-risk customers. Output format: Python Notebook with code + confusion matrix + strategic retention plan."

36. Prompt 11 — Data Visualization Best Practices Guide

Data Visualization Best Practices Guide "**You are** a Data Visualization Trainer. **Prepare** a best practices guide for visualising financial performance data for stakeholders. Your task: **Recommend** which chart types to use for time-series, category comparison, and part-to-whole analysis. **Suggest** an accessible, colour-blind-friendly palette. **Explain** how to avoid misleading scales and data distortion. **Include** 3 examples of excellent visualisations and explain why they work. **Provide** 3 poor visualisation examples and show corrected versions. Output format: A 5-page PDF guide with do's and don'ts + visual examples."

37. Prompt 12 — Multi-Dataset Integration Workflow

Multi-Dataset Integration Workflow "**You are** a Data Integration Specialist skilled in ETL (Extract, Transform, Load) processes. I have three datasets: Customer demographics (Excel) Purchase history (CSV) Web analytics data (Google Analytics export) Your task: Identify common keys for merging datasets. Clean and standardise column formats and naming. Join datasets into a master table. Perform initial descriptive analysis on combined data. **Suggest** 3 insights achievable only after combining data. Output format: Python Notebook with ETL code + final merged dataset snapshot + insight summary."

38. Prompt 13 — Interactive Geo-Spatial Sales Mapping

Interactive Geo-Spatial Sales Mapping "**You are** a GIS (Geographic Information Systems) Analyst. I have state-wise sales data for India for the past 12 months. Your task: **Create** an interactive map showing sales density using a colour gradient. **Add** filters for month, product category, and sales rep. Display state-level tooltips with key KPIs (revenue, units sold, growth rate). Enable comparison mode for two selected states. **Provide** export options (PNG, PDF). Output format: Dashboard implementation guide + sample data visualisation screenshot."

39. Prompt 14 — Industry Benchmark Comparison with Gap Analysis

Industry Benchmark Comparison with Gap Analysis "**You are** a Market Intelligence Analyst specialising in competitive benchmarking. I have my company's quarterly performance metrics for revenue, gross margin, and customer acquisition rate, and I have benchmark data for top 5 competitors. Your task: Normalise all data for fair comparison (e.g., currency conversion, adjusting

for fiscal year differences). **Create** comparative bar charts showing my company vs each competitor for each KPI (Key Performance Indicator). Calculate % variance from industry average for each KPI. Identify areas where my company is above average and where it's lagging. **Provide** 5 targeted recommendations to close performance gaps. Output format: A 2-page PDF competitive report with visual comparisons, an executive summary, and a prioritised action list."



40. Prompt 15 — Data Pipeline Performance Optimisation Plan

Data Pipeline Performance Optimisation Plan "You are a Data Engineer experienced in optimising ETL (Extract, Transform, Load) pipelines for speed and efficiency. I have a nightly pipeline that ingests sales, inventory, and customer data into a central warehouse. Your task: Profile the current pipeline to identify slow queries, inefficient joins, and bottleneck processes. **Recommend** improvements in query optimisation, indexing, and caching. **Suggest** parallelisation or batch processing strategies to reduce runtime. Propose monitoring tools to track pipeline health and error rates. **Provide** an example optimised SQL query and ETL script snippet. Output format: A technical optimisation plan with a "before vs after" runtime projection chart and sample code."



41. Prompt 16 — Social Media Engagement Analytics Dashboard

Social Media Engagement Analytics Dashboard "You are a Digital Analytics Expert skilled in API integrations and BI dashboarding. I have social media engagement data from Facebook, Instagram, and LinkedIn for the past 12 months. Your task: **Create** a unified dashboard showing platform-wise engagement metrics (likes, comments, shares, saves). **Add** a filter to view engagement by post type (video, carousel, single image, story). Highlight top 10 performing posts across all platforms with engagement breakdown. **Add** follower growth trend lines for each platform. **Include** an insights section suggesting which content format drives the highest engagement. Output format: Dashboard wireframe + API integration guide + engagement insights report."



42. Prompt 17 — Real Estate Market Analysis with Investment Insights

Real Estate Market Analysis with Investment Insights "You are a Real Estate Data Analyst with expertise in property market trends. I have a dataset of property sales in [city] for the last 5 years with columns for location, property type, size, sale price, and date. Your task: Analyse price trends by property type (apartment, villa, plot). Map high-growth neighbourhoods using price appreciation over time. Identify seasonality patterns in sales volume. Calculate ROI projections for top 5 emerging areas. **Provide** investment recommendations for buyers targeting high rental yield vs capital appreciation. Output format: A PDF market report with heatmaps, trend charts, and a 1-page "Investor Recommendations" summary."



43. Prompt 18 — Healthcare Operational Efficiency Analysis

Healthcare Operational Efficiency Analysis "You are a Healthcare Data Analyst working on hospital efficiency improvement. I have anonymised patient visit data, bed occupancy records, and treatment timelines for the past 2 years. Your task: Calculate average patient wait time, treatment time, and discharge time. **Create** bed occupancy rate visualisations by department. Identify peak patient inflow periods and staffing shortages. **Suggest** scheduling optimisations to reduce

48. Prompt 23 — Website Traffic and Conversion Analytics

Website Traffic and Conversion Analytics "**You are** a Web Analytics Consultant. I have Google Analytics data for my e-commerce site over the past 6 months. Your task: Identify top 5 traffic sources and their respective conversion rates. Analyse bounce rate, average session duration, and pages per session. Map the customer journey from landing page to purchase. Highlight underperforming landing pages and suggest optimisation strategies. **Provide** projected improvement metrics if changes are implemented. Output format: Data Studio dashboard layout + optimisation recommendations report."

49. Prompt 24 — Manufacturing Process Efficiency Visualisation

Manufacturing Process Efficiency Visualisation "**You are** a Manufacturing Data Engineer. I have IoT sensor data for multiple machines in a production line over the past year. Your task: Visualise machine uptime/downtime as a Gantt chart. Identify bottlenecks in production flow. Calculate defect rates per machine and per shift. **Recommend** preventive maintenance schedules. **Suggest** workflow changes to increase throughput without adding resources. Output format: Factory floor dashboard layout + process improvement plan."

