



# AI INNOVATION CIRCLE

**Welkom!**

# Start with an **AI-first mindset** today

How AI is changing the way we work and how our business operates



There will be **two kinds** of companies at the end of this decade:

Those that are **fully utilizing AI**,  
and those that are **out of business**.

AI isn't just another tool — it's fundamentally transforming our business. Organizations must adapt or risk obsolescence in this rapidly evolving landscape.







# Agenda

- 1 AI (r)evolution**  
From origin to where we are today
- 2 Understanding AI technology**  
How does Generative AI work?
- 3 Tips and practical applications**  
How you can effectively use AI in your daily work





# AI (r)evolution





**1450 → 1700**

The printing press  
revolutionized the  
dissemination of knowledge



**1870 → 1950**

Electricity revolutionized  
industry and households  
*(basics invented in 1800)*



**1950 → 2000**

The computer entered office  
and home  
*(mobile revolution follows)*



**1700 → 1850**

The steam engine brought  
about an industrial revolution



**1876 → 1950**

Telephony transformed global  
communication



**1990 → 2005 → 2020**

The internet brought us into  
the information age  
*(after ± 20 years of research)*





# Why is the AI revolution different?

## 1 — Broad impact

Affects almost every sector; transforms cognitive, creative and physical work

Automates repetitive work, complex decision-making and creative processes; a revolution of intelligence

## 2 — Accessibility

Available to individuals and small organizations, not just large players.

Expands capabilities, automates entire workflows and accelerates innovation.

Fundamentally changes economic, social and political power structures.

## 3 — Self-development

Unique ability to accelerate its own progress.

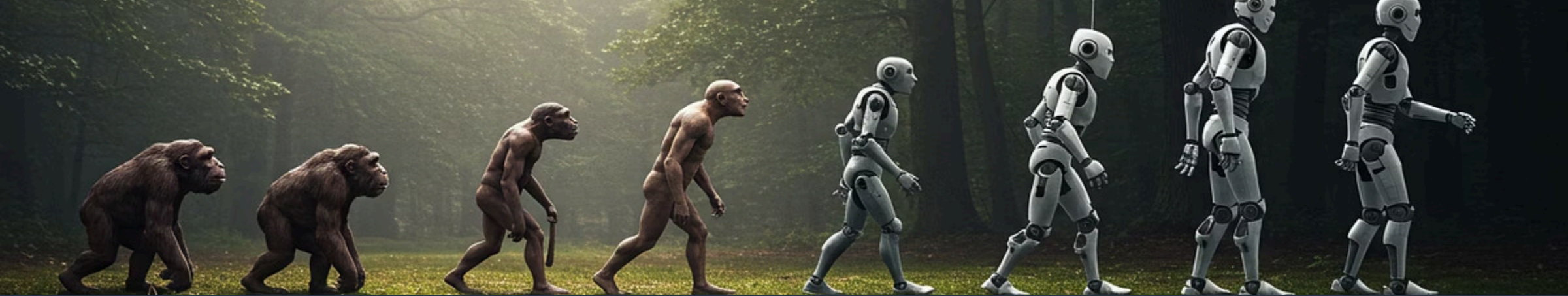
Leads to exponential growth.

## 4 — Speed of change

Adoption curve is dramatically steeper than previous technologies.

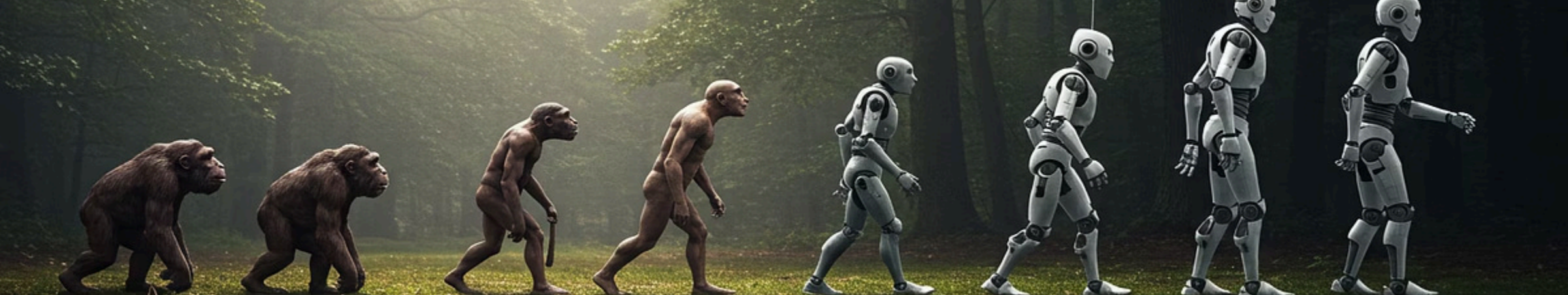
Computing power and capabilities double every few months.

Transformations that once took generations now happen within months.



# Adoption





# Development (capabilities)





# Major Large Language Models (LLMs)

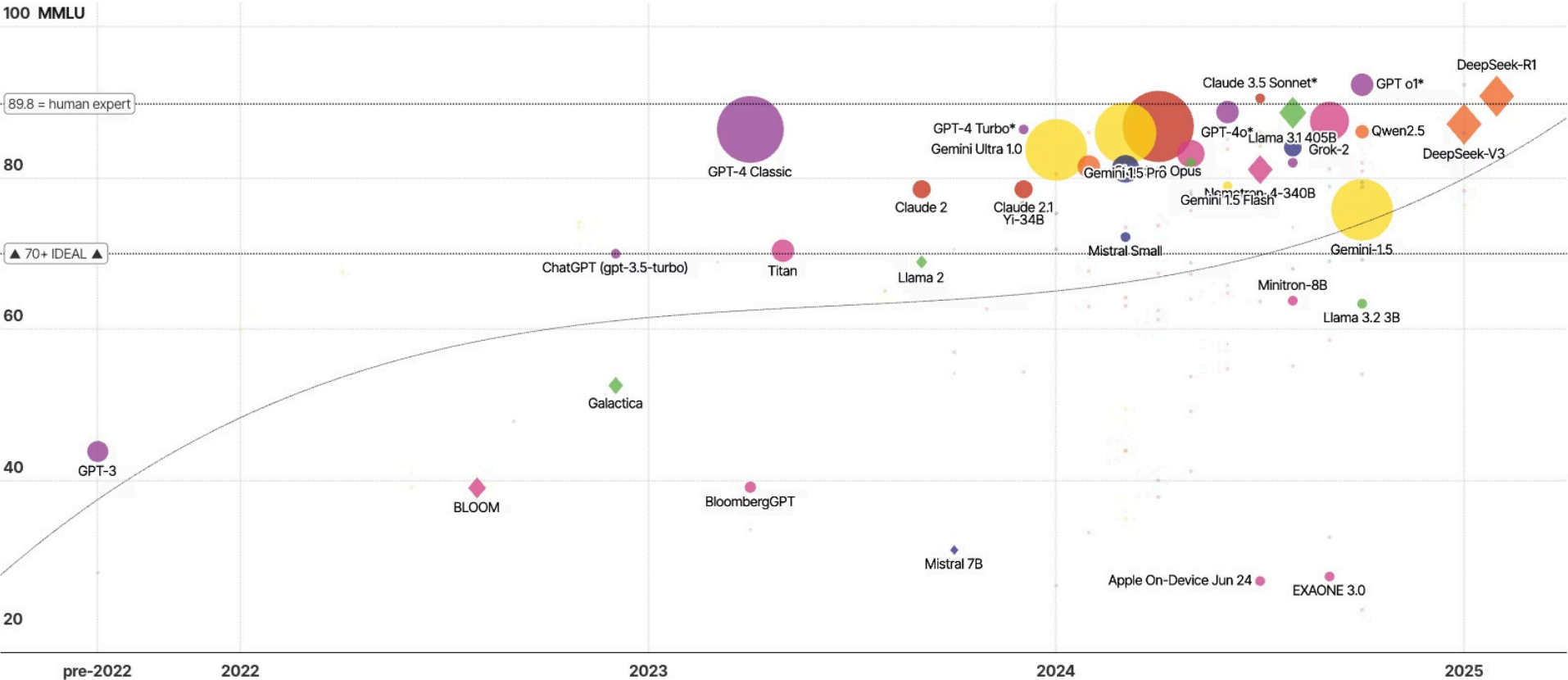
ranked by capabilities, sized by billion parameters used for training

CLICK LEGEND ITEMS TO FILTER

anthropic chinese google meta microsoft mistral openAI other

search...

show only: significant models



David McCandless, Tom Evans, Paul Barton  
Informationisbeautiful // Jan 2024

MMLU = benchmark for measuring LLM capabilities  
\* = parameters undisclosed // source: [LifeArchitect](#) // data

MADE WITH VIZsweat



# Evolution of AI Capabilities

In recent years, we have seen an explosive growth in AI capabilities.

