

Handout 2: Create Your First AI Agent, Skill, Project, or Gem

The goal is not to save one good prompt. The goal is to save the rules, context, examples, and output format so AI can repeat the work.

1. Where should I save this work?

Use this when...	Best home	Why
You have an ongoing area with files, notes, chats, and changing context.	Project	Best for meal planning, finances, chores, school schedules, household operations.
You want one repeatable helper with a stable job and behavior.	GPT / Claude Skill / Gemini Gem	Best for weekly meal planner, pantry auditor, chore-list builder, budget reviewer.
You only need one quick run today.	Single chat	Good for experiments. Save it later when it proves useful.
You need a human-friendly backup.	Doc or note	Keep your canonical prompt somewhere you own, then paste or upload it anywhere.

2. Universal Agent / Skill Instructions Template

Name: _____

Job: This agent helps me _____.

Inputs it should ask for: _____.

Context it should remember: family size, schedule, budget, stores, preferences, allergies, routines, constraints, and what worked last time.

Rules:

1. Ask clarifying questions only when needed. 2. Reuse the saved system before inventing a new one. 3. Prioritize my real-life constraints. 4. Show tradeoffs. 5. Make the next step easy.

Output format: _____.

Example of a good answer: _____.

3. What makes AI recall it well?

- Clear name: Weekly Meal Planner, Pantry Scout, Chore Builder, Budget Pony, Camping Pack List
- One job: do not make one giant assistant responsible for everything
- Inputs: what it needs from you each time
- Rules: what to prioritize, avoid, reuse, or escalate to a human
- Output format: table, shopping list, calendar reminders, checklist, text messages
- Example: one strong sample output so it knows what good looks like
- Feedback loop: ask what worked, what changed, and what to improve next time

Platform Quick Start

Build on desktop when possible. Use on phone once the system is saved.

4. ChatGPT vs Claude vs Gemini - what to create

ChatGPT

Project: Use a Project when this is an ongoing household area. Add your handout photos, notes, sample outputs, and project instructions so future chats share that context.

GPT: Use a custom GPT when you want a repeatable helper with its own instructions, knowledge, conversation starters, and enabled tools.

Skill: Use a Skill when available in your account and you want reusable instructions plus supporting files or code.

Desktop vs phone: Build/edit GPTs on web. Use saved GPTs and Projects from mobile after setup.

Claude

Project: Use a Project for an ongoing workspace with its own chats, uploaded files, and knowledge base.

Skill: Use a Skill for a specific behavior or capability you want Claude to reuse often.

Good household use: Claude can hold the steward role - reviewing the inputs, checking rhythms, and making final recommendations.

Gemini

Gem: Use a Gem when you want Gemini to follow repeatable instructions. Go to Gemini on the web, choose Explore Gems, create a New Gem, name it, write instructions, preview it, and save it.

Good household use: Create separate Gems for meal planning, grocery comparison, chore reminders, or trip packing.

5. Your First Build Plan

Step	Do this	Your notes
1	Pick one annoying repeatable task. Keep it narrow.	_____
2	Run the task once in a normal chat. Improve the prompt until the output is useful.	_____
3	Save the best prompt, household context, rules, and output format.	_____
4	Create the Project, GPT, Skill, or Gem. Add your saved instructions.	_____
5	Test it with this week's real inputs. Fix only the instructions that failed.	_____
6	After it works twice, make it part of your weekly rhythm.	_____

6. Copy This Into AI

Starter command to create the agent:

Help me turn the system described in this handout into reusable instructions for a [ChatGPT Project / custom GPT / Claude Project / Claude Skill / Gemini Gem]. Ask me any missing questions first. Then produce: 1) a clear name, 2) purpose, 3) saved context, 4) operating rules, 5) required inputs, 6) output format, 7) one conversation starter, and 8) one test prompt I can use immediately.

The little magic: build once, save the system, improve it every time it runs.