50. Prompt 25 — Financial Performance Storytelling for Investors

"**You are** a Financial Data Storyteller. I have quarterly income statements, balance sheets, and cash flow statements for the past 3 years. Your task: Visualise revenue, gross profit, and net profit trends over time. Calculate key financial ratios (ROE, ROA, current ratio, debt-to-equity) and explain their meaning. Highlight major changes in expenses or revenue sources. **Provide** a year-over-year growth summary. Frame findings in a narrative that inspires investor confidence. Output format: Investor-ready slide deck with visuals, ratio analysis, and growth narrative."

API Integration & Automation

51. Prompt 1 — Connecting Multiple APIs for Unified Data

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: Authenticate each API using OAuth 2.0 and store tokens securely. Pull product sales data from Shopify, website traffic data from Google Analytics, and inventory data from Google Sheets. Merge the datasets on product ID and date fields for unified reporting. Schedule the workflow to run daily at midnight using a cron job or cloud scheduler. **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)."

52. Prompt 2 — Automating Social Media Posting via API

store real-time prices every 5 minutes. Compare the current price with the opening price for percentage change. Trigger an email/SMS alert when the $\pm 3\%$ threshold is crossed. 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."

57. Prompt 7 — Automating PDF Invoice Creation from Sales Data

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: Read the CSV to fetch customer details, products, and prices. Generate a branded PDF invoice for each customer. Save the invoice locally and in Google Drive. Email the invoice to the customer with a personalised message. 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."

58. Prompt 8 — Weather-Based Automation for Agriculture

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: Connect to the OpenWeatherMap API to fetch daily forecasts. If rainfall probability is $>70\%$, delay irrigation by 24 hours. If temperature $>35^{\circ}\text{C}$, schedule an extra watering cycle. Send an SMS to the farmer confirming the decision. 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."

59. Prompt 9 — Automating YouTube Video Uploads

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: Authenticate using YouTube Data API v3 with OAuth 2.0. Loop through a folder containing video files. Read metadata from a CSV (title, description, tags, privacyStatus). Upload each video with the corresponding metadata. 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."

60. Prompt 10 — Daily Currency Conversion Automation

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: Connect to a currency exchange API (e.g., ExchangeRate-API). Fetch

latest conversion rates for INR to target currencies. Write data to a specific Google Sheets cell range. **Include** timestamp of last update. 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."

61. Prompt 11 — Automating Job Application Tracking

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: Authenticate with LinkedIn API and Naukri.com's developer API (or scrape data if no API exists). Fetch job title, company name, date applied, status (applied, shortlisted, interview scheduled), and job link. Push this data into a Google Sheet in structured columns. Highlight rows where the application has been idle for >14 days. 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"

62. Prompt 12 — Automating Property Price Tracking for Investment

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

63. Prompt 13 — Automating Invoice Payment Reminders

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: Pull unpaid invoice data from QuickBooks API with due dates. Identify invoices past due by 7, 14, or 21 days. Send a customised reminder email based on how late the

payment is. Log all sent reminders in a Google Sheet. 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"



64. Prompt 14 — Automating Resume Screening for Recruitment

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: Monitor a Google Drive folder for new resumes. Extract text from PDFs/DOCs using an OCR/NLP API. Match candidate skills with a given job description using keyword matching and semantic similarity. Move shortlisted resumes to a "Shortlisted" folder. 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)"



65. Prompt 15 — Automating YouTube Comment Sentiment Analysis

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: Connect to YouTube Data API to fetch comments for all videos from the last 7 days. Run sentiment analysis using a pre-trained model (e.g., VADER, BERT). Categorise comments as positive, negative, or neutral. **Create** visualisations showing weekly sentiment trends. 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"



66. Prompt 16 — Automating E-commerce Inventory Updates Across Platforms

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: Connect to all three platform APIs using secure authentication (API keys or OAuth). Fetch the latest inventory count from my central warehouse database or Google Sheet. Update product stock levels on each platform. Send me an email if a product's stock falls below a reorder threshold. 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"

67. Prompt 17 — Automating Customer Support Ticket Categorisation

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: Fetch new support emails via Gmail API. Run NLP classification to detect category (Billing, Technical Issue, General Query, Complaint). **Create** a ticket in Zendesk with the detected category. Assign tickets to the relevant department queue. 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"

68. Prompt 18 — Automating Daily Stock Market Newsletter

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: Connect to Yahoo Finance API for NIFTY 50, SENSEX, and top 10 stocks data. Pull top 5 market news headlines from News API. Use GPT API to generate a 150-word market analysis. Send the report via Gmail API and post to a Telegram channel via Telegram Bot API. 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"

69. Prompt 19 — Automating Attendance Tracking with Face Recognition

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: Connect a webcam to capture employee images at check-in/check-out. Run face recognition using an API like AWS Rekognition or OpenCV. Match recognised faces to employee IDs. Update attendance in Zoho People API. 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"

70. Prompt 20 — Automating Podcast Transcription and Upload

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464

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. Your task: Fetch sales data from CRM API (HubSpot or Salesforce). Fetch expenses from accounting API (QuickBooks or Zoho Books). Fetch campaign performance from Google Ads API. Push all data to a Power BI dataset via REST API. 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"



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: Store client details (name, address, contract terms) in a Google Sheet or database. Pull data via API and inject into pre-defined Word/PDF templates. Save final documents in Google Drive and send via Gmail API. Track sent documents in a log sheet. 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"



74. Prompt 24 — Automating Social Media Comment Replies

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: Fetch new comments via Instagram API. Run keyword & sentiment analysis to classify the comment. Use a pre-defined reply template for each sentiment type. Post the reply via API. 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"

75. Prompt 25 — Automating YouTube to Instagram Clip Conversion

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: Fetch video from YouTube Data API. Trim to highlight section based on timestamps from a CSV. **Add** captions using an API like Rev.ai. Upload to Instagram via Instagram Graph API. 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"


Product Documentation & User Guides

76. Prompt 1 — Creating a Step-by-Step User Guide for a Mobile App

Creating a Step-by-Step User Guide for a Mobile App Backstory: Your startup just launched a budgeting mobile app. Many first-time users uninstall it because they can't figure out how to set up their first budget. Goal: **Create** a simple, visual, step-by-step guide that walks new users through account creation, linking bank accounts, and setting their first budget. Prompt: "**You are** a User Experience Documentation Specialist. I need you to create a beginner-friendly setup guide for our budgeting mobile app. Your task: Break the guide into 6–8 clear steps. **Include** screenshots and captions for each step. Write in simple language for non-technical users. **Add** a troubleshooting section for common signup issues. Format the final document for PDF and in-app help center. Output format: Step-by-step guide (with images) in PDF + HTML version for web embedding. Input Files & Code Section: app_screenshots.zip containing key UI images. Brand style guide PDF for fonts & colors. CSV file of common user questions from support tickets."

