AI Case Studies: Practical Workflows I Built with ChatGPT

by Irene Baiden

I use AI daily to optimise my workflows and turn ideas into systems. These case studies show how I designed and refined prompts to save time, improve consistency, and learn faster.

Using AI to Cut Planning Time from Days to Minutes

The Problem:

Planning large projects used to take days. Even after finishing a plan, I often had to rework it, which was draining and time-consuming

The Process:

▼ Anatomy of a Prompt

Set a persona: Act as an event planner.

Instructions: Create a 10-day prep plan for my presentation.

Context & constraints: 1–2 hours per day, focus-heavy mornings, lighter evenings, rest days included.

▼ Prompt Patterns Used

Break it down into parts : Structured it across 10 days with rest days.

Format your answer as... : Delivered a checklist for easy scanning.

Thinking step by step: Ensured the plan was built progressively (outline to slides to practice to polish).

▼ Systematic Debugging

Failure mode: First drafts were too vague and overloaded with tasks.

Weakness: No session caps, unclear outcomes, and unclear format.

Refinement: Added workload caps, outcome focus (confidence + clarity), checklist format, and conversational tone.

▼ Progress Recap

Original Prompt: "Act as an event planner and create a 10-day prep plan..."

Output: Decent structure but too open-ended.

Final Prompt: "Act as an event planner and create a 10-day prep plan... include session caps, rest days, step-by-step build, confidence focus, and format as a checklist."

Output: Polished, structured, supportive plan.

The Results:

💡 Reduced planning time from 2 hrs 43 mins manually to 18 minutes with AI (≈89% faster).

💡 Produced a clear 10-day prep checklist:

💡 Freed up afternoons for creative, high value work.

Example output snippet:

- Day 1- Foundation
- Day 3 - Slide Draft
- Day 6- Rest/Reset

The Takeaway:

> 💡 AI can take away the draining parts of planning while keeping the human touch.
>
> By combining structured prompting with systematic refinement, I turned a vague idea into a step-by-step prep plan that builds confidence and clarity.
>
> This workflow is reusable across personal, professional, and team projects.

Designing 30-Day Goal Challenges with AI to Boost Consistency

The Problem

Momentum often dropped on both short- and long-term goals. Without a clear strategy, I spent more time overthinking than making progress.

The Process :

▼ Anatomy of a Prompt

Set a persona: Act as a supportive challenge designer.

Give instructions: Design a 30-day Python challenge with a mix of quick wins and deep-focus tasks

Add context & constraints: 45- 60 minutes daily, include rest/review days, keep the pacing gentle.

▼ Prompt Patterns Used

Break it down into parts : Structured the challenge week by week

Format your answer as... : Produced a checklist with Day, Task, Goal, Motivation.

Thinking step by step: Kept the progression logical and achievable.

▼ Systematic Debugging

Failure mode: First draft was too broad and too heavy.

Weakness: No pacing, no motivation, unclear outcomes.

Refinement: Added time caps, weekly milestones, rest days, and motivational tips.

▼ Progress Recap

Original Prompt: "Act as a supportive challenge designer and design a 30 day course on python. There needs to be some rest days."

> **Output:** Clear request but role is vague, rest days are not defined and format is unspecified.

Final Prompt: "Act as a supportive challenge designer and create a 30-day Python course for beginners... include rest days, motivation boosts, outcome focus, and format as a checklist."

> **Output:** A polished, motivating 30-day challenge.

The Results:

💡 Reduced planning time from 3 hrs 16 mins manually to 25 minutes with AI (≈87% faster)

💡 Completed a full 30-day challenge for the first time without burnout.

Example snippet (Week 3):

- Day 15 — Functions drill (25m)
- Day 16 — Refactor yesterdays's code (20m)
- Day 17— Mini project: CLI calculator (30m)

The Takeaway :

💡 AI helped transform vague goals into paced, motivating steps.

By combining role-based prompts, structured patterns, and systematic refinement, I created a repeatable system for personal growth, team projects, and professional upskilling.

Turning Complex Subjects into One-Page Learning Guides with AI

The Problem:

I often struggled to gather accurate information and structure it in a way that fit my learning style. Learning new topics beyond the basics could take hours of scattered research.

