

# WhatToEat+: Using AI to Turn Grocery Receipts into Recipes



Caitlin Heggerud

[Follow](#)

14 min read · May 11, 2025



For Professor Mohammed Keyhani. This was a course project in the Master of Management and the Bachelor of Commerce at the Haskayne School of Business, University of Calgary.

*Created for ENTI 333–633 | Haskayne School of Business, University of Calgary*

*By Caitlin Heggerud, Shirong Huang, Ashley Jung, Lily McCoubrey, and Shirley Xin*

. . .

What if your grocery receipts could plan your next meal?

Our team set out to explore how AI can simplify daily life. We chose to approach one of the most relatable, recurring frustrations: deciding what to

eat.

Our team developed **WhatToEat+** — an AI-assisted web app that transforms scanned grocery receipts into a personalized recipe generator based on what you have in your house.

Our project merged creativity, business thinking, and modern AI development tools to tackle a common daily challenge: deciding what to eat — quickly, healthily, and without waste.

. . .

## **Table of Contents:**

Meet the Team

The Problem

Our Solution

Demo Video

How We Build WhatToEat+ Using AI

Example Prompts

The Idea Development Journey

Lessons Learned

What is Next?

. . .

### Meet the Team:

Caitlin Heggerud — Oversaw the deployment and blog, creating access to the world

Shirong Huang — Managed the API to read the receipts, a lengthy process

Ashley Jung — Led the problem definition, ensuring the project's focus

Lily McCoubrey — Led the AI prompt engineering, taking the coding steps of the project

Shirley Xin — Oversaw the presentation, making sure our idea was explained perfectly

. . .

### The Problem:

In today's fast-paced world, the act of choosing what to eat has become surprisingly difficult. According to the *New York Post*, the average person makes about **35,000 decisions per day**, many of which are small, repetitive, and mentally draining. Among them, food-related choices stand out — not just because of their frequency, but because of the cognitive and emotional

load they carry. In fact, research shows that people spend more than **2.5 hours each week** simply deciding what to eat, which contributes to decision fatigue, impulsive choices, and even skipped meals. Over time, this daily strain takes a toll on both physical health and mental well-being.

Busy lifestyles only make this worse. With so many people juggling work, school, family responsibilities, and social obligations, finding the time to plan, shop for, and prepare healthy meals is increasingly difficult. These **time constraints** lead many to default to takeout or processed foods, not because they want to, but because they feel they have no other realistic option. **Budget limitations** also play a major role in food decisions. As grocery prices rise, especially for fresh or specialty items, people are often forced to make cost-driven choices that prioritize short-term savings over long-term health, leading to less nutritious eating patterns.

Even when time and money aren't the primary barriers, **information overload** can still be paralyzing. From keto to vegan to low-FODMAP, the internet is saturated with conflicting dietary advice and wellness trends. What one source claims is healthy, another says to avoid. This contradictory information leaves many unsure of what to trust, complicating decisions that should feel simple. Add to that the challenge of **overwhelming choices** — aisles of products, dozens of cooking apps, thousands of online recipes — and even the most motivated eaters can feel stuck.

For the **33 million Americans living with food allergies**, the stakes are even higher. Food choices for these individuals aren't just about health and preference; they're about safety and survival. With **1 in 10 adults** and **1 in 13 children** affected, the need for meal planning tools that can automatically filter out allergens and offer safe alternatives is more urgent than ever.

All of these factors feed into broader patterns of **unhealthy habits**. When people are stressed, busy, and unsure, they're more likely to skip meals, overeat emotionally, or rely on fast food. This contributes to long-term health issues and reinforces a negative cycle of decision fatigue and poor nutrition.

Our team recognized that this combination of **time scarcity, budget concerns, conflicting advice, choice overload, allergy risks, and unhealthy routines** represents a real and recurring pain point in people's lives. We saw an opportunity to address this complex challenge with a simple, user-friendly tool — one that uses AI to turn your grocery receipt into personalized, practical, and safe meal suggestions. WhatToEat+ was our answer to a deceptively simple question that everyone struggles with: *What's for dinner?*

. . .