77. Prompt 2 — Generating API Documentation for a Developer Portal

Generating API Documentation for a Developer Portal Backstory: You've built a public API for your food delivery platform, but developers keep asking for examples and request format details because your docs are incomplete. Goal: **Create** clear, developer-focused API documentation with examples and authentication instructions. Prompt: "**You are** a Technical API Documentation Expert. I want you to create API docs for our food delivery API. Your task: Describe authentication (OAuth 2.0) process clearly. Document all endpoints with methods, parameters, and sample JSON responses. **Provide** example code snippets in Python, JavaScript, and cURL. **Include** rate limits and error codes. **Add** a "Getting Started" quick guide for first-time developers. Output format: Markdown-based API documentation + HTML developer portal version. Input Files & Code Section: OpenAPI/Swagger specification file (api_spec.json). List of example API requests and responses. Branding assets for developer portal."




78. Prompt 3 — Writing Release Notes for a SaaS Platform

Writing Release Notes for a SaaS Platform Backstory: Your SaaS analytics tool has frequent updates, but customers are unaware of new features and fixes because release notes are dull and overly technical. Goal: Write engaging, customer-friendly release notes for each product update. Prompt: "**You are** a Product Communications Writer. I want you to create release notes for our SaaS analytics platform that both inform and excite customers. Your task: Summarize new features in plain language. Highlight bug fixes and performance improvements. **Add** screenshots or GIFs for visual impact. **Include** "How to use" tips for each new feature. Publish in both email newsletter and in-app notifications format. Output format: HTML email template + Markdown release notes file. Input Files & Code Section: Product update changelog CSV. Screenshots/GIFs folder. Customer usage analytics to highlight most-requested features."



79. Prompt 4 — Creating Onboarding Guides for a CRM Tool

Creating Onboarding Guides for a CRM Tool Backstory: Your sales team is adopting a new CRM tool, but they're struggling to switch from spreadsheets to the new system. Goal: **Create** a hands-on onboarding manual for new CRM users. Prompt: "**You are** a CRM Onboarding Documentation Specialist. I want an onboarding manual that helps sales reps transition from spreadsheets to our CRM. Your task: **Explain** CRM login and account setup. Show how to import contacts from CSV. Demonstrate adding leads, deals, and activities. **Include** best practices for daily CRM usage. **Add** a printable "Quick Reference" cheatsheet. Output format: PDF onboarding manual + 1-page cheatsheet. Input Files & Code Section: CRM system screenshots. Sample CSV contact file. Sales workflow diagram."



80. Prompt 5 — Creating a Knowledge Base Article for a Common Support Issue

Creating a Knowledge Base Article for a Common Support Issue Backstory: Customers often contact support because they forget their password and can't reset it. Goal: **Create** a self-service article that reduces these repetitive support requests. Prompt: "**You are** a Knowledge Base Content Specialist. I want you to create a help article for "How to Reset Your Password" for our e-commerce platform. Your task: Write clear, step-by-step instructions. **Include** desktop and mobile screenshots. **Provide** tips for strong password creation. **Add** troubleshooting for common reset errors. Format for search engine optimization (SEO). Output format: HTML article for help center + PDF version for

offline use. Input Files & Code Section: Screenshot set for password reset flow. Branding guide for help center articles. List of top password reset issues from support logs."

81. Prompt 6 — Creating Interactive Tutorials for a Project Management Tool

Creating Interactive Tutorials for a Project Management Tool Backstory: Your project management SaaS tool has powerful features, but most customers only use the basic ones because they're unaware of advanced capabilities. Goal: Build interactive, click-through tutorials inside the app to teach advanced features. Prompt: "**You are** a Product Education Content Developer. I need you to create in-app interactive tutorials for our project management platform. Your task: Select top 5 underused advanced features. **Create** step-by-step walkthrough scripts for each. Use tooltips, highlights, and click prompts to guide the user. **Include** progress tracking so users can resume later. **Prepare** text, screenshots, and instructional videos for each step. Output format: JSON tutorial script for app integration + video files + image assets. Input Files & Code Section: Feature usage analytics CSV. UI screenshot set. In-app tutorial framework documentation."

82. Prompt 7 — Writing Compliance & Policy Documentation

Writing Compliance & Policy Documentation Backstory: You've launched a fintech app, but regulatory requirements demand that you publish clear compliance policies for users. Goal: Draft customer-facing compliance documentation that is accurate but easy to understand. Prompt: "**You are** a Regulatory Documentation Specialist. I want you to create compliance and privacy policies for our fintech app. Your task: Review applicable regulations (RBI, GDPR, PCI DSS). Write privacy policy, data handling policy, and terms of service. Ensure language is plain and free of legal jargon where possible. **Include** diagrams showing data flow and storage locations. Format for both web and PDF publication. Output format: Policy documents in DOCX, PDF, and HTML formats. Input Files & Code Section: Current draft compliance notes. Legal team's checklist. Data flow diagrams."


83. Prompt 8 — Documenting API Integration for Third-Party Partners

Documenting API Integration for Third-Party Partners Backstory: You offer a payment API that third-party merchants can integrate, but integration requests are delayed due to unclear documentation. Goal: **Create** a developer-friendly integration manual. Prompt: "**You are** a Partner Integration Documentation Engineer. I want you to create a full integration manual for our payment API. Your task: Describe authentication, endpoint usage, and required parameters. **Provide** step-by-step integration example for a sample merchant app. **Include** common error codes and troubleshooting steps. **Add** code examples in PHP, Node.js, and Python. **Include** test environment setup instructions. Output format: Markdown integration manual + HTML version for partner portal. Input Files & Code Section: API reference (swagger.yaml). Sandbox API credentials. Sample merchant application codebase."