The Process:

▼ Anatomy of a Prompt

Set a persona : “Act as a web developer planner.”

Instructions: “Create a one-page HTML reference guide.”

Context & constraints: Audience = beginner, exclude design aesthetics, focus only on technical functions.

▼ Prompt Patterns Used

Break it into parts: Separated the setup, elements, forms, and media.

Format your answer as…: A checklist/table → made it concise and scannable.

Thinking step by step: Structured the guide into a logical sequence.

▼ Systematic Debugging

Failure mode: First draft was too broad and long-form.

Weakness: No format, unclear audience, and no defined outcome.

Refinement: Narrowed the scope to HTML functions, defined a beginner audience, added a checklist/table format, and clarified the outcome.

▼ Progress Recap

Original Prompt: “Act as a web developer planner and place in 1 page everything I need to know about creating a website using HTML…”

Output: long, unfocused text.

Final Prompt: “Act as a web developer planner and create a 1-page HTML reference guide for beginners… format as a checklist/table, focus on technical functions, exclude design aesthetics.”

Output: a concise, scannable HTML Quick Reference Checklist.

The Result :

💡 Reduced research and structuring time from 3 hrs 30 mins manually to 21 minutes with AI (≈90% faster).

💡 Created a reusable system for turning complex technical subjects into practical one-page guides.

Example Output Snippet:

- Page setup → <html>, <head>, <body>
- Forms → <form>, <input>, <button>

The Takeaway:

💡 This workflow turned scattered, overwhelming research into a single page of clarity.

By combining structured prompting and systematic refinement, I can generate quick-reference sheets that help learners (and myself) move from confusion to practical action.

The same approach could be applied to onboarding, training, or upskilling in any professional setting.

Appendix- Full Prompts

10-Day Presentation Prep Plan

▼ **Original Prompt:**

Act as event planner and create a 10 day prep plan for a presentation. I have 1–2 hours daily to spend on this and ensure morning sessions are more focused while evening sessions are lighter. Include rest days.

▼ **Final Prompt:**

Act as an event planner and create a 10-day prep plan for my presentation. I have 1–2 hours per day, with morning sessions capped at 1 hour of deep work (e.g. outlining, building slides, practicing runs). Evening sessions should be lighter (e.g. reviewing slides, visualization, casual reflection). Include at least 2 rest days. The plan should build step by step (outline → slides → practice → polish). The prep should focus on building confidence and clarity in my delivery. Format the output as a checklist, so it's easy to scan and tick off. Keep the plan structured but not overly rigid (avoid making it too open-ended). Present it in a clear style (checklist format, not long narrative). The delivery style should be conversational, engaging, and natural

30-Day Python Challenge

▼ **Original Prompt:**

Act as a supportive challenge designer and design a 30 day course on python. There needs to be some rest days.

▼ **Final Prompt:**

Act as a supportive challenge designer and create a 30-day Python course for beginners. Learners can spend 45–60 minutes per day. The plan should build step by step: fundamentals to logic to problem-solving to small projects to final showcase. Include at least 2 rest or light review days per week for gentler pacing. Add motivation boosts (small wins, reflection prompts, progress celebrations) to keep learners engaged. The focus is on building confidence, clarity, and consistent practice. Format the output as a weekly checklist (Day, Task, Goal, Motivation) so it's easy to tick off. Keep the tone supportive, encouraging, and friendly.

One-Page Website Guide

▼ **Original Prompt:**

Act as a web developer planner and place in 1 page everything I need to know about creating a website using html. Focus on the technical side and functions rather than the aesthetics.

▼ **Final Prompt:**

Act as a web developer planner and create a 1-page HTML reference guide for building a basic website. Audience: beginner learner. Focus only on technical structure and functions of HTML (elements, attributes, page setup, forms, media, integration points). Exclude design aesthetics. Format the output as a structured checklist or table so it’s easy to scan in one page. The outcome should be a functional foundation for a website (ready to connect with CSS/JavaScript later). Keep the tone clear, concise, and practical, like a quick-reference sheet.

Thank you for reading — these case studies are part of my ongoing journey using AI to save time, learn faster, and design better systems.