## **Our Solution:**

**WhatToEat+** is a web-based application that transforms a standard grocery receipt into an intelligent meal-planning assistant. With just a photo upload, the app uses a custom-built API and advanced AI models to scan and extract all recognizable food items, creating a real-time digital inventory of what's currently in your kitchen. This means no more guessing what's in the fridge — or forgetting that ingredient you bought last week.

Once your inventory is created, the app provides tailored recipe suggestions based on the food you actually have. These suggestions are powered by integrated AI tools and are designed to reduce decision fatigue, cut down on

food waste, and promote healthier, home-cooked meals. To ensure dietary needs are respected, users can filter recipes by common allergens like dairy, gluten, or nuts, and also by meal type — whether you're planning for breakfast, lunch, dinner, snacks, or desserts.

Recognizing that every household is different, WhatToEat+ allows you to adjust recipe serving sizes to fit your needs — whether you're cooking for one or feeding a large group. You can also manually enter expiry dates for any food item, helping you stay ahead of spoilage and prioritize what needs to be used first. For added personalization, the app includes the option to favorite recipes you enjoy, so you can easily return to them without searching. It's perfect for creating a collection of reliable, go-to meals.

In addition to inventory and recipes, the app helps users track their grocery spending. By analyzing receipt data, WhatToEat+ provides insights into your purchasing patterns and helps you stay within budget — an especially valuable feature for students or families managing rising food costs.

And when you're not sure what you feel like eating, the built-in “**Feeling Lucky**” feature generates a random recipe based on your current ingredients. It's a playful way to break out of routine, try something new, and keep mealtimes interesting.

Altogether, WhatToEat+ combines practical functionality with AI-driven intelligence to help users make better food decisions, eat more intentionally, and reduce waste. It's a simple yet powerful answer to one of the most common — and most frustrating — questions of daily life: *What should I eat today?*