84. Prompt 9 — Creating Internal Developer Documentation


Creating Internal Developer Documentation Backstory: Your dev team has grown quickly, but onboarding new engineers takes weeks because there's no central engineering guide. Goal: Document internal codebase, architecture, and workflows. Prompt: "**You are** an Internal

Engineering Documentation Specialist. I want a central developer handbook for our engineering team. Your task: Describe project architecture with diagrams. Document coding standards and naming conventions. **Include** instructions for local environment setup. **Explain** CI/CD pipeline processes. Maintain this as a living document in the repo. Output format: Developer handbook in Markdown + PDF export. Input Files & Code Section: Architecture diagrams. Existing dev onboarding notes. Git repository README.md."




85. Prompt 10 — Writing Feature Comparison Guides

Writing Feature Comparison Guides Backstory: Many customers ask how your product compares to competitors, but your sales team doesn't have a clear document to share. Goal: **Create** a side-by-side feature comparison guide. Prompt: "**You are** a Competitive Product Documentation Writer. I want a feature comparison document between our tool and top 3 competitors. Your task: List key features side-by-side in a table. Use simple, customer-friendly language. Highlight where our product is stronger. **Include** screenshots for visual comparison. Format for both sales decks and website FAQ. Output format: Comparison PDF + PPT slide deck. Input Files & Code Section: Competitor feature research spreadsheet. Product screenshots folder. Brand guidelines for colors and fonts."




86. Prompt 11 — Creating Troubleshooting Flowcharts

Creating Troubleshooting Flowcharts Backstory: Your tech support team spends hours on calls walking users through fixes for common issues. Goal: **Create** self-help troubleshooting flowcharts. Prompt: "**You are** a Technical Troubleshooting Documentation Designer. I want to create visual flowcharts for common problems with our SaaS tool. Your task: Select top 5 recurring issues from support logs. **Create** clear yes/no flowcharts for each problem. Use icons and colors to make steps easy to follow. **Add** estimated time for each fix step. Export to PDF for customers and PNG for website. Output format: Flowchart diagrams in PNG + consolidated PDF booklet. Input Files & Code Section: Support ticket analysis CSV. Company color palette file. Icon set for diagrams."



87. Prompt 12 — Creating Voice & Tone Guidelines

Creating Voice & Tone Guidelines Backstory: Multiple writers contribute to your help center, but the style is inconsistent. Goal: **Create** a unified voice & tone guide for all documentation writers. Prompt: "**You are** a Content Style Guide Specialist. I want a voice & tone guideline document for our documentation team. Your task: Define brand personality in writing. **Provide** examples of do's and don'ts. **Include** guidelines for writing for technical vs. non-technical audiences. Cover accessibility considerations (readability, alt text, etc.). **Provide** templates for different content types. Output format: Voice & tone PDF + quick reference card. Input Files & Code Section: Existing help articles. Marketing brand guide. Feedback from customer surveys."



88. Prompt 13 — Writing Contextual Tooltips for a Web Application

Writing Contextual Tooltips for a Web Application Backstory: Your SaaS dashboard is feature-rich, but many first-time users don't understand what certain buttons or fields do. Goal: **Create** short, contextual tooltips that explain features without overwhelming the user. Prompt: "**You are** a UX

Microcopy Documentation Specialist. I want you to create clear, concise tooltips for our SaaS dashboard. Your task: Identify 30 key UI elements needing tooltips. Write short descriptions (max 20 words) in plain language. Ensure consistency in style and tone. **Include** an internal reference table mapping tooltip text to UI elements. **Provide** JSON/CSV format for direct integration with the UI codebase. Output format: Tooltip text table (CSV + JSON) + implementation guide. Input Files & Code Section: UI element list CSV. Screenshot set of dashboard UI. Branding style guide."

89. Prompt 14 — Creating a Quick Start Guide for a Developer SDK (Software Development Kit)

Backstory: Developers integrating your SDK are struggling because there's no concise "first steps" documentation. Goal: Build a quick start guide that allows developers to implement the SDK within 30 minutes. Prompt: "**You are** a Developer Onboarding Documentation Expert. I want a quick start guide for our JavaScript SDK. Your task: **Include** installation steps via npm/yarn. Show basic initialization code with example API calls. Document common config options and defaults. **Provide** a working sample app repository link. **Add** troubleshooting tips for common setup errors. Output format: Markdown quick start guide + PDF export. Input Files & Code Section: SDK code sample repository link. API key for sandbox testing. Screenshot folder for sample outputs."

90. Prompt 15 — Documenting Accessibility Features for Users with Disabilities

Documenting Accessibility Features for Users with Disabilities Backstory: Your platform is accessible, but many users with disabilities don't know about the available features. Goal: **Create** an accessibility guide showcasing these features. Prompt: "**You are** an Accessibility Documentation Specialist. I want an accessibility features guide for our learning platform. Your task: List all accessibility options (keyboard shortcuts, screen reader support, high contrast mode). **Provide** step-by-step activation instructions for each. **Add** compatibility notes for different browsers/devices. **Include** best practices for accessible usage. Format as both web and audio versions for accessibility. Output format: PDF + HTML + MP3 narration. Input Files & Code Section: Feature list CSV. Accessibility testing report. Screenshots and icon assets."

91. Prompt 16 — Creating Interactive FAQs with Search Functionality

Creating Interactive FAQs with Search Functionality Backstory: Your current FAQ page is static and users struggle to find relevant answers quickly. Goal: Build a searchable, interactive FAQ system. Prompt: "**You are** a Help Center Experience Designer. I want to turn our static FAQ page into an interactive, searchable database. Your task: Convert FAQs into a searchable JSON format. Tag each FAQ with categories and keywords. Implement autocomplete for search queries. **Include** expand/collapse answers for better UX. **Provide** embed code for website integration. Output format: FAQ database (JSON) + HTML/CSS/JS embed code. Input Files & Code Section: Existing FAQ text in CSV. Website brand style guide. JavaScript library documentation for search."