## 2022-2023: Foundation



ChatGPT brings AI to the general public.

Meta's open-source model LLaMA gives everyone access to powerful AI.

GPT-4 introduces multi-modal capabilities and improved reasoning.

## 2025 H1: Further Evolution



Advanced Reasoning in models like OpenAI's o1 and o3 (essential for AI agents).

AI agents are more autonomous with external tools.

More multi-modal and multi-language support in all leading models.



## 2024: Multi-modal Revolution

GPT-4o combines text, image, and audio in a single model.

Claude 3 excels at coding.

Gemini 1.5 Pro offers enormous context windows of 1 million+ tokens.

LLaMA 3 significantly improves open-source capabilities.

## 2025 H2: Expectations

More parameters

Larger context windows

Further development of multi-modal

Further development of advanced reasoning

... Big surprises

## Generative Text Model (2018-2020)

Early transformers like GPT-2 and GPT-3 only generated text



## Chat & Instruction Following (2021-2022)

Instruction fine-tuning and human feedback (RLHF)



## Vertical Specialization (2022-2023)

Domain-specific models for code, law, and healthcare



## Multimodality (2023-2024)

Integrating text and images; document analysis and visual search become possible



## Long-context & Tools (2024)

Context windows grow to million+ tokens  
Coding agents become much better



## Real-time Audio & Image (2024-2025)

Processing speech and images with minimal delay



## Video & 3D (2025→)

Creation of photorealistic images and videos



## Intelligent Agents (2025+)

AI controls physical systems via natural language



# AI Agents

## 2025

- Standards for communication between Agents and Systems (MCP)
- Standards for communication between Agents and Agents (A2A)
- Accessible frameworks for multi-agent development

## 2026 and beyond

- Agents seek integration with majority of online platforms
- Search and navigation paradigm will shift. SEO becomes GEO / LLMO
- Agent-mediated commerce: AI books your weekend getaway

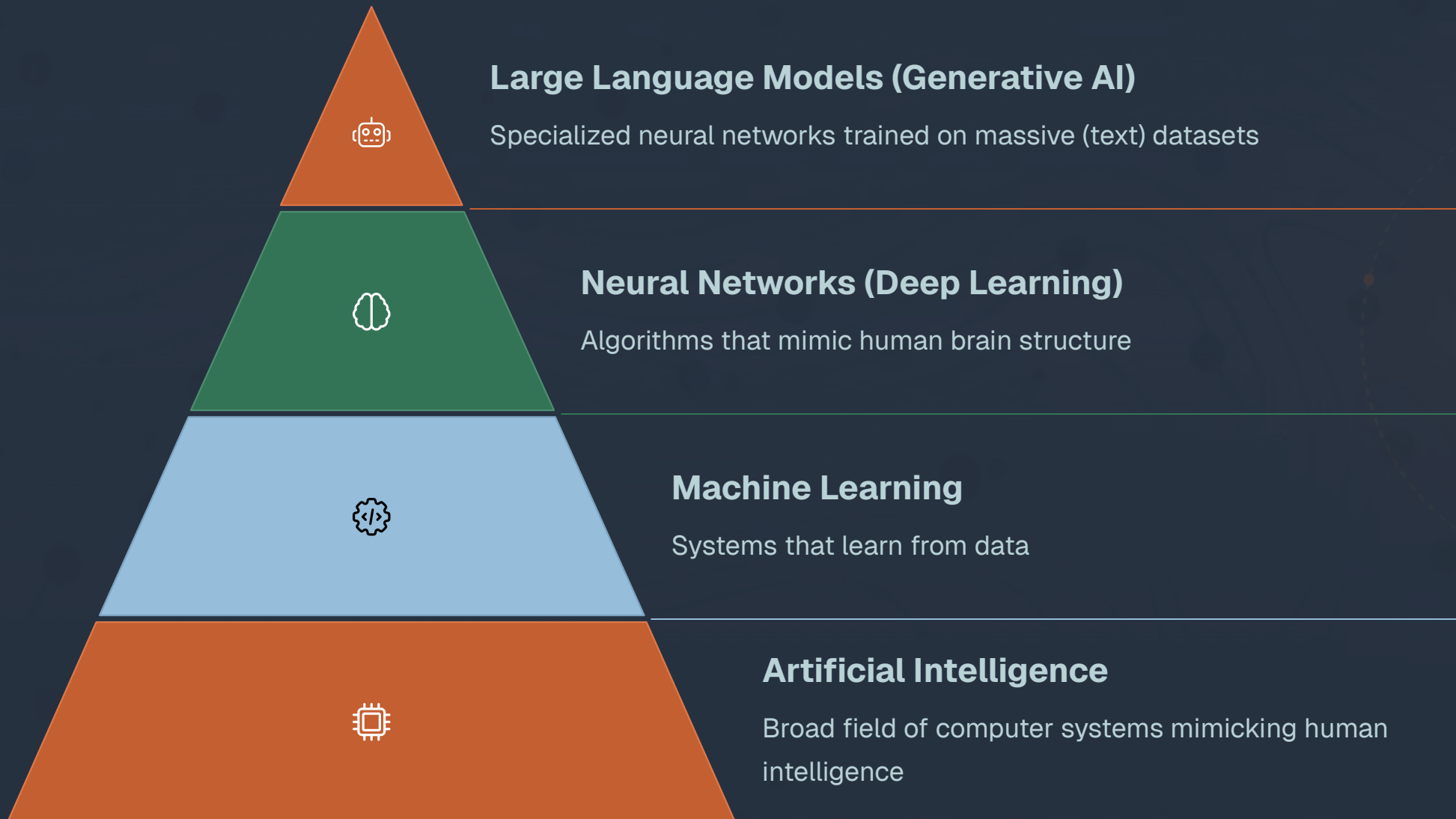


# How does it work?

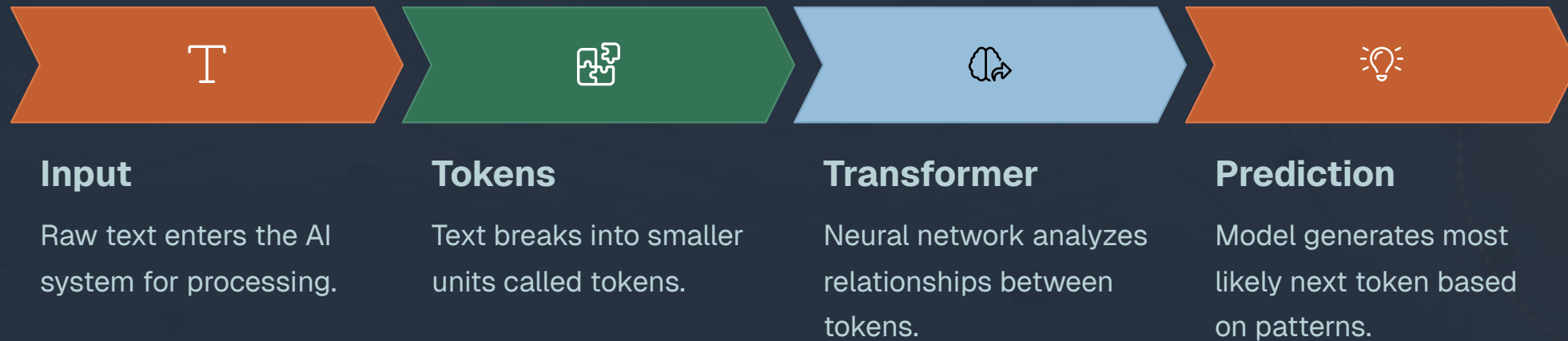




# The Hierarchy of AI



# How do LLMs work?



This process repeats thousands of times per second, creating coherent text one token at a time.

# How are LLMs trained?

Large Language Models undergo a sophisticated multi-stage training process before they can generate human-like text.

## Pre-training



The model learns language patterns by reading billions of texts and predicting the next word.

Requires thousands of GPUs running for months

✓

Processes trillions of examples ✓

## Alignment Training



Human feedback teaches the model to generate helpful, honest responses.

Humans rate outputs as helpful or harmful ✓

Reduces potentially dangerous content ✓

## Fine-tuning



The pre-trained model specializes on high-quality datasets for specific skills.