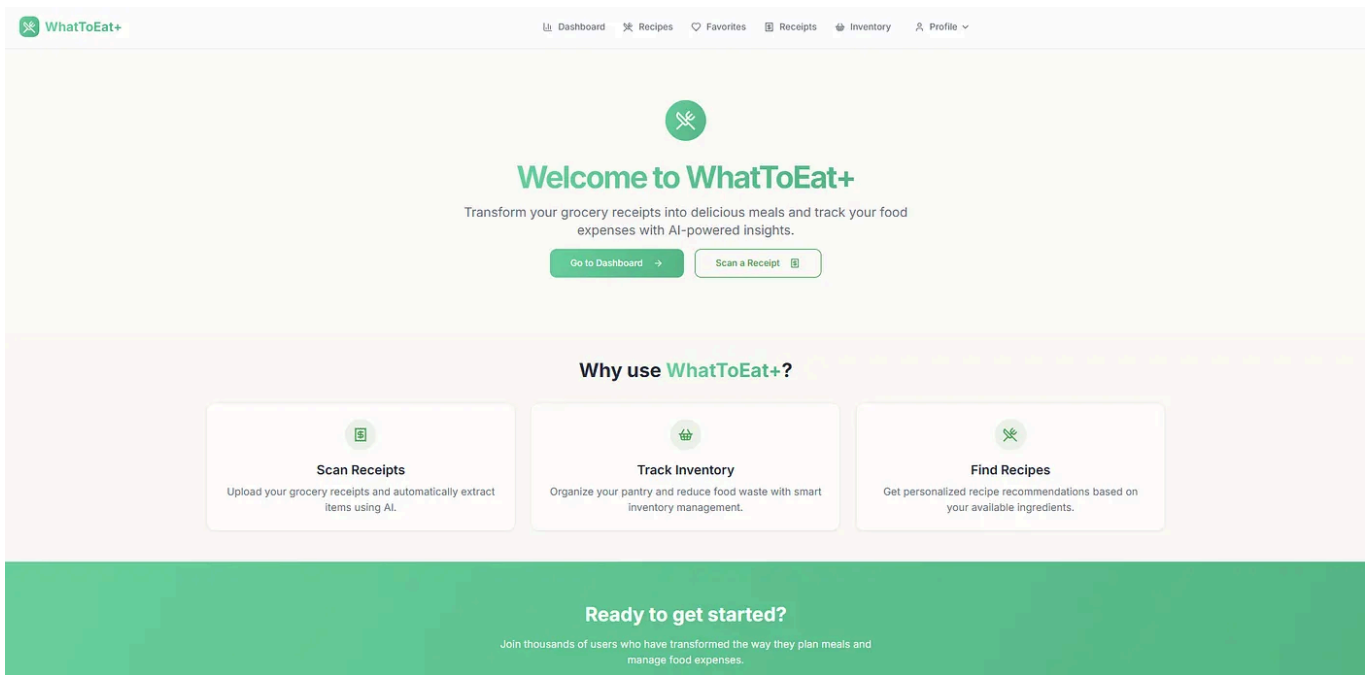


Image 1. Photo of the landing page of WhatToEat+

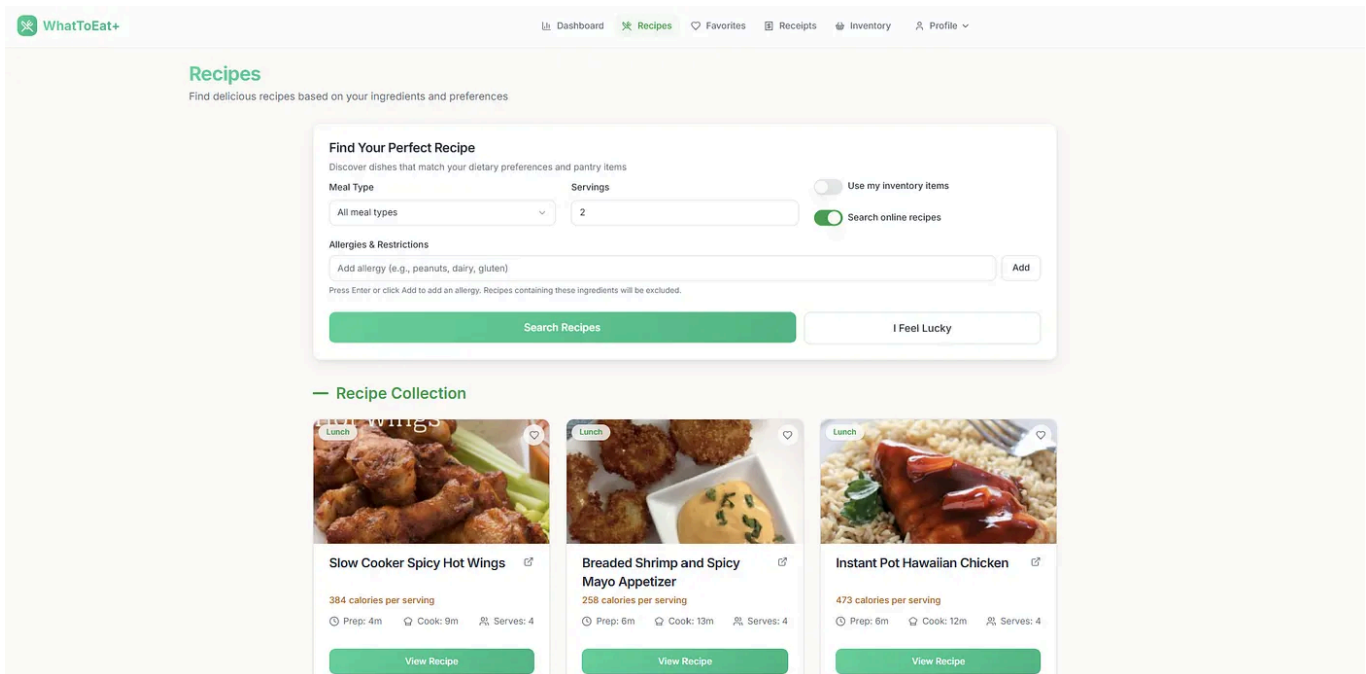


Image 2. Photo of the recipe page of WhatToEat+

• • •

Demo Video:

See WhatToEat+ in action.