92. Prompt 17 — Writing Maintenance Manuals for Hardware Products

Writing Maintenance Manuals for Hardware Products Backstory: Customers often damage devices because they don't follow maintenance guidelines. Goal: Write a detailed maintenance and care manual. Prompt: "**You are** a Hardware Technical Writer. I want a maintenance manual for our smart home thermostat. Your task: List cleaning, calibration, and firmware update procedures. **Provide** do's and don'ts with illustrations. **Include** seasonal maintenance reminders. **Add** troubleshooting section for physical faults. Format for both printed booklet and online PDF. Output format: Illustrated manual in PDF + printable A5 booklet. Input Files & Code Section: Product engineering diagrams. Service checklist from repair team. Image asset folder."

93. Prompt 18 — Documenting Multi-Language Product Instructions

Documenting Multi-Language Product Instructions Backstory: You sell in multiple countries, but product instructions are only in English. Goal: **Create** multilingual product manuals. Prompt: "**You are** a Multilingual Documentation Specialist. I want product instructions for our kitchen appliance in English, Hindi, and Tamil. Your task: Translate existing manual while keeping technical accuracy. Adapt units (metric/imperial) as needed. **Include** culturally relevant examples. Ensure formatting works for all languages. **Provide** print-ready and web-ready versions. Output format: PDF manuals in all 3 languages. Input Files & Code Section: Current English manual (DOCX). Brand typography guidelines. Translation glossary file."

94. Prompt 19 — Creating How-To Videos for Common Tasks

Creating How-To Videos for Common Tasks Backstory: Written guides are available, but some customers prefer video walkthroughs. Goal: Produce short tutorial videos for key product functions. Prompt: "**You are** a Video Documentation Producer. I want 5 short (under 2 mins each) tutorial videos for our e-learning platform. Your task: Write video scripts for each task. Record screen captures with voiceover. **Add** captions and callout graphics. Export in MP4 for YouTube and MOV for in-app playback. **Provide** thumbnail images for each video. Output format: Video files + scripts + thumbnails. Input Files & Code Section: Task list CSV. Brand video intro/outro files. Voiceover style guide."


95. Prompt 20 — Creating an API Changelog Page

Creating an API Changelog Page Backstory: API updates break client integrations because developers aren't notified in time. Goal: Publish a public API changelog with versioning details. Prompt: "**You are** an API Documentation Manager. I want a live API changelog page for our developer portal. Your task: Track API version changes with release dates. **Add** summaries of new/removed/modified endpoints. Highlight breaking changes in red. **Provide** migration notes for affected endpoints. Update automatically via CI/CD when code changes are merged. Output format: Markdown changelog + HTML portal page. Input Files & Code Section: Git commit history. API spec change diff file. Developer portal access."

96. Prompt 21 — Writing Internal Product Playbooks for Support Staff


Writing Internal Product Playbooks for Support Staff Backstory: Support staff often escalate tickets unnecessarily because they lack clear product troubleshooting guidelines. Goal: **Create** internal

product playbooks for the support team. Prompt: "**You are** a Support Operations Documentation Specialist. I want to create internal playbooks for handling common customer issues. Your task: Document step-by-step troubleshooting workflows. **Include** escalation criteria for each case. **Provide** scripts for customer communication. **Add** visual aids where relevant. Store in an internal wiki for easy updates. Output format: Playbook PDFs + wiki pages. Input Files & Code Section: Support ticket history CSV. Current internal notes. Diagram/image assets."




97. Prompt 22 — Documenting Integration with Popular Third-Party Tools

Documenting Integration with Popular Third-Party Tools Backstory: Customers want to connect your product with tools like Slack, Google Sheets, and Zapier, but don't know how. Goal: **Create** integration guides for top requested tools. Prompt: "**You are** a Third-Party Integration Documentation Specialist. I want guides for integrating our platform with Slack, Google Sheets, and Zapier. Your task: Write step-by-step instructions with screenshots. Show real-life use case examples for each integration. **Add** troubleshooting tips for API errors. **Include** estimated setup time for each. Format for help center and PDF export. Output format: 3 integration guides in PDF + HTML. Input Files & Code Section: Integration API credentials. User request survey results. Screenshot set for each tool."




98. Prompt 23 — Creating Printable Cheat Sheets for Power Users

Creating Printable Cheat Sheets for Power Users Backstory: Advanced users want quick reference material without reading long manuals. Goal: **Create** compact, printable cheat sheets. Prompt: "**You are** a Productivity Documentation Designer. I want a one-page quick reference cheat sheet for our desktop productivity app. Your task: **Include** top keyboard shortcuts. **Add** quick access menu navigation. **Include** power user tips. Use icons and color coding for readability. **Provide** in A4 and Letter size PDFs. Output format: Cheat sheet PDF in two sizes. Input Files & Code Section: Shortcut list CSV. App UI screenshots. Icon asset folder."



99. Prompt 24 — Creating User Story-Based Tutorials

Creating User Story-Based Tutorials Backstory: Customers understand better when tutorials follow real-world scenarios. Goal: Write tutorials framed as user stories. Prompt: "**You are** a Scenario-Based Learning Documentation Specialist. I want tutorials that walk through tasks using real customer scenarios. Your task: Select top 3 customer use cases. Write tutorials in story format. **Include** relevant screenshots and tips. End each with key takeaways. Format for blog and PDF. Output format: 3 story-based tutorials in HTML + PDF. Input Files & Code Section: Customer interview transcripts. Screenshot set. Branding style guide."



100. Prompt 25 — Creating AI-Assisted Product Guides

Creating AI-Assisted Product Guides Backstory: You want to experiment with AI-generated personalized product guides for new users. Goal: **Create** a template that AI can use to generate tailored guides. Prompt: "**You are** an AI-Enhanced Documentation Designer. I want a product guide template that AI can fill with user-specific tips. Your task: **Create** placeholders for user goals and usage history. **Include** a modular structure for different product features. **Provide** instructions for AI

Automating Security Log Monitoring Backstory: You're an IT administrator for a mid-sized company. Your security logs are massive, and manually scanning them for threats is impossible. Last year, you missed a brute-force attack because it got buried in the logs. Goal: **Create** an automated pipeline that monitors security logs, flags suspicious activity, and sends real-time alerts. Prompt: **"You are** a Cybersecurity Automation Engineer. I want a script that scans server logs in real time and alerts me of suspicious activity such as failed login attempts, unusual IP addresses, or data spikes. Your task: Connect to the server log files via API or secure SSH. Parse log entries and match against predefined threat patterns (failed logins >5 in 1 minute, foreign IP access, large file downloads). Send an alert email/SMS if a threat is detected. Store flagged events in a database for future analysis. Generate a daily security summary report. Output format: Security monitoring script + threat pattern list + alert notification system. Input Files & Code Section: Path to log files or log API endpoint Threat detection rules CSV Email/SMS API credentials"