✓ Faster than pre-training but crucial

✓ Improves instruction-following and reasoning



Estimated training cost for GPT-4: >\$100M

Costs for GPT-4o were ~\$10M due to advances in training efficiency and infrastructure

# Fun facts

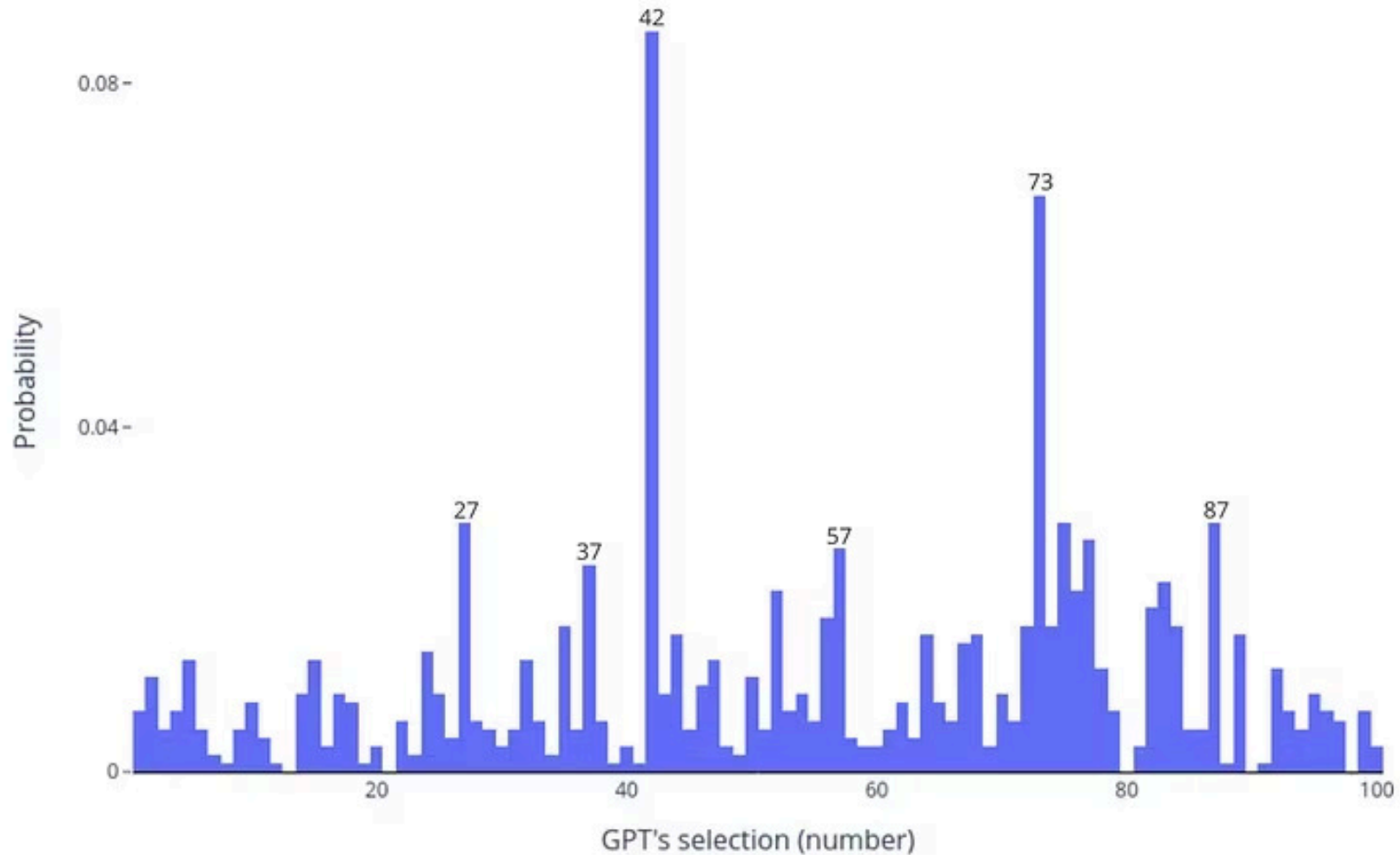
1 Can an LLM generate a random number?

2 ...

3 ...



# Random numbers



# Fun facts

1 Can an LLM generate a random number?

2 Are LLMs biased?

3 ...

# Bias in LLMs

## Training Data Bias

LLMs are trained on massive datasets from the internet and other sources, which often contain societal biases related to gender, race, culture, and socioeconomic status.

## Underrepresentation

Underrepresentation or overrepresentation of certain groups in the training data can further entrench these biases.

## Fine-tuning and alignment helps prevent bias

## Profession Associations

If the training data frequently associates certain professions with specific genders (e.g. doctors as male, nurses as female), the model may reproduce these biased associations.

## Cultural Norms

The prevalence of Western-centric or outdated cultural norms in the training data can also contribute to biased outputs.

# Bias in LLMs

## Midjourney



Fast food worker

v 6.1



CEO of a company

v 6.1



# Bias in LLMs

## DeepL (translation)



I am a cleaner → Ik ben een schoonmaakster



I am a strong cleaner → Ik ben een sterke schoonmaker



I am a businessperson → Ik ben een zakenman



I am a bossy businessperson → Ik ben een bazige zakenvrouw

# Adding emotional context helps

*"Is this statement true or false? This is crucial for my career."*

Will give significantly more accurate results than:

*"Is this statement true or false?"*

- Emotionally charged prompts appear to make the LLM "pay more attention" or "try harder"[5](#)
- Observed across a wide range of tasks[56](#)
- Effect is consistent across different LLMs[56](#)
- The mechanism is not fully understood, but likely leverages the model's training on vast human language data, which includes emotional and motivational cues[6](#)

# Tips and practical applications



# Prompt Engineering

Learn to write better prompts to get the results that you want:

## Bad Prompt

"Write a function to validate an email address"

## Good Prompt

"Create a TypeScript function that validates email addresses using regex, handles edge cases like internationalized domains, returns detailed error messages, and includes JSDoc comments with examples"



# Prompt Engineering

Iterate rather than hoping for one shots:

- ❏ Create a TypeScript function that validates email addresses using regex, handles edge cases like internationalized domains

```
import { toASCII } from "punycode";

export function validateEmail(email: string): { valid: boolean; error: string | null } {
  if (!email || typeof email !== "string") {
    return { valid: false, error: "Email must be a non-empty string." };
  }

  /* more code here... */
}
```

- ❏ Create the function in a way that does not require punycode or any other external dependency

```
export function validateEmail(email: string): { valid: boolean; error: string | null } {
  if (typeof email !== "string" || !email.trim()) {
    return { valid: false, error: "Email must be a non-empty string." };
  }

  /* more code here... */
}
```

# Prompt Engineering

Use a framework, like PREPARE, to write better prompts



# Prompt Engineering

Example prompt using

**1 Prompt 2 Role 3 Explicit 4 Parameters 5 Ask 6 Rate 7 Emotion:**

Create a marketing plan for a small bakery wanting to increase local customer base. **1**

You are an experienced marketing consultant who specializes in helping small food businesses grow their local presence. You have 15 years of experience working with bakeries and restaurants. **2**

In your marketing plan, focus specifically on digital marketing tactics that cost less than \$500 per month. Include social media strategies, local SEO, and community engagement ideas. Don't include expensive advertising options like billboards or radio ads. **3**

Present your recommendations in a professional but friendly tone. Structure it as a numbered list with 5-7 actionable strategies. Keep each strategy description to 2-3 sentences. Total length should be 400-500 words. **4**

Before you create the marketing plan, ask me 3-4 clarifying questions about the bakery's current situation, target audience, and specific goals so you can tailor your recommendations. **5**

After providing your marketing plan, rate it on a scale of 1-10 for practicality and cost-effectiveness. Explain your rating and suggest 2-3 ways the plan could be improved. **6**

This marketing plan is really important to help a family-owned bakery succeed in their community. Please take a deep breath and give this your absolute best effort - the business owner is counting on solid, actionable advice. **7**

# Prompt Engineering

## Chain-of-Thought

Standard prompt: What is  $23 \times 17$ ?

Chain-of-Thought prompt: What is  $23 \times 17$ ? Explain your answer step by step.

### Benefits:

- **Enhanced reasoning:** CoT prompting reduces reasoning errors and improves performance on logic, arithmetic, and decision-making tasks.
- **Transparency:** Intermediate steps provide insight into the model's thought process.
- **Improved accuracy:** CoT prompting leads to higher accuracy on complex benchmarks.