WhatToEat+ AI Web App Demo – ENTI 333/633



. . .

### How We Built WhatToEat+ Using AI:

Throughout the development of **WhatToEat+**, our team approached the project not just as builders, but as learners — applying core concepts from the ENTI 333–633 course to turn an everyday problem into an AI-powered solution. From the initial idea to the final code, **AI was our co-creator, our tutor, and our collaborator.** This experience gave us firsthand insight into what Professor Keyhani called “*the generative AI disruption*” — where non-



technical teams can build high-functioning applications by thinking like designers and prompting like engineers.

We began with **ideation**, using ChatGPT to help us brainstorm, refine our problem statement, and differentiate our app from existing tools. Guided by what we learned about *machine psychology* and *prompt engineering*, we didn't just ask AI to build something — we asked it to challenge us. One of our earliest prompts was: *How can we design an app that does more than just suggest recipes?* That question — structured carefully using the *Prompt Crafting Framework* — helped us ground our solution in **real grocery data**, leading to the idea of scanning receipts to generate inventory-based recipes.

During **frontend development**, we built the user interface in Replit using HTML, CSS, and JavaScript. With the support of AI-assisted tools like Replit's AI Agent and GitHub Copilot, we rapidly prototyped features like upload buttons, filter menus, and a responsive design. We used prompt engineering principles to iteratively improve our AI-generated components, applying techniques like “few-shot prompting” and “step-by-step breakdowns” to refine behavior. AI also helped us write the logic for filtering recipes by allergen, adjusting serving sizes, and allowing users to favorite meals — a feature it directly recommended during early design consultations.

On the **backend**, one of our biggest technical challenges was parsing and interpreting real receipt images. To tackle this, we used optical character recognition (OCR) to extract the raw text from the images. At the final stage, we applied Gemini Flash to process and clean the extracted data. This aligned with our lecture content about how large language models are becoming multimodal, with the ability to “see” and “understand” visual inputs like receipts. With AI's help, we transformed raw text into a structured inventory that could power the rest of the app.

We also experimented with aspects of **Retrieval-Augmented Generation (RAG)** — combining real-world user input (e.g., receipt text and manually entered expiry dates) with an AI model’s generative capabilities. Instead of fine-tuning a large model (which we learned can be costly and inefficient), we opted for a *lightweight, modular approach* powered by prompt engineering. This aligned with Keyhani’s point that **prompting, when done well, can outperform fine-tuning for many applications.**

Beyond technical work, AI helped us decide *what* to build. It recommended features like **manual expiry tracking, budget insights**, and the “**Feeling Lucky**” button for generating surprise recipes — all of which became core parts of our app. These features weren’t just practical — they were rooted in our problem analysis, where AI encouraged us to think about **decision fatigue, cognitive load**, and the psychology of food choice.

Throughout the process, we used **GitHub** for version control, leveraging GitHub Copilot to assist with commits, code readability, and our AI-assisted `README.md`. We also practiced **collaborative prototyping**, a concept we learned in class that emphasized simultaneous individual development followed by group consolidation. Initially, we each started our own prototype using different prompts. Once one version began to take shape, we converged on that codebase and continued building as a team—translating and reworking features along the way.

Our learning didn’t stop at code. Professor Keyhani’s lecture on **AI ethics and superintelligence** reminded us to consider broader implications: data privacy, bias, and the line between assistance and overreach. While our app is simple, we were intentional about keeping the user fully in control — no surprise outputs, no autonomous agents, just helpful suggestions based on visible inputs.