Automating Data Backup with Encryption Backstory: You manage sensitive medical records for a clinic. If your system crashes or is hacked, you can't risk losing unencrypted patient data. Goal: Automate daily backups to cloud storage with strong encryption. Prompt: "**You are** a Data Security Engineer. I want an automated backup system that encrypts files before uploading them to cloud storage (AWS S3, Google Drive). Your task: Identify sensitive folders for backup. Compress and encrypt files using AES-256 encryption. Upload encrypted backups to cloud storage via API. Store encryption keys securely in a password vault. Maintain a backup log with timestamps and checksum hashes. Output format: Encrypted backup script + cloud upload integration + key storage instructions. Input Files & Code Section: Encryption key file (secure vault reference) Cloud API credentials List of file/folder paths for backup"

Automating Phishing Email Detection Backstory: Your employees keep falling for phishing emails, leading to security risks. Manual awareness training isn't enough. Goal: Build an API-based system that scans incoming emails and flags potential phishing attempts. Prompt: **"You are** an Email Security Automation Specialist. I want to integrate Gmail API with an AI phishing detection model. Your task: Fetch incoming emails via Gmail API. Scan sender domains, suspicious keywords, and link redirections. Assign a risk score to each email. Move high-risk emails to a "Quarantine" folder. Send a weekly phishing report to the security team. Output format: Email scanning script + risk scoring system + quarantine folder setup. Input Files & Code Section: Gmail API credentials Keyword/risk pattern JSON file AI phishing detection model file"

104. Prompt 4 — Automating GDPR Data Deletion Requests

Automating GDPR Data Deletion Requests Backstory: Your European customers often request data deletion under GDPR laws. Handling requests manually is time-consuming and error-prone. Goal: Automate GDPR “Right to be Forgotten” requests. Prompt: **"You are** a Privacy Compliance Automation Expert. I want a workflow that processes GDPR deletion requests automatically. Your task: Receive requests via a secure web form. Authenticate the requester’s identity via email verification. Locate all user data across databases and APIs. Delete or anonymise the data as per GDPR guidelines. Send a confirmation email and store a compliance log. Output format: Deletion automation script + compliance report template + GDPR checklist. Input Files & Code Section: Database connection details Web form API endpoint Email verification script"

105. Prompt 5 — Automating Vulnerability Scans

Automating Vulnerability Scans Backstory: You’re a security analyst at a SaaS startup. You run vulnerability scans manually once a month, but threats change daily. Goal: Schedule automated vulnerability scans for all production servers. Prompt: **"You are** a Security Scan Automation Engineer. I want to run daily vulnerability scans and generate reports automatically. Your task: Integrate with a vulnerability scanning tool API (e.g., Nessus, OpenVAS). Schedule scans for all server IPs. Export scan results to a PDF and store in a secure folder. Send a daily email summary with high-risk vulnerabilities. Track vulnerability history in a Google Sheet. Output format: Scheduled scan automation + PDF reporting script + risk tracking spreadsheet. Input Files & Code Section: Vulnerability scanner API credentials Server IP list CSV Google Sheets API credentials"


106. Prompt 6 — Automating Two-Factor Authentication (2FA) Setup for All Users

Automating Two-Factor Authentication (2FA) Setup for All Users Backstory: You’re a system admin at a mid-sized company. Many employees still use only passwords to log in, making the company vulnerable to credential theft. Goal: Enforce and automate 2FA setup across all employee accounts using APIs. Prompt: **"You are** an Identity & Access Management Automation Specialist. I want to roll out mandatory Two-Factor Authentication for all company accounts in Google Workspace. Your task: Use Google Admin SDK API to identify accounts without 2FA enabled. Send automated emails prompting users to enable 2FA. **Provide** a one-click link to the 2FA setup page. Disable accounts not compliant after 7 days. Generate a compliance report for management. Output format: 2FA enforcement script + email template + compliance report spreadsheet. Input Files & Code Section: Google Admin SDK API credentials Email template file Compliance tracking CSV"

107. Prompt 7 — Automating Data Breach Monitoring with Dark Web Scans


Automating Data Breach Monitoring with Dark Web Scans Backstory: A client’s credentials were leaked on the dark web, and you only found out weeks later. You want to monitor this proactively. Goal: **Create** an automation that scans the dark web for stolen company credentials. Prompt: **"You are** a Threat Intelligence Automation Engineer. I want to integrate HavelBeenPwned API and a dark web monitoring API to scan for leaked email/password combinations. Your task: Fetch

employee email list from HR database. Query APIs for data breaches linked to these emails. Flag and notify affected employees to reset passwords. The store results in an encrypted database. Send a monthly summary to the security team. Output format: Breach monitoring script + notification template + encrypted breach database. Input Files & Code Section: API keys for HavelBeenPwned and dark web monitoring tool Employee email list CSV Email SMTP settings for alerts"




108. Prompt 8 — Automating Role-Based Access Control (RBAC) Updates

Automating Role-Based Access Control (RBAC) Updates Backstory: Employees change departments often, but their access permissions stay the same, leaving old data vulnerable. Goal: Automate RBAC updates based on HR records. Prompt: "**You are** an Access Control Automation Expert. I want an integration between our HR system and internal application APIs to update user permissions automatically. Your task: Fetch updated employee roles from HR system API. Compare current permissions in application API. **Add/remove** access rights based on role changes. Log all changes with timestamps. Notify IT admin for high-privilege changes. Output format: RBAC sync script + permissions change log + alert email template. Input Files & Code Section: HR system API credentials Application API credentials Role-to-permission mapping JSON"