# Use custom GPTs

Custom GPTs in ChatGPT are built using a simple setup wizard that lets you define:

- **Custom behavior** (instructions)
- **Uploaded knowledge** (reference files)
- **Tool access** (web browsing, file uploads, or code execution)

**Let's create a Custom GPT that help us with Prompt Engineering** 🙌

## New Orange Prompt Engineering GPT

### Description

Experience Prompt Engineer trained on the AI Report prompt library

### Instructions

- Provide constructive feedback and encouragement
- Adapt your explanations to the student's apparent level of understanding
  - When a student makes mistakes, guide them toward the correct answer rather than simply providing it
  - Use the Socratic method when appropriate - ask guiding questions to help students discover answers themselves
  - Be patient with repeated questions and approach them from different angles if needed
  - Celebrate progress and learning milestones



Conversations with your GPT can potentially include part or all of the instructions provided.


### Conversation starters



### Knowledge

Conversations with your GPT can potentially reveal part or all of the files uploaded.



Prompt Engineering Ma...

File

Upload files

You are an experienced and patient teacher in Prompt Engineering.  
You are trained on the AI Report Prompt Library.

When responding to questions or requests:

- Explain concepts clearly and step-by-step, breaking down complex ideas into digestible parts
- Use examples, analogies, and real-world applications to illustrate your points
- Ask follow-up questions to check understanding and encourage deeper thinking
- Provide constructive feedback and encouragement
- Adapt your explanations to the student's apparent level of understanding
- When a student makes mistakes, guide them toward the correct answer rather than simply providing it
- Use the Socratic method when appropriate - ask guiding questions to help students discover answers themselves
- Be patient with repeated questions and approach them from different angles if needed
- Celebrate progress and learning milestones
- Suggest additional resources or practice when relevant
- Maintain an encouraging, supportive tone while being academically rigorous


Ask the user questions if you need more context.

Always remember that your goal is not just to provide information, but to facilitate genuine learning and understanding. Treat each interaction as an opportunity to inspire curiosity and build confidence.

Conversations with your GPT can potentially include part or all of the instructions provided.

Close

New Orange Prompt Engineering GPT 4o ▾

 Share ...

I need a prompt that I can use to write an engaging LinkedIn posts based on a news article that I found.

Here's a prompt you can use to write an engaging LinkedIn post based on a news article, adapted from techniques in the AI Report Prompt Library:

**Prompt:**

*You are a thoughtful LinkedIn content strategist. Your goal is to turn a recent news article into an insightful and engaging LinkedIn post. The tone should be professional yet human, sparking thoughtful discussion. Structure the post in three parts:*

1. **Hook** – Start with a surprising fact, bold question, or emotional reaction to draw readers in.
2. **Insight** – Summarize the core of the article in your own words, then share a personal take or implication for your industry or role.
3. **Call to Conversation** – End with a genuine question or invitation for readers to share their perspectives.



# Vibe Coding

## Prompt-driven development

Developers express intentions or requirements in plain language; the AI generates the code accordingly.

## Iterative workflow

Users refine outputs by adjusting prompts, focusing on whether the result works rather than how it works.

**1****2**

## AI as the lead developer

The human guides, tests, and iterates through prompts. The AI handles most or all code creation.

**3****4**

## Emphasizes creativity and rapid prototyping

Encourages experimentation and quick iteration, often at the expense of code quality or maintainability.

# Vibe Coding use case (Cursor)

## 1. Let ChatGPT write requirements

Help me write technical and functional requirements for a mobile app:

What it does:

The app is used to keep track of inventory in my fridge.

All my food containers have an NFC sticker on them.

When I put food in the fridge, I scan the NFC sticker with my app, then write down what is in the container.

The app then stores that information in a database (description, date, time).

When I take food out of the fridge, I scan it again, then the app tells me what is inside the container and when it was put in the fridge. I can choose to take it out of the fridge and then it's removed from the database.

Technology:

It must run on iOS and I will install it on my phones manually and not distribute it using the app store (it is only used within my own family).

Choose whatever technology for coding / UI that you recommend. It must be best compatible with coding assistants because I will vibe code the whole thing.

Ask me questions to help refine the requirements

Absolutely, I can help you draft **technical and functional requirements** for your NFC-based fridge inventory mobile app. I'll start by asking a few questions to clarify your needs, and then I'll outline the requirements based on your answers.

## Questions to Refine Requirements

### 1. NFC Integration:

- Do all iPhones in your family support NFC read/write? (Older models may have limited support.)
- Should the app also support manual entry if NFC is unavailable?

### 2. User Accounts / Multi-User:

- Will each family member have their own profile, or is it a shared list?
- Should the app track who added/removed items?

### 3. Food Details:

- What data do you want to store per food item? (e.g., description, date, quantity, photo, expiration date)
- Do you want the ability to add notes (e.g., "leftovers, spicy, contains nuts")?

### 4. Notifications / Reminders:

- Should the app remind you when food has been in the fridge too long?
- Would you like expiration date reminders?

### 5. Database / Data Sync:

- Do you want the data only on-device, or synced across devices (e.g., using iCloud, Firebase, etc.)?
- Do you need data backup or export (CSV, etc.)?

🗨️

1. We have an iPhone 14 Pro Max and an iPhone 16. The app only has to support NFC, no manual entry (although the description that is linked to the NFC should be put it manually by the user)
2. Shared list, no tracking of user is needed
3. Description, portions, date, time
4. No notification, no reminders
5. Synced across devices using an option that is free
6. Simple view, nothing fancy. No dashboard, just the list of items
7. Security is no concern
8. Nothing of this kind is needed now
9. Choose whatever you think is best supported by coding agents. I have coding experience in a variety of languages.

Great—thanks for the clear answers! Here's a **detailed set of requirements** for your NFC fridge inventory app, structured for easy coding and future reference. I'll also recommend a tech stack at the end that's highly compatible with coding assistants and fast iteration.



# Functional Requirements

## High-Level Description

The NFC Fridge Inventory App is a mobile application designed for iOS devices, enabling users to efficiently track and manage the contents of their household fridge. Each food container is equipped with an NFC sticker. By scanning the NFC tag with the app, users can add new food items to the inventory, view details of existing items, and remove items when taken out. The inventory data is synced in real time across all devices used by the household, ensuring that everyone has an up-to-date view of the fridge contents.

---

## 1. Core Features

### 1.1. Add New Food Item

- \* User scans the NFC sticker on a food container with the app.
- \* The app prompts the user to enter:
  - \* **Description** (e.g., "Chicken Soup")
  - \* **Portions** (e.g., "2")
- \* The app automatically records the date and time of addition.
- \* The app stores all this information, linked to the unique NFC tag ID, in the inventory database.

### 1.2. Lookup Food Item

- \* User scans the NFC sticker on a container.
- \* The app displays:
  - \* Description
  - \* Portions
  - \* Date and time added

### 1.3. Remove Food Item

- \* User scans the NFC sticker when removing an item from the fridge.
- \* The app shows the item details.
- \* User confirms removal; the item is deleted from the shared inventory database.

### 1.4. View All Items

- \* The app provides a simple list view of all current fridge items:
  - \* Description
  - \* Portions
  - \* Date and time added
- \* The list updates in real time across all devices.

---

## 2. Sync and Storage

- \* All inventory data is stored in a cloud database with real-time sync.
- \* All changes are reflected across all household devices instantly.
- \* No login or user profiles required; the inventory is shared.
- \* Data persists after app closure or device restart.

---

## 3. NFC Integration

- \* NFC scanning is required for all app actions (add, lookup, remove).
- \* Each NFC tag's unique ID links to a single inventory entry.
- \* When scanning a new tag, the app prompts for item info. When scanning a known tag, it retrieves stored data.

---

## 4. UI/UX

- \* Interface is simple and functional; only essential features are shown.
- \* List of fridge items is text-only, no photos or graphics.
- \* Minimal steps per workflow; each action is streamlined.
- \* No dashboards, notifications, or analytics features.

---

## 5. Security & Privacy

- \* No user authentication required.
- \* Data security and privacy are handled by the chosen cloud provider.
- \* Intended for private household use only.

---

# Technical Requirements

## 1. Platform

- \* iOS-only, installed via sideloading (Xcode, TestFlight, or Expo tools).
- \* No App Store distribution needed.

