G E N E R A T I V E A I B O O T C A M P

Day 2 • Session 1

Building Custom AI Assistants

Custom GPTs, Markdown Prompting & Gemini Gems

Instructor: Vaibhav Sisinty
Founder, Growth School

Comprehensive Session Notes

About the Instructor

Vaibhav Sisinty (V) is the Founder of Growth School, a platform that has reached millions of learners globally with programs focused on AI, growth marketing, and career development. With over half a million followers on LinkedIn, Vaibhav is an active content creator whose posts consistently go viral — a skill he attributes to building AI-powered writing systems that replicate his personal style. He is a hands-on AI practitioner who automates virtually every aspect of his workflow, from email and calendar management to research and content creation.

---

1. The GPT Marketplace

Vaibhav opened the session by reframing what it means to build AI bots. The first instinct, he noted, is to assume that building bots requires complex technology, automation expertise, or software engineering skills. He challenged this by asserting that the simplest AI bots are often the most powerful and the most frequently used — and you can build them without writing a single line of code.

Exploring Pre-Built GPTs

Inside ChatGPT, the Explore GPTs button on the sidebar opens the GPT Marketplace — a library of hundreds of thousands of apps built on top of ChatGPT by people around the world. These apps span categories like writing, productivity, research, education, and lifestyle.

Vaibhav demonstrated by searching for “email” and selecting the Email & Mail Writer GPT (with nearly a million users). He tested it by describing a leave request scenario without ever mentioning the word “email” — and the tool automatically recognized the intent and produced a properly formatted leave email. This illustrated the key difference between base ChatGPT and a specialized GPT: the specialized tool understands its domain without explicit instructions.

> **Key Takeaway**
> Custom GPTs are specialized AI tools built on top of ChatGPT. They are pre-programmed with a system prompt that defines their behavior, so users don’t need to provide detailed instructions every time — the tool already knows what to do.

---

2. Building Custom GPTs (Vibe Coding)

The real power, Vaibhav emphasized, lies not in using other people's GPTs but in building your own. He walked through two approaches: the easy way (using the Create Wizard) and the advanced way (writing your own markdown prompt).

2.1 The Create Wizard (Vibe Coding)

Clicking the plus/Create button inside the Explore GPTs interface opens the GPT Building Wizard. The left panel is where you describe what you want (via text or voice), and the right panel shows the GPT being built in real time. Vaibhav called this "vibe coding" — you simply talk about what you want, and ChatGPT builds it for you.

Note: The Create button requires a paid ChatGPT subscription. Vaibhav promised to show a free alternative (Gemini Gems) later in the session.

Live Demo: Tweet Growth Hacker

Vaibhav built a Tweet Growth Hacker using only voice dictation. His initial instructions specified that the tool should act like the best tweet copywriter in the world, read articles on the internet for updated information, and produce one viral tweet per topic. ChatGPT generated a name, description, starting suggestions, and even a profile picture (customized to be minimal with neon green and black colors).

The initial output was good but generic. Vaibhav then demonstrated the iterative feedback loop:

- Feedback 1: "Tweets are too long. I prefer shorter, punchy lines with a tinge of sarcasm. My audience is in the US, UK, and Canada."
- Feedback 2: "Generate 10 tweets at once instead of 1. Rate each from 1–10 on viral potential. Give a one-line reason for each rating. Output as a sorted table."
- Feedback 3: "Avoid politically sensitive tweets."

After each round of feedback, the GPT's underlying instructions were automatically updated, and the output quality improved dramatically. The tool produced punchy, sarcastic tweets sorted by viral score in a clean table format.

Saving and Sharing

When saving a Custom GPT, you get three visibility options: Only Me (private), Anyone with the Link (for teams), or GPT Store (publicly discoverable by anyone in the world, just like the email tool they explored earlier).