Ultimately, WhatToEat+ was more than a final project. It was our case study in **learning by building**, where AI was both a subject of study and an active partner in creation. The course taught us that we don't need to be software developers to make software anymore — we need to be creative thinkers, careful prompters, and responsible builders. AI did not replace us; it **amplified us**.

. . .

### Example Prompts:

As non-technical developers, our team relied on something we all knew how to use: language. We approached the development of WhatToEat+ through **clear, task-based prompting**, using ChatGPT to help us write the kinds of prompts that Replit's AI coding tools could understand and act on. This approach aligned with Dr. Keyhani's lessons on *Prompt Engineering and Machine Psychology*, especially the idea that prompting is not just about getting code — it's about *thinking clearly and communicating precisely*.

We began our project with each team member developing their own prototype, experimenting with different prompts. When one version successfully parsed receipt images, we collectively adopted that version — even though the initial prompt had been written in another language. Replit helped us translate the codebase to English, preserving both logic and structure. From there, we refined the app feature by feature — prompt by prompt.

What made our approach effective was **simplicity and specificity**. Rather than asking for complex systems, we focused on concrete, understandable

tasks. Here are real examples of prompts we used:

- *“Add a button titled ‘Add Expiration Date’ in the Inventory section, please make it functional.”*
- *“Create a `calculateSpendingByCategory()` function that takes an array of items and returns an object like { Produce: 12.50, Dairy: 5.20 }.”*
- *“I want the dashboard to update based on the prices in ‘View Details’. Please only add what’s necessary — don’t change or rewrite anything else in the app.”*
- *“Formatting error with the date — I do not want a date unless it is manually added.”*

We quickly learned that being specific about **what not to change** was just as important as describing what we wanted. Prompts like *“Please do not overwrite or refactor any existing components”* helped prevent the AI from breaking functional features or redesigning layouts unnecessarily. Once we added guardrails like this to our prompts, the output became more targeted, reliable, and easier to integrate.

One of the most effective parts of our process was using **ChatGPT to help write the prompts we used in Replit**. We would first describe the problem or idea in ChatGPT, ask for help writing a clear prompt for Replit, and then copy that result into the Replit environment. This made our requests more organized and better understood by the coding assistant.

When the AI didn’t understand what we meant or returned errors, we would **attach screenshots or describe visual problems** to clarify. For example, if a dashboard section wasn’t updating correctly, we’d show the issue and explain what was missing — like “the prices aren’t being pulled from the receipt

data.” Adding visual context helped the AI locate bugs faster and offer more accurate solutions.

Our prompting style evolved to include structured, layered requests like:

- *“Don’t remove or refactor my existing Receipt components or UI. I want to manually plug this in to my current system.”*
- *“I need a utility function that calculates totals per category, an example of how to store it in context, and a JSX example for rendering.”*

These were not advanced engineering requests — they were clear user stories written in accessible language. This speaks directly to what we learned in ENTI 333–633: **good prompt engineering isn’t about complexity — it’s about clarity, structure, and iteration.**

Ultimately, we didn’t need to be developers to build a working app. We just needed to describe our ideas well, test often, and adjust our prompts until the AI became a true collaborator. The process was not about controlling the AI — it was about learning to guide it.

. . .

## The Idea Development Journey:

Our original concept for WhatToEat+ was centered on reducing food waste by tracking expiry dates and recommending recipes based on which ingredients were about to go bad. We even envisioned a feature where users could take a photo of their fridge, and the app would use AI to identify the food items inside. However, we quickly discovered the limitations of current

AI tools — interpreting fridge images was unreliable, and the models often misidentified or missed key items entirely.

We also ran into significant challenges when trying to capture expiry date information. Most groceries don't print expiry dates on packaging or receipts, and manual tracking proved too burdensome for users. These obstacles prompted us to pivot.