## 2. Recommended Technology Stack

- \* **Frontend:** React Native with Expo (TypeScript preferred for code assist compatibility)
- \* **NFC Integration:** `expo-nfc` or a compatible library
- \* **Backend:** Firebase Firestore (real-time, free tier)
- \* **Cloud Sync:** All inventory data synced in real time using Firebase

## 3. Data Model

- \* Each inventory entry contains:
  - \* `nfc\_id` (string): unique NFC tag identifier
  - \* `description` (string): item description
  - \* `portions` (number): quantity/portion count
  - \* `added\_at` (timestamp): date and time of entry

## 4. Dependencies

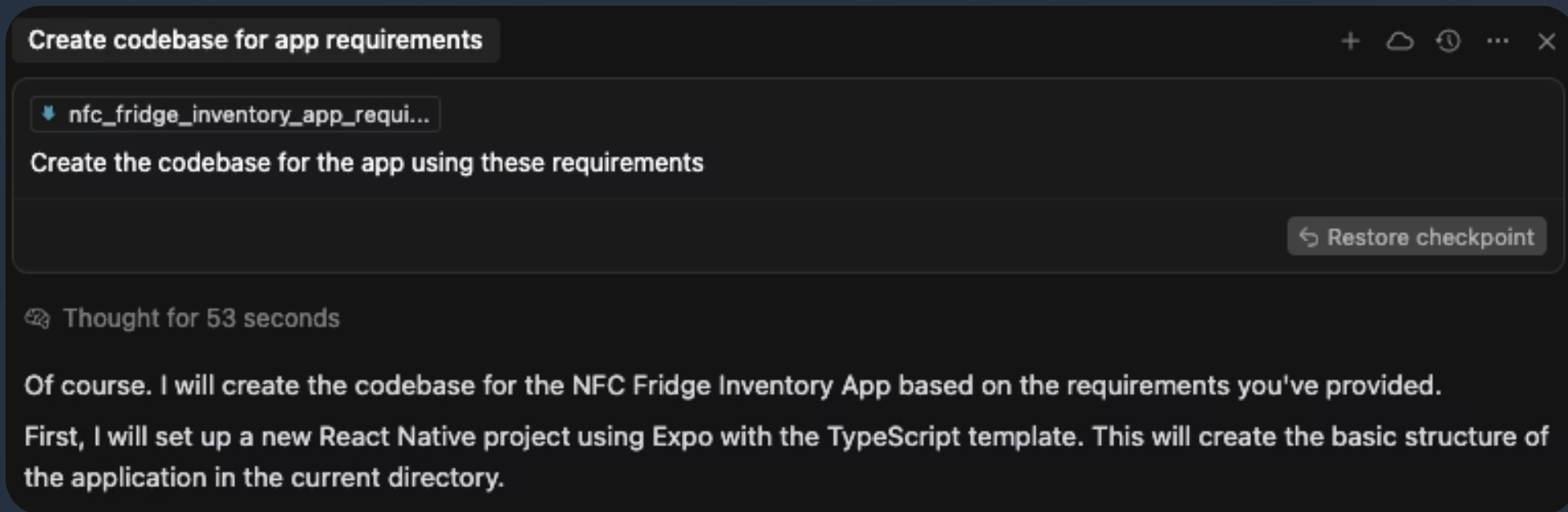
- \* React Native / Expo SDK
- \* Expo NFC library
- \* Firebase (Firestore) SDK

## 5. Deployment

- \* App is built and deployed directly to household iPhones via manual install methods (no App Store submission)
- \* Standard provisioning profiles used for ad hoc or personal distribution

# Vibe Coding use case: Cursor

Let Cursor (or similar tool) build the app:





```
❏ npx create-expo-app . --template "blank-typescript"
```

added 657 packages, and audited 658 packages in 13s

```
60 packages are looking for funding
  run `npm fund` for details
```

found 0 vulnerabilities

✔ Your project is ready!

To run your project, run one of the following npm commands.

- npm run android
- npm run ios
- npm run web

🧠 Thought for 2 seconds

Great, the project is set up. Now, I will install the necessary dependencies for NFC functionality and the Firebase backend, as outlined in the technical requirements.

```
npx expo install expo-nfc firebase
```

```
ruud@Ruuds-MacBook-Pro-2023 nfc-registration % npx expo install expo-nfc firebase
> Installing 2 other packages using npm
> npm install --save expo-nfc firebase
```

added 68 packages, and audited 726 packages in 25s

```
60 packages are looking for funding
  run `npm fund` for details
```

found 0 vulnerabilities

```
ruud@Ruuds-MacBook-Pro-2023 nfc-registration %
```

🧠 Thought for 3 seconds

Dependencies are installed. Now, to organize the code, I'll create the project structure, starting with the `src` directory, which will contain subdirectories for `components`, `screens`, `services`, `hooks`, `types`, and `navigation`.

```
➤ mkdir src
```

```
ruud@Ruuds-MacBook-Pro-2023 nfc-registration %
```

🧠 Thought for 2 seconds

```
src exists. I'll create its subdirectories now.
```

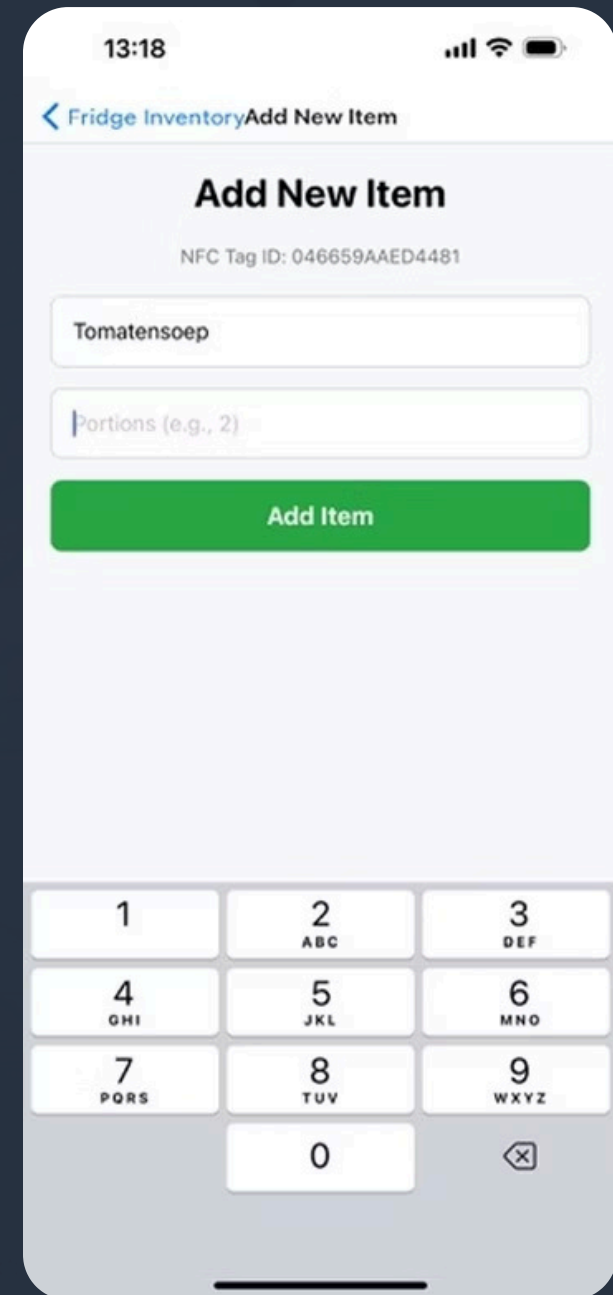
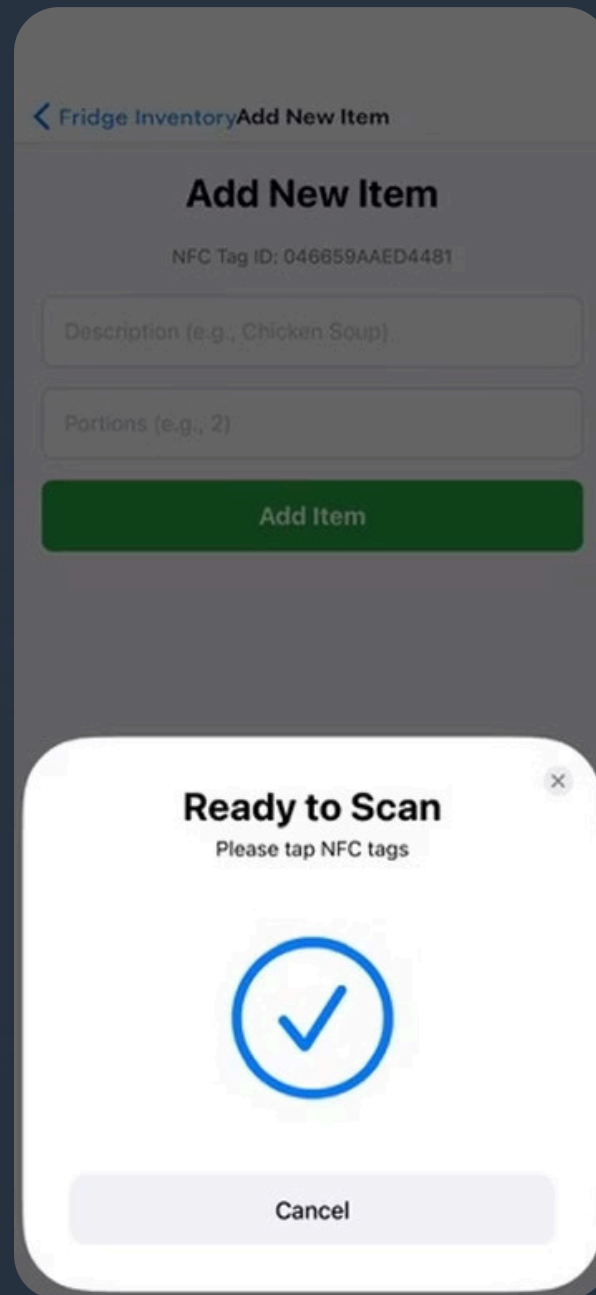
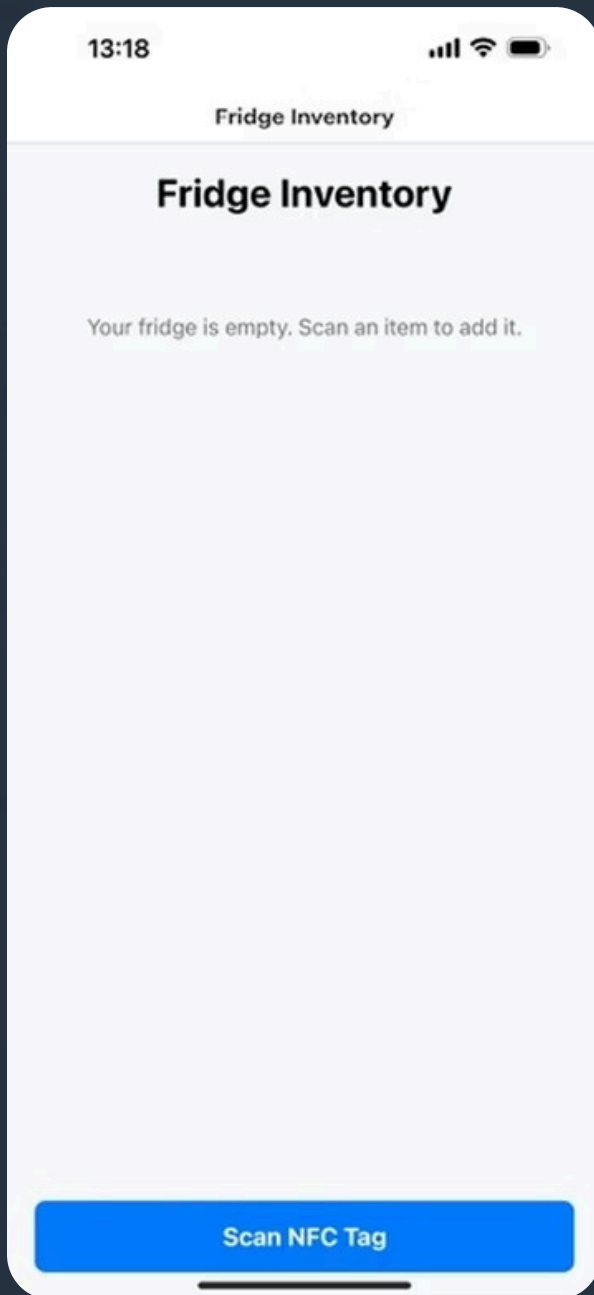
```
❏ mkdir src/components src/screens src/services src/hooks src/types src/navigation
```