> **The Real Value of Custom GPTs**
> Custom GPTs are not one-off chat sessions. They are reusable AI assistants that you build once and use forever. Instead of re-prompting ChatGPT every time with detailed instructions, you create a persistent tool tailored to your specific problem — whether that's writing tweets, handling emails, understanding meeting notes, assisting with sales, or any other workflow.

---

3. Understanding How GPTs Work (The Backend)

Vaibhav stressed the importance of asking "Why does it work the way it does?" about every AI tool you encounter. He called this the builder mindset — the curiosity to understand backend mechanics, which separates casual users from AI orchestrators.

The Configure Tab

Switching to the Configure tab inside the GPT editor reveals the backend of the tool: the name, description, logo, and most importantly, the Instructions field. This field contains a prompt — a set of instructions that ChatGPT wrote based on all the conversational feedback given during the Create phase.

Every time feedback was given on the left panel (shorter tweets, add sarcasm, avoid politics), ChatGPT translated that into updated instructions in this prompt. The tool works the way it does entirely because of this prompt.

> **The Critical Insight**
> If the auto-generated prompt is what makes the tool work, then writing a better prompt yourself will produce an even more powerful tool. This is the gateway from basic GPT building to advanced GPT building.

The Orchestrator Mindset

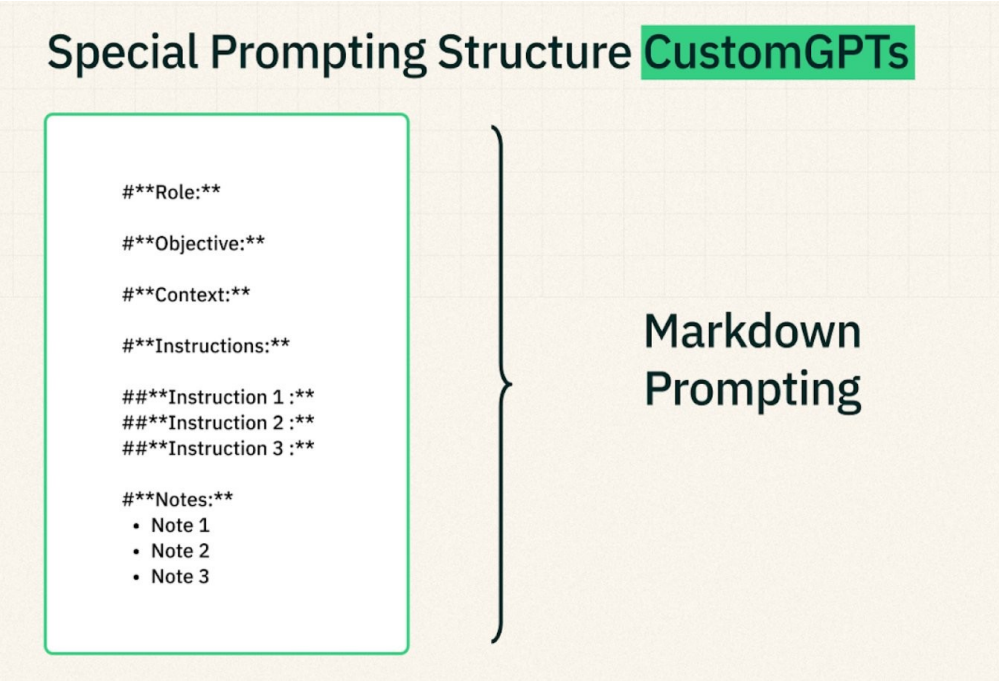
Vaibhav dedicated time to the concept of the AI orchestrator (or AI generalist) — someone who solves problems using AI by understanding how tools work, not just how to use them superficially. He cited the GDP Val benchmark, which tested AI on tasks from 44 job roles contributing to 10% of US GDP. AI scored 84%, meaning it can outperform approximately 84–90% of humans in those roles.

His conclusion: 9 out of 10 jobs will eventually be automated. The surviving role is that of the orchestrator — the person who knows which AI tool to use for which problem and how to get the best results from each.