With support from ChatGPT during our brainstorming sessions, we arrived at a more feasible and still impactful idea: scanning grocery receipts. ChatGPT helped us explore the strengths and limitations of this approach, and we realized that receipt data was structured, consistent, and could be easily parsed by AI. We tested the concept, and ChatGPT successfully identified items and quantities from sample receipts. This confirmed that a receipt-based inventory system was a practical foundation for our app.

While we had to let go of automated expiry tracking for now, we preserved the option for users to manually input expiry dates — recognizing that this could be a future enhancement once better data sources or integrations become available.

One of the most valuable parts of working with ChatGPT in this stage was asking a key question: *How is our app providing value beyond simply using ChatGPT directly?* That question kept us focused. It reminded us to build meaningful features and avoid creating an “AI wrapper” that merely repackaged existing tools. Instead, we focused on integrating AI in ways that solved real user problems — bridging grocery receipts and personalized meal planning in a way that felt both useful and unique.

## Lessons Learned:

The process of building WhatToEat+ offered us valuable insights that extended well beyond coding. As students without formal technical backgrounds, we learned not just how to use AI tools, but how to think critically, troubleshoot effectively, and design with both users and machines in mind.

One of our early realizations was that APIs, while incredibly powerful, are not always easy to work with. We faced challenges around authentication, rate limits, and inconsistent or vague documentation. Integrating APIs often required incremental testing, reading between the lines of documentation, and developing a stronger understanding of how external systems communicate with one another. We learned to be patient, to read carefully, and to debug one step at a time.

We also discovered that while AI is a powerful accelerator, it is not infallible. It helped us write code, fix bugs, and generate new ideas, but it also made mistakes — sometimes small, sometimes significant. We started noticing patterns in how it responded: it sometimes misunderstood our intent or made confident guesses that turned out to be wrong. This made us more thoughtful in how we phrased our prompts, and more critical in how we evaluated its suggestions. We realized that AI can be persuasive — even when it's wrong — so maintaining a human-in-the-loop approach was essential.

Prompting, we came to understand, is more than just inputting instructions — it's a form of thinking. As we worked, prompting evolved into an iterative conversation where we clarified, revised, and reasoned through problems alongside the AI. It wasn't just about making the AI do things; it was about

using AI to *think with us* as we designed, tested, and improved each feature. This approach helped us move faster while still staying in control of the outcomes.

Another major takeaway was the importance of prompt quality. Specific, structured prompts with clear context produced far better results than vague or general ones. We also learned that including constraints — such as “don’t change anything else in the app” — prevented the AI from making unnecessary or disruptive edits. Adding screenshots or describing exactly what we wanted to change made a noticeable difference in the AI’s ability to help effectively.

We also found that simplicity often led to the best outcomes. When we tried to do too much at once or over-automate certain features, the app became harder to use and harder to maintain. The features that worked best were clean, intuitive, and easy to build upon. This reinforced the value of designing for clarity — both in code and in user experience.

Finally, we kept returning to one key question that guided our development from start to finish: *What value are we adding that someone couldn’t already get from ChatGPT directly?* That question grounded our project and helped us avoid creating an app that was just an AI wrapper. It pushed us to design features — like receipt parsing, inventory tracking, recipe filtering, and manual expiry entry — that solved specific problems in everyday life.

Altogether, these lessons gave us a deeper appreciation for the creative and collaborative nature of working with AI. We didn’t just use tools — we learned how to guide them, challenge them, and build with them.



## What is Next?

We believe WhatToEat+ has strong potential for real-world application and continued development. Looking ahead, our next priority is to explore different monetization strategies — specifically comparing direct user payments (such as subscriptions or premium features) with advertising-based revenue models. This will help us determine how to make the app both sustainable and accessible for our target users.

Another key focus is improving the accuracy of our OCR (optical character recognition) and receipt-parsing systems. Currently, WhatToEat+ performs best with mock receipts generated by ChatGPT, but real-world receipts can be far more complex and varied. Enhancing the app's ability to reliably interpret actual receipt formats is a necessary step toward wider adoption and greater user trust.