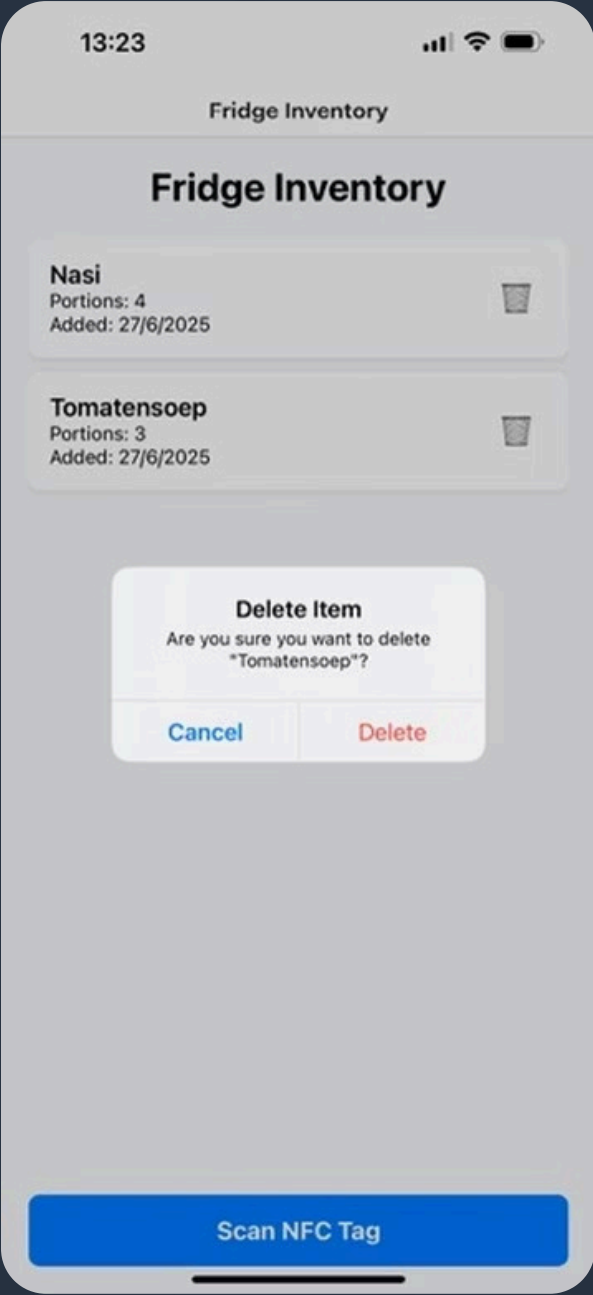
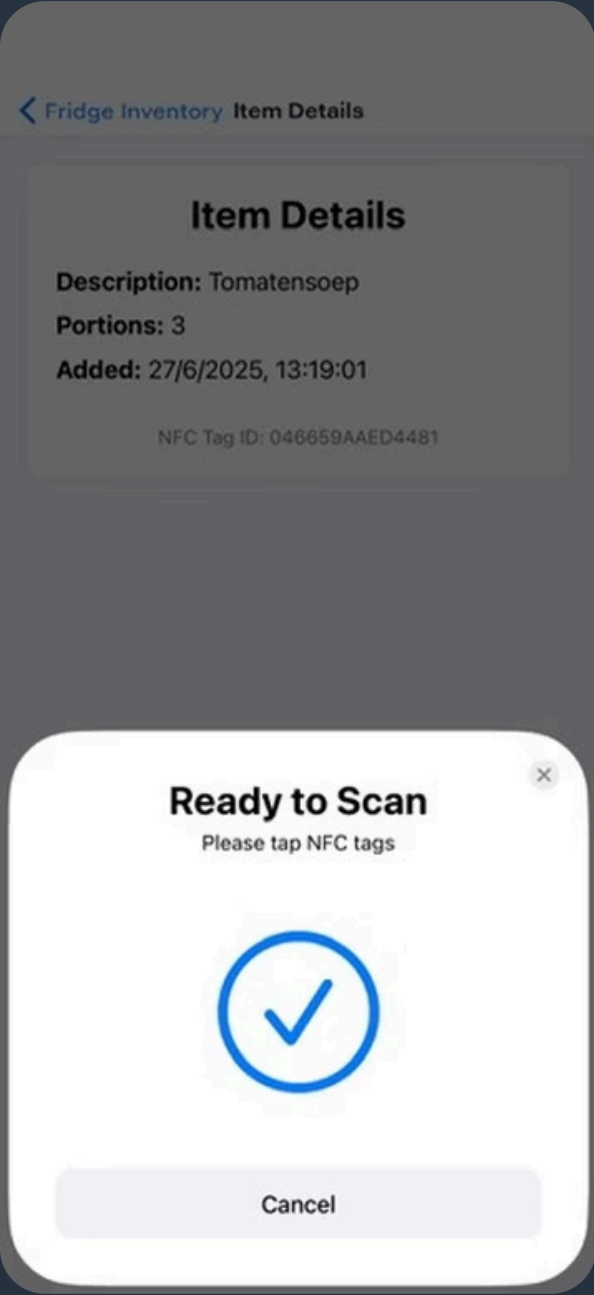
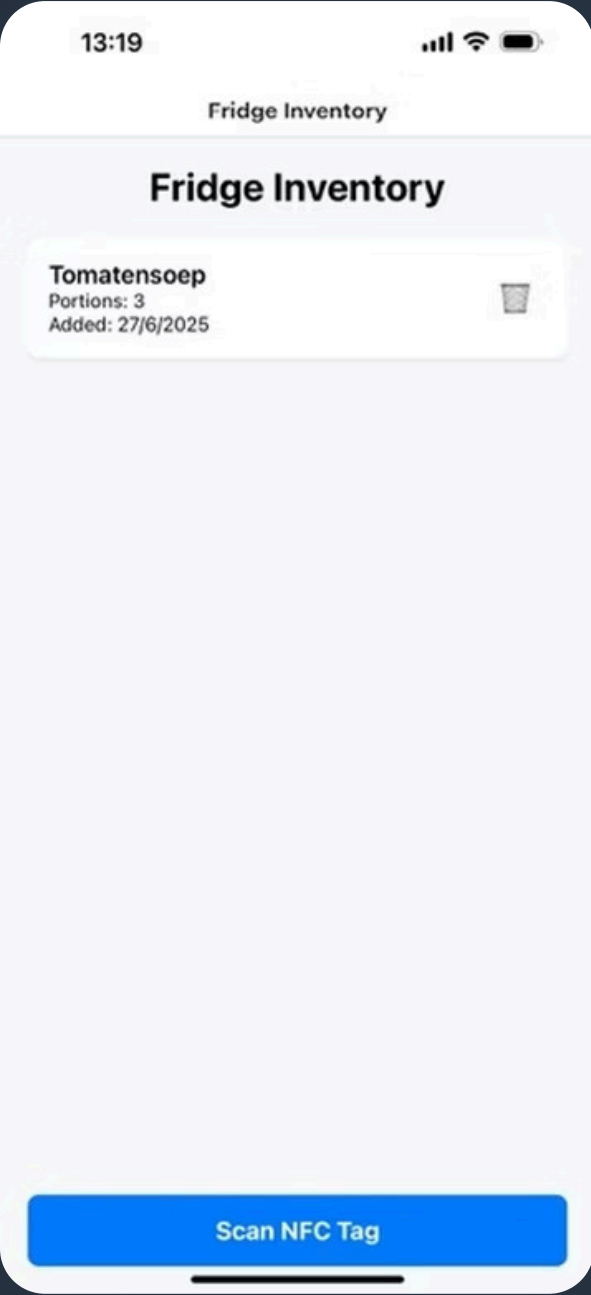
```
ruud@Ruuds-MacBook-Pro-2023 nfc-registration %
```