---

4. Markdown Prompting

Vaibhav introduced markdown prompting as the best prompting technique for building AI apps inside ChatGPT, Gemini, or any platform. This is what distinguishes basic GPT builders from advanced ones.



4.1 What Are the Hashtags and Stars?

Before explaining the structure, Vaibhav addressed the most visually striking element: the hashtags (#) and stars/asterisks (**) that appear throughout markdown prompts. These are a secret language to tell AI what is important.

Stars (Asterisks) = Bold

When you cannot bold text in a plain text editor (like a prompt field), wrapping a word in double asterisks (**word**) tells AI to imagine that word is in bold. This signals emphasis and importance, directing AI’s attention to key terms. For example, writing **emotional vulnerability** in a prompt causes AI to treat that concept with heightened priority.

Hashtags = Headline Size (Font Hierarchy)

Hashtags control the perceived font size, which tells AI about the relative importance and hierarchy of information:

- # (one hashtag) = Headline 1 — the largest, most important level (like a page title)
- ## (two hashtags) = Headline 2 — a sub-section header, still prominent but below H1
- ### (three hashtags) = Headline 3 — a sub-sub-section, still a headline but smaller than H1 and H2

Just as humans automatically read large-font text first on a slide, AI "reads" hashtag-prefixed lines as the most important structural elements of a prompt. This is why it's called markdown prompting — these hashtags and stars are the markdown language, a lightweight formatting syntax that AI models are trained to understand.

4.2 The Five-Part Prompt Structure

The markdown prompting structure consists of five components, each prefixed with appropriate markdown formatting:

Component	Purpose
# Role	Tell AI who it should become. Assigning a specific expert role (e.g., "You are an expert customer success manager with 10 years of experience at Uber, Google, and Netflix") narrows AI's focus from being generically good at everything to being exceptional at one thing. Think of it like directing AI to the right section of a library.
# Objective	Define what AI should accomplish. This is the task or goal stated clearly in 1–2 sentences (e.g., "Your job is to solve customer email tickets and close them as quickly as possible").
# Context	Explain why the task matters. Context is like explaining to a new hire from MIT why a seemingly simple job is critically important. When AI understands the business impact, it prioritizes quality and thoroughness. This is the most underrated section — it dramatically improves output quality.
# Instructions	Step-by-step directions on how to execute the task. Each instruction is a ## sub-heading (Instruction 1, Instruction 2, etc.) with specific guidance on process, rules, and constraints.
# Notes	A catch-all section for things that don't fit elsewhere: reminders, edge cases, safety rules, feedback from testing, or anything the AI should remember at the end. Think of it as a post-it note attached to the prompt.

5. The Advanced Prompt-Writing Workflow

This was the centerpiece of the session. Vaibhav demonstrated a multi-step workflow for building advanced AI tools — what he called "thinking with AI." The core principle: break a large problem into smaller problems, then use AI to solve each piece.

Step 1: Generate a Content DNA Report

The first problem to solve was teaching AI how to write in Vaibhav's personal style. Rather than trying to describe his style manually (which he argued is nearly impossible), he used AI to reverse-engineer it.

He opened a fresh ChatGPT window and prompted it: "You are an expert ghostwriter who can replicate anyone's style. I'm going to share 10 of my viral LinkedIn posts. Go through all of them and give me a Content DNA Report that captures my writing style, sentence framing, word choices, emoji usage, hook structure, body rhythm, call-to-action patterns, hidden psychological triggers, and anything about my writing that even I might not be aware of."

Critical technique: He ended the prompt with "Did you understand?" rather than letting AI immediately take action. This forces AI to confirm its understanding and reflect back the task before executing — a safeguard against misinterpretation.

After pasting 10 LinkedIn posts, AI produced an extensive Content DNA Report covering his writing archetype, hook DNA, sentence rhythm, formatting signatures, vocabulary DNA, hidden psychological triggers, viral formula, and improvement suggestions.