# GROCERIES

STEAK	1	79	7.99
POTATOES	5	0.79	3.95
SALMON	2	5.79	11.58
CABBAGE	1	12	1.29
BROCCOLI	3	1.49	4.47
CHICKEN	1	69	6.49
PEAS	1	19	1.19

---

SUBTOTAL	36.96
TAX	1.81
<b>TOTAL</b>	<b>\$38.77</b>





Image 3. Photo of a mock receipt made with ChatGPT

We also plan to integrate **e-receipt functionality**, allowing the app to automatically process receipts sent via email. This would reduce the manual burden on users and streamline the inventory-building process. In the longer term, we envision expanding WhatToEat+ beyond post-shopping assistance. One of our goals is to introduce **pre-shopping recipe recommendations** — helping users plan meals before they head to the store, and generating intelligent grocery lists to support healthier, more efficient shopping.

While we started with a simple concept, we now see WhatToEat+ as a scalable, AI-powered platform that could help users reduce food waste, improve nutrition, and make smarter food decisions — before and after checkout.



**Written by Caitlin Heggerud**

0 followers · 1 following

Follow

---

**No responses yet**




Derrick Howell

What are your thoughts?

## Recommended from Medium




 In How To Profit AI by Mohamed Bakry

### Your ChatGPT History Just Went Public on Google. Here's What I Di...

Safety/Privacy Check Prompt Template Is Included

★ Aug 9 🖱️ 7.4K 💬 233  

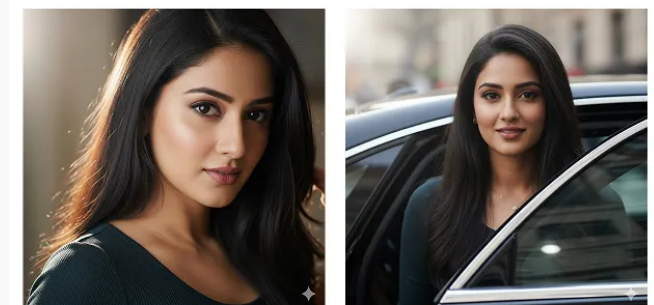
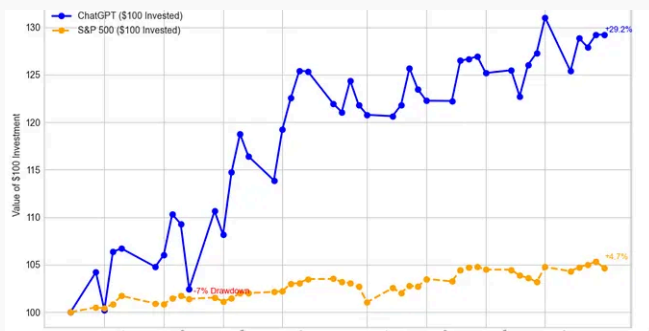


 In Long. Sweet. Valuable. by Ossai Chinedum

### I'll Instantly Know You Used Chat Gpt If I See This

Trust me you're not as slick as you think

★ May 16 🖱️ 23K 💬 1370  





In Coding Nexus by Civil Learning

## I Handed ChatGPT \$100 to Trade Stocks—Here's What Happened i...

What happens when you let a chatbot play Wall Street? It's up 29% while the S&P 500...



Sep 2



1.5K



33



In LearnAltoprofit.com by PRADIP KAITY

## 12+ Practical Ways to Turn Nano Banana into Real Income

Read this article for FREE by Clicking [HERE](#).



Sep 3



1.4K



48



Harishsingh

## Why Every Senior Developer I Know Is Planning Their Exit

After 10 years in software development, with the last three in high-frequency trading, I'm...



Sep 4



1.6K



76



Yashwanth Sai

## I have built around 300 agents, worked at 5 startups. Here's what ...

Lessons learnt after working with agents for over an year.



Aug 24



3.3K



115



[See more recommendations](#)