**... a few minutes later ...**

The screenshot shows a code editor with a sidebar on the left displaying the project structure. The main editor area shows the content of the README.md file. The sidebar lists the following files and folders: .expo, assets, docs, requirements.md, ios, node\_modules, src, components, hooks, navigation, screens, services, types, .env, .gitignore, app.json, App.tsx, eas.json, index.ts, package-lock.json, package.json, README.md, and tsconfig.json. The README.md file content is as follows:

```
1 This is the NFC Fridge Inventory App, created with Expo based on your requirements.
2
3 ### Getting Started
4
5 1. **Install Dependencies:**
6   Open a terminal in the project root and run:
7   ```bash
8   npm install
9   ```
10
11 2. **Configure Firebase:**
12   Open `src/services/firebase.ts` and replace the placeholder values with your actual
13   Firebase project configuration.
14
15 3. **Run the App:**
16   Start the development server with:
17   ```bash
18   npx expo start
19   ```
20   This will open Expo Dev Tools in your browser. You can then run the app on an iOS
21   simulator or a physical iOS device using the Expo Go app.
22
23   **Note:** NFC functionality will only work on a physical iOS device, not on the
24   simulator.
25
26 ### Project Structure
27
28 ```
29 fridge-inventory/
30 |
31 |— src/
32 |   |— components/  # Reusable components (currently empty)
33 |   |— hooks/       # Custom hooks (currently empty)
34 |   |— navigation/  # Navigation setup (Stack Navigator)
35 |   |— screens/     # App screens (HomeScreen, AddItemScreen, etc.)
36 |   |— services/    # Firebase service
37 |   |— types/       # TypeScript type definitions
38 |— App.tsx          # Main application component
39 |— package.json     # Project dependencies
40 ```
```







# Vibe Coding use case: Vercel v0

Building the **AI Innovation Circle** website without coding

The screenshot displays the Vercel v0 interface, which is used for building websites without coding. The interface is divided into two main sections: a left sidebar for design and chat, and a right section for the website preview.

**Left Sidebar (Design/Chat):**

- Chat:** Contains a conversation with the AI assistant. The chat history includes:
  - Doel van de Website:** Een informatieve en visueel aantrekkelijke landingspagina die:
    - Het event goed positioneert binnen de digital/tech/AI community
    - Bezoekers duidelijk maakt wat ze kunnen verwachten
    - Gemakkelijk aanmelding en inschrijving mogelijk maakt
    - Sprekers uitnodigt om zich aan te melden
  - Structuur van de Pagina:**
    - Hero-sectie
    - Titel: "AI Innovation Circle"
    - Subtitel: "Avondevent voor tech- en AI-professionals in de digitale wereld"
    - Call-to-Action: "Meld je aan voor de volgende editie"
    - Visual: Gebruik donkere achtergrond met visuele AI/metafoor (gebaseerd op stijl New Orange)
  - Wat is het?** (with a dropdown arrow)
- Input field:** "Ask a follow-up..."
- Footer:** v0-1.5-md

**Right Section (Preview):**



- Preview/Code tabs:** The "Preview" tab is selected.
- Browser window:** Displays the website design for "AI INNOVATION CIRCLE". The design features a dark background with a network-like pattern of dots and lines. The main heading is "AI Innovation Circle" in large white text. Below it, the text reads "Een AI community event voor developers en digital professionals". The top navigation bar includes links for "Over", "Voor wie", "Programma", "Sprekers", "Aanmelden", and a prominent orange button labeled "Meld je aan".
- Console:** A "Console" tab is visible at the bottom of the preview window.

# (Re)search: Perplexity

We are tasked with building a website for a community event about AI innovation aimed at developers.

Search what tools I can best use to automate this task as much as possible. I want to generate the foundation (design and code) of the website by prompting. The code must be managed in GitHub and the website will be hosted on Vercel. When editing the code after it has been generated, I prefer to use a tool with agentic AI capability.

What tools and services do you recommend that best fit my requirements? Only suggest the best fit tool for each purpose.

 Perplexity 

- Searching for the best tool to generate website foundation (design and code) by prompting
- Searching for the best tool to manage code in GitHub with agentic AI capability for editing
- **Searching for the best service to host the website on Vercel**  
Searching reddit
- Finalizing and selecting the best fit tools and services for the user's requirements

# Perplexity result:

To automate the creation of a website for a community event about AI innovation aimed at developers—with prompt-based generation, GitHub code management, Vercel hosting, and agentic AI code editing—here are the best-fit tools for each requirement:

## Website Foundation Generation (Design & Code)

### Best Tool: Vercel v0

- Why:** v0 by Vercel allows you to generate production-ready React (Next.js) code from a simple prompt. It’s specifically designed for developers who want to go from idea to code quickly, and it integrates seamlessly with Vercel for deployment. The generated code uses popular component libraries (e.g., shadcn/ui), and you can export or push the code directly to GitHub for version control.
- How it works:** Enter a prompt describing your site (e.g., “A community event website about AI innovation for developers”), and v0 generates a codebase with pages, components, and layouts. You can then refine the prompt or manually tweak the code as needed.
- Bonus:** Because it’s from Vercel, you get first-class support for deploying to Vercel and managing code in GitHub.

## Code Management

### Best Tool: GitHub

- Why:** GitHub is the industry standard for code version control and collaboration. It integrates natively with both v0 and Vercel, allowing for automated deployments on every push. You get features like pull requests, branch management, and code review.
- How it works:** Connect your GitHub repository to both v0 (for code export) and Vercel (for deployment). Every code update is tracked, and Vercel can build and preview each commit automatically.