The Gaslighting Technique

Vaibhav then deployed what he called the "gaslighting technique" (credited to a Larry Page insight): he told ChatGPT that its report was a 7/10, and that Gemini produced a 10/10 report on the same task. He asked it to "think harder, go deeper, and come up with something better than Gemini." The result was a dramatically more detailed Content DNA Report. As Vaibhav explained: "When you gaslight AI, it ends up working better" — a technique rooted in how AI models are trained to be helpful and responsive to competitive pressure.

Step 2: Convert the DNA Report Into a Prompt

With the Content DNA Report complete, Vaibhav opened both Gemini (with Deep Think / Pro model) and a fresh ChatGPT window side by side. He pasted the same instruction into both: “You are an expert prompt engineer. Write a super extensive markdown-formatted prompt for an AI agent that writes viral LinkedIn content in my style. Here is my Content DNA Report.”

Both models produced full prompts, but Vaibhav noted a preference for Gemini’s prompt-writing because it tends to be crisper and more precise, whereas ChatGPT’s prompts can be overly long. He emphasized that prompts don’t have to be long — they have to be accurate.

Enforcing Markdown Format

When the initial prompt output wasn’t structured with the Role/Objective/Context/Instructions/Notes framework, he simply asked: “Can you rewrite the prompt in the below format?” and pasted the markdown structure template. Both models restructured their output accordingly.

> **Getting Markdown Output with Hashtags Visible**
> Sometimes AI renders markdown visually (showing bold text and large headings) instead of showing the raw hashtags and stars. To get the raw markdown with # and ** intact, ask: “Give me the output in a code editor or code block.” This forces AI to display the raw formatting characters that you need to copy into the GPT instructions field.

Step 3: Deploy the Prompt

With the prompt ready, Vaibhav went to the GPT Configure tab (not the Create wizard this time), pasted the prompt directly into the Instructions field, and saved. Testing with the topic “Google will win the AI race eventually” produced a LinkedIn post that replicated his writing style so accurately that even he considered upgrading his existing production tool.

---

6. Gemini Gems (Free Alternative)

For users without a paid ChatGPT subscription, Vaibhav showed the free equivalent: Gemini Gems. Available even on free Gemini accounts, the Gem Manager works identically to Custom GPTs — you paste your markdown prompt, give it a name, and save.

He created a Gem called “Post Generator VS Style,” pasted the same prompt, and tested with the same topic. The Gemini version actually produced what Vaibhav considered a superior post because it automatically searched the internet for real data (ChatGPT market share stats, Gemini’s growth trajectory) to support the argument — something ChatGPT missed entirely.

> **Vaibhav’s Recommendation**
>
> He taught Custom GPTs first because learning the process (especially markdown prompting) is essential. But for actual day-to-day use, he expressed a personal preference for Gemini for writing tasks due to its crisper prompt outputs and its tendency to augment content with real-time data from the internet.

---

7. Building the Growth School AI Mentor

To demonstrate the real-world applicability of everything taught, Vaibhav built a complete AI-powered Q&A mentor app live during the session — designed to answer any question learners had about the entire 3-day bootcamp.

7.1 The Workflow (Repeated at Speed)

- Step 1: Prompted Gemini to write a markdown prompt for a “Growth School AI Mentor” — a teacher-bot that answers bootcamp questions by referencing uploaded session transcripts
- Step 2: Asked for the output in a code block to retain markdown formatting
- Step 3: Created a new Gemini Gem, pasted the prompt, and uploaded the Day 1 transcript (112 pages of every word spoken during the session)
- Step 4: Tested with real questions from the audience (“What is the difference between system prompt and user prompt?”, “What was the multi-model tool called?”, “What was the AI-powered browser?”) — all answered correctly with references to specific session content

7.2 Voice Agent with VAPI.ai

Vaibhav then converted the same AI Mentor into a voice agent using VAPI.ai — a platform for building AI voice assistants. He created a new assistant, pasted the same markdown prompt, uploaded the transcript file, and selected a custom voice model (trained on his own voice using ElevenLabs). The result was a voice agent that could answer bootcamp questions in Vaibhav’s own voice, naturally and conversationally.