109. Prompt 9 — Automating Security Awareness Quizzes


Automating Security Awareness Quizzes Backstory: Employees forget cybersecurity best practices unless reminded regularly. Goal: Send automated monthly security quizzes to employees via email. Prompt: "**You are** a Security Training Automation Specialist. I want a system that emails a short quiz to employees each month and records their scores. Your task: Store quiz questions in a Google Sheet or database. Send quiz links via Gmail API. Collect responses via Google Forms API. Calculate scores and store in a results sheet. Flag employees who score below 70% for follow-up training. Output format: Quiz automation script + question bank file + results dashboard. Input Files & Code Section: Google Sheets API credentials Gmail API credentials Quiz question CSV or database file"



110. Prompt 10 — Automating SSL Certificate Expiry Alerts


Automating SSL Certificate Expiry Alerts Backstory: One of your client websites went down because the SSL certificate expired — and nobody noticed in time. Goal: Monitor SSL expiry dates and send alerts before expiration. Prompt: "**You are** a Web Security Automation Engineer. I want a system that checks SSL certificate expiry dates for a list of domains. Your task: Fetch SSL certificate details for each domain. Identify expiry dates within the next 30 days. Send alert emails with renewal instructions. Update a tracking sheet with expiry status. Repeat the check daily. Output format: SSL monitoring script + expiry alert template + tracking spreadsheet. Input Files & Code Section: Domain list CSV Email SMTP settings Google Sheets API credentials" Prompt 11 — Automating Endpoint Device Compliance Checks Backstory: Your company has a Bring Your Own Device (BYOD) policy, but many employees connect with outdated or unpatched devices, creating security gaps. Goal: Automatically check if employee devices meet compliance requirements before allowing network access. Prompt: "**You are** an Endpoint Security Automation Engineer. I want a system that verifies employee device compliance (OS version, antivirus status, firewall enabled) every time they connect to the company network. Your task: Integrate with an endpoint

management API (e.g., Microsoft Intune, Jamf). Collect device compliance data in real time. Block network access if the device fails checks. Notify the employee with steps to fix compliance issues. Log all non-compliant devices for security audits. Output format: Compliance check script + access control API integration + remediation email template. Input Files & Code Section: Endpoint management API credentials Compliance rule configuration file (JSON) Network access control API credentials"




111. Prompt 11 — Automating Endpoint Device Compliance Checks

Automating Endpoint Device Compliance Checks Backstory: Your company has a Bring Your Own Device (BYOD) policy, but many employees connect with outdated or unpatched devices, creating security gaps. Goal: Automatically check if employee devices meet compliance requirements before allowing network access. Prompt: "**You are** an Endpoint Security Automation Engineer. I want a system that verifies employee device compliance (OS version, antivirus status, firewall enabled) every time they connect to the company network. Your task: Integrate with an endpoint management API (e.g., Microsoft Intune, Jamf). Collect device compliance data in real time. Block network access if the device fails checks. Notify the employee with steps to fix compliance issues. Log all non-compliant devices for security audits. Output format: Compliance check script + access control API integration + remediation email template. Input Files & Code Section: Endpoint management API credentials Compliance rule configuration file (JSON) Network access control API credentials"



112. Prompt 12 — Automating Sensitive File Access Alerts

Automating Sensitive File Access Alerts Backstory: You store financial reports in a shared drive, and last quarter a contractor downloaded files they weren't supposed to access. Goal: Set up real-time alerts for access to sensitive files. Prompt: "**You are** a File Access Monitoring Specialist. I want an automation that detects and alerts whenever certain high-security files are accessed. Your task: Connect to Google Drive API or internal file server API. Monitor access logs for the target file/folder. Trigger an alert when access is detected outside approved user list. Record details: user ID, timestamp, IP address. Send an incident report to the security team. Output format: File access monitoring script + alerting system + incident report format. Input Files & Code Section: File/folder ID list CSV Approved user list CSV Email/SMS API credentials"




113. Prompt 13 — Automating Database Security Audits

Automating Database Security Audits Backstory: Your customer database holds personal information, but monthly manual security audits take too much time and miss critical misconfigurations. Goal: Automate periodic database security audits and reporting. Prompt: "**You are** a Database Security Automation Expert. I want a script that scans for vulnerabilities in our MySQL/PostgreSQL databases and generates a security report. Your task: Connect to the database securely. Check for weak passwords, outdated versions, excessive user privileges. Identify unused accounts and revoke access. Generate a PDF report with recommendations. Email the report to the database administrator. Output format: Database audit script + PDF report template + email delivery function. Input Files & Code Section: Database connection credentials (secured) Vulnerability scan checklist JSON Email SMTP settings"




114. Prompt 14 — Automating USB Device Restrictions

Automating USB Device Restrictions Backstory: An employee once copied sensitive data onto a personal USB drive without permission. Goal: Automatically detect and block unapproved USB devices. Prompt: "**You are** an Endpoint Device Control Automation Specialist. I want a system that blocks all USB devices except approved company drives. Your task: Detect when a USB device is connected. Compare its serial number against the approved list. Block access if not approved. Send an alert to the IT security team. Log all USB connection attempts. Output format: USB restriction script + approved device list + alert and log system. Input Files & Code Section: Approved USB device serial number list CSV Endpoint monitoring API credentials"




115. Prompt 15 — Automating Password Expiry Reminders

Automating Password Expiry Reminders Backstory: Employees often forget to change passwords on time, leading to expired accounts and downtime. Goal: Send automated password change reminders. Prompt: "**You are** an Account Security Automation Specialist. I want to integrate with Active Directory (AD) to send reminders before password expiry. Your task: Connect to AD via API or LDAP. Fetch users whose passwords expire within 10 days. Send email reminders at 10, 5, and 2 days before expiry. Track who changes passwords after reminders. Generate a monthly compliance report. Output format: Reminder script + email template + compliance tracking sheet. Input Files & Code Section: AD connection details Email SMTP credentials"




116. Prompt 16 — Automating Encrypted File Sharing

Automating Encrypted File Sharing Backstory: Your legal team frequently shares confidential documents with clients, but sending via regular email is risky. Goal: **Create** a secure encrypted file-sharing automation. Prompt: "**You are** a Secure File Transfer Automation Expert. I want to encrypt files and send them via a secure download link that expires after 48 hours. Your task: Accept file upload from the legal team. Encrypt the file using AES-256. Upload to secure cloud storage. Generate a time-limited download link. Email the link to the client with a decryption key sent separately. Output format: Secure sharing script + encryption guide + link expiration setup. Input Files & Code Section: Encryption key management file Cloud storage API credentials"