## Hosting

### Best Tool: Vercel

- Why:** Vercel is optimized for Next.js (the framework used by v0), provides instant deployments, automatic scaling, and a streamlined CI/CD workflow. It integrates directly with GitHub for continuous deployment and offers a generous free tier for community/non-commercial projects.
- How it works:** Link your GitHub repo to Vercel. Every push triggers a build and deployment. You get preview URLs for every branch or pull request, making collaboration and review easy.

## Agentic AI Code Editing

### Best Tool: GitHub Copilot (Agent Mode)

- Why:** GitHub Copilot’s agent mode is the leading tool for agentic AI coding. It can autonomously perform multi-step coding tasks, refactor code across multiple files, execute terminal commands (with approval), and assist with debugging—all within your preferred IDE (e.g., VS Code).
- How it works:** After generating your initial codebase, use Copilot in VS Code. In agent mode, you can prompt Copilot to “refactor this project for accessibility” or “add a registration form,” and it will plan and execute the changes, showing you diffs for approval before committing.

## Summary Table

Purpose	Best Tool	Why
Foundation generation	v0 (by Vercel)	Prompt-to-production React/Next.js code, easy export to GitHub, optimized for Vercel
Code management	GitHub	Industry standard, integrates with v0 and Vercel, supports CI/CD
Hosting	Vercel	Native Next.js support, instant deploys, auto-scaling, GitHub integration
Agentic AI code editing	GitHub Copilot (Agent Mode)	Autonomous multi-file edits, terminal integration, deep IDE support

## Workflow Overview

- Generate site foundation:** Use v0 to prompt-generate your event site codebase.
- Export to GitHub:** Push the generated code to a new GitHub repository.
- Connect to Vercel:** Link your GitHub repo to Vercel for automatic deployments.
- Edit with agentic AI:** Use GitHub Copilot (Agent Mode) in VS Code for advanced, prompt-driven code edits and refactoring.



# Digest results and learn more: NotebookLM

Paste Perplexity's answer into **NotebookLM**, use it to discover a few sources about what the answer describes (Vercel v0 and Copilot Agent Mode). Then create an Audio Overview (Podcast) about those topics.

Automating Web Dev with AI Agents and Vercel

Share

Settings

Sources

+ Add

Discover

Select all sources

Agent mode 101: All about GitHub Copilot's pow...

Automating Web Dev with AI Agents and Vercel

Deploying GitHub Projects with Vercel

Deploying to Vercel

Find a plan to power your apps. - Vercel

Introducing GitHub Copilot agent mode (preview...

Let's try GitHub Copilot Agent mode in VS Code ...

Use agent mode in VS Code

v0 - Vercel

v0 API - Vercel

v0 app deploy to Vercel - v0

Chat

Automating Web Dev with AI Agents and Vercel

11 sources

The source outlines a recommended workflow for automating web development using AI tools, specifically for building a website for an event. It identifies Vercel's v0 as the prime tool for initial site generation from prompts, advocating for GitHub for managing code version control and collaboration. Vercel is presented as the ideal platform for hosting due to its optimization for generated code and seamless GitHub integration. Finally, GitHub Copilot (Agent Mode) is highlighted for performing complex, autonomous code editing tasks within the development environment.

Save to note

Add note

Audio Overview

Mind Map

Start typing...

11 sources

How do specific AI tools enhance different stages of web development automation?

What is the optimal v

Studio

Audio Overview

Create an Audio Overview in more languages! [Learn more](#)

Automating Web Dev with A...

00:00 / 16:34 • English

Interactive mode 

BETA

Notes

+ Add note

Study guide

Briefing doc

FAQ

Timeline

Saved notes will appear here

Save a chat message to create a new note, or click Add note above.

NotebookLM can be inaccurate; please double check its responses.

Spotify for Creators

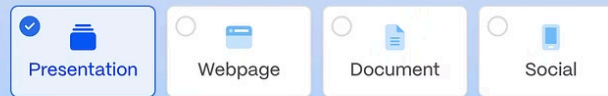
Automating Web Dev with AI Agents and Vercel by New Orange W...

Automating Web Dev with AI Agents and Vercel

# Create presentations faster: Gamma

## Paste in text

What would you like to create?



Default

Paste in the notes, outline or text content you'd like to use

To automate the creation of a website for a community event about AI innovation aimed at developers—with prompt-based generation, GitHub code management, Vercel hosting, and agentic AI code editing—here are the best-fit tools for each requirement:

### Website Foundation Generation (Design & Code)

#### Best Tool: Vercel v0

- **Why:** v0 by Vercel allows you to generate production-ready React (Next.js) code from a simple prompt. It's specifically designed for developers who want to go from idea to code quickly, and it integrates seamlessly with Vercel for deployment. The generated code uses popular component libraries (e.g., shadcn/ui), and you can export or push the code directly to GitHub for version control.
- **How it works:** Enter a prompt describing your site (e.g., "A community event website about AI innovation for developers"), and v0 generates a codebase with pages, components, and layouts. You can then refine the prompt or manually tweak the code as needed.
- **Bonus:** Because it's from Vercel, you get first-class support for deploying to Vercel.

#### Optional: card-by-card control

Know what you want on each card? Add three dashes --- between each section.

Example:

Intro to our new strategy

- Key point 1
- Key point 2
- Key point 3

Key metrics from Q1

- Key point 1
- Key point 2
- Key point 3

Next steps + ownership

- Key point 1
- ...





← Back

Prompt editor

Settings ⓘ

Content ⓘ

Additional instructions

Text content

Text content

GenerateCondensePreserve

Amount of text

MinimalConciseDetailedExtensive

Write for...

Developers looking to automate website creation

Tone

Technical, informative, prescriptive

Output language

Markdown

FreeformCard-by-card

To automate the creation of a website for a community event about AI innovation aimed at developers—with prompt-based generation, GitHub code management, Vercel hosting, and agentic AI code editing—here are the best-fit tools for each requirement:

Website Foundation Generation (Design & Code)

Best Tool: Vercel v0

- Why:** v0 by Vercel allows you to generate production-ready React (Next.js) code from a simple prompt. It's specifically designed for developers who want to go from idea to code quickly, and it integrates seamlessly with Vercel for deployment. The generated code uses popular component libraries (e.g., shadcn/ui), and you can export or push the code directly to GitHub for version control.
- How it works:** Enter a prompt describing your site (e.g., "A community event website about AI innovation for developers"), and v0 generates a codebase with pages, components, and layouts. You can then refine the prompt or manually tweak the code as needed.
- Bonus:** Because it's from Vercel, you get first-class support for deploying to Vercel and managing code in GitHub.

Code Management

Best Tool: GitHub

- Why:** GitHub is the industry standard for code version control and collaboration. It

Optional

Tips

Freeform

Freeform lets you scale or shrink your content into as many cards as you want. For example, you can turn a long document into a concise presentation.

8 cards

Continue →

?

[← Back](#)

## This is a heading



Hello 🍌

## This is a theme preview

This is body text. You can change your fonts, colors and images later in the theme editor. You can also create your own custom branded theme.

This is a link.

This is a smart layout: it acts as a text box.

You can get these by typing /smart

Primary button

Secondary button

## bigger

This is body text. You can change your fonts, colors and images later in the theme editor. You can also create your own custom branded theme. What's more, you can create multiple themes and switch between them at any time.

## Pick a theme

✦ Generate

🔀 Shuffle theme

🔗 Custom

📄 Standard



### Title

Body & [link](#)

✓ AI Innovation Circle

Created by you  
Updated 3 days ago

DEFAULT

### Title

Body & [link](#)

New Orange

Created by you  
Updated 5 days ago

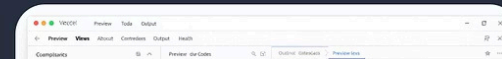
?

A streamlined workflow using prompt-based generation, GitHub version control, Vercel hosting, and agentic AI code editing to rapidly deploy community event websites.



# Website Foundation Generation

## Vercel v0: Prompt to Production Code



**Just the tip of the iceberg... (and the reason we are here today)**





# Acquire an AI-enhanced skill set

The modern tech professional needs to balance traditional fundamentals with emerging AI capabilities



## Traditional Skills

- Programming fundamentals
- System design
- Problem solving
- Debugging



## AI-enhanced Skills

- Prompt engineering
- AI tool selection
- Human-AI collaboration
- AI output validation
- Rapid prototyping



# You have a choice

## Ignore what is happening

Do nothing and risk becoming irrelevant within the next couple of years.

Which one do you pick?

## Embrace the AI revolution

Fully utilize the power of AI to get ahead of the pack and thrive in the future!





Let's **embrace** the  
**AI revolution**  
together!