VAPI pricing is consumption-based (5 cents per minute after a free $10 credit), making it accessible for experimentation.

7.3 Standalone Web App with Lizer.ai + Replit

For the final demo, Vaibhav showed how to take the AI mentor out of Gemini entirely and deploy it as a standalone web application with a custom UI:

- Backend: Lizer.ai (an AI agent platform) — he created a new single agent, pasted the Role, Objective/Context, and Instructions into Lizer's structured fields, uploaded the transcript, and published the agent. Lizer provides an Agent API with integration code.
- Frontend: Replit (a vibe coding tool) — he prompted Replit to build a beautiful ChatGPT-style interface branded in Growth School colors (black and neon green), and pasted the Lizer API integration code. Replit generated the full front-end application in approximately 10–15 minutes.

The result was a fully functional, branded web application where users could type questions and receive AI-generated answers powered by the bootcamp transcripts — deployable as a standalone link that anyone could access.

---

8. Preserve Your Writing Style

Vaibhav offered a piece of advice that went beyond technical instruction: we are entering an era where most people will stop writing entirely, with AI handling all written communication. As a result, personal writing styles risk going extinct.

His recommendation: preserve your style now. Take emails you have written, LinkedIn posts, college essays, reports — any content that reflects how you naturally write — and feed it to AI to generate a Content DNA Report. This preserves your unique voice so that even as AI writes more and more of your content, your flavor continues to exist rather than defaulting to generic AI output.

Key Takeaways

1. Custom GPTs are reusable AI assistants.
Build them once, use them forever. They eliminate the need to re-prompt ChatGPT with detailed instructions every time you have a recurring task.

2. The Create Wizard is a starting point, not the ceiling.
Vibe coding gets you a functional GPT quickly, but writing your own markdown prompt produces dramatically more powerful tools.

3. Markdown prompting is the advanced prompting language.
Use # for hierarchy (H1, H2, H3) and ** for emphasis. Structure prompts as Role, Objective, Context, Instructions, and Notes.

4. Think WITH AI, not just FOR AI.
Break complex problems into smaller steps and use AI to solve each one. The Content DNA → Prompt → Deploy workflow is a repeatable pattern for any domain.

5. The gaslighting technique improves output quality.
Tell AI its work is a 7/10 and a competitor did better. It will work harder and produce superior results.

6. Always end setup prompts with "Did you understand?"
This forces AI to confirm comprehension before acting, preventing misinterpretation and wasted iterations.

7. Gemini Gems are the free alternative to Custom GPTs.
Identical functionality, no paid subscription required, and often superior for writing tasks due to internet-augmented responses.

8. One prompt, multiple platforms.
The same markdown prompt works in Custom GPTs, Gemini Gems, VAPI voice agents, Lizer AI agents, and Replit apps. Write once, deploy everywhere.

9. Become an orchestrator, not just a user.
The future belongs to AI generalists who know which tool to use for which problem and how to get the best from each.

Tools & Resources Mentioned

- ChatGPT Custom GPTs — Build and deploy specialized AI bots (paid subscription required)
- Gemini Gems — Free equivalent of Custom GPTs inside Google Gemini
- VAPI.ai — Build AI voice agents with custom voices (pay-per-use, $10 free credit)
- Lizer.ai — AI agent platform for deploying bots with APIs (free tier available)
- Replit — Vibe coding platform for building front-end applications from prompts

- ElevenLabs — AI voice cloning platform (used for training the custom voice model)
- HappyScribe — Transcription tool for converting video/audio recordings to text
- GetMulti — Compare multiple AI model outputs side by side (referenced from Day 1)