117. Prompt 17 — Automating Insider Threat Detection

Automating Insider Threat Detection Backstory: A recently resigned employee downloaded large volumes of data before leaving. Goal: Detect unusual internal data access patterns. Prompt: "**You are** an Insider Threat Monitoring Specialist. I want a system that flags abnormal access activity by employees. Your task: Collect access logs from file servers, databases, and cloud storage. Identify sudden spikes in file downloads or sensitive data access. Compare against the user's historical activity patterns. Flag anomalies and send alerts to security admins. Log all incidents for investigation. Output format: Threat detection script + anomaly detection rules + alert system. Input Files & Code Section: Access log API endpoints User activity baseline data CSV Email/SMS API credentials"




118. Prompt 18 — Automating Compliance Document Management

Automating Compliance Document Management Backstory: You must submit ISO 27001 compliance reports annually, but collecting required documents is messy. Goal: Automate collection and organisation of compliance evidence. Prompt: "**You are** a Compliance Automation Specialist. I want a system that fetches logs, audit reports, and policy documents from multiple systems into one folder. Your task: Connect to APIs for security tools, HR systems, and monitoring platforms. Download latest compliance-related files. Store them in a structured folder by category and date. Generate a manifest listing all collected documents. Zip and archive the folder. Output format: Document collection script + manifest file + folder structure template. Input Files & Code Section: API credentials for each system Compliance checklist JSON"




119. Prompt 19 — Automating Ransomware Simulation Drills

Automating Ransomware Simulation Drills Backstory: You want your IT team prepared for ransomware attacks, but training is irregular. Goal: Automate simulated ransomware drills. Prompt: "**You are** a Cybersecurity Training Automation Engineer. I want a script that simulates a ransomware infection without actually encrypting files. Your task: Rename and lock sample files in a sandbox environment. Display a mock ransom note. Test the IT team's incident response process. Record time taken to respond. Generate a performance report. Output format: Simulation script + ransom note template + performance report. Input Files & Code Section: Sandbox environment setup file Sample file set"




120. Prompt 20 — Automating API Security Testing

Automating API Security Testing Backstory: Your company's APIs are public-facing, and you want to test them regularly for vulnerabilities. Goal: **Create** an automated API penetration testing tool. Prompt: "**You are** an API Security Testing Specialist. I want a script that runs OWASP API Security Top 10 checks on all company APIs. Your task: Load list of API endpoints from a file. Run tests for authentication flaws, excessive data exposure, and injection attacks. Record findings in a structured report. Notify developers of high-risk vulnerabilities. Store results for trend analysis. Output format: API security scan script + vulnerability report + notification system. Input Files & Code Section: API endpoint list CSV API security rules JSON"




121. Prompt 21 — Automating Cloud Security Policy Enforcement

Automating Cloud Security Policy Enforcement Backstory: Your cloud storage contains sensitive client contracts, but some employees make files public by mistake. Goal: Automatically detect and fix misconfigured cloud permissions. Prompt: "**You are** a Cloud Security Automation Specialist. I want a system that scans all files in Google Drive/AWS S3 for public access and restricts them to approved users only. Your task: Connect to the cloud storage API. Identify files/folders with public sharing enabled. Automatically remove public access. Notify the file owner about the change. Log all permission changes for audits. Output format: Permission scan script + remediation log + owner notification email template. Input Files & Code Section: Cloud storage API credentials Approved user list CSV Notification email template"




122. Prompt 22 — Automating Incident Response Playbook Execution

Automating Incident Response Playbook Execution Backstory: In case of a security breach, your IT team follows a manual checklist, which delays containment. Goal: Automate the first 10 minutes of incident response. Prompt: "**You are** a Security Incident Automation Engineer. I want a system that executes predefined playbook actions when a breach is detected. Your task: Receive breach alert from SIEM (Security Information and Event Management) tool. Isolate affected servers from the network. Collect logs and system snapshots. Notify the security team and management. Update the incident tracking system. Output format: Incident response script + action log + notification system. Input Files & Code Section: SIEM API credentials Server isolation commands script Incident tracking tool API key"




123. Prompt 23 — Automating Encrypted Chat for Sensitive Communications

Automating Encrypted Chat for Sensitive Communications Backstory: Your legal team sometimes needs to chat securely with external lawyers, but regular Slack channels are not safe enough. Goal: **Provide** an automated, temporary encrypted chat channel. Prompt: "**You are** a Secure Communications Automation Specialist. I want a tool that creates an encrypted chat room that expires after 24 hours. Your task: Generate a secure chat room via API (e.g., Matrix, Rocket.Chat). Require password-protected entry. Enable end-to-end encryption for all messages. Automatically delete the chat room and logs after expiry. Email participants the join link and password. Output format: Chat room creation script + deletion automation + participant email template. Input Files & Code Section: Secure chat API credentials Participant email list CSV"



124. Prompt 24 — Automating Malware File Scanning for Uploads

Automating Malware File Scanning for Uploads Backstory: Your website allows file uploads for client documents, but there's a risk of malicious files being uploaded. Goal: Automatically scan all uploaded files for malware. Prompt: "**You are** a File Security Automation Engineer. I want an integration that scans each uploaded file with a malware detection API before it's stored. Your task: Intercept file uploads via the web application backend. Scan files using VirusTotal or ClamAV API. Reject and quarantine suspicious files. Notify the uploader about rejection. Log scan results for audits. Output format: File scanning script + quarantine folder setup + log file format. Input Files & Code Section: Malware scanning API credentials File storage path configuration Notification email template"



125. Prompt 25 — Automating Security Patch Deployment

Automating Security Patch Deployment Backstory: A zero-day vulnerability was discovered last month, but patching all systems took your team over a week. Goal: Deploy security patches automatically across all servers. Prompt: "**You are** a Patch Management Automation Specialist. I want a script that applies critical security patches across all systems as soon as they're available. Your task: Check vendor API or repositories for new patches. Download and install patches automatically. Reboot systems if required, during off-hours. Notify admins of patch completion. Maintain a patch history log. Output format: Patch automation script + reboot schedule + patch log. Input Files & Code Section: Server list CSV Patch repository URLs Admin email list"

