AI Bootcamp Week 2 Company Brain Prompt Pack

Atlas

AI Bootcamp Week 2 Company Brain Prompt Pack - 2026-06-11

How To Use This In The Live Session

1. Create a fresh Obsidian vault.
2. Open Claude Cowork and point it at the fresh vault folder.
3. Paste Prompt 1 to create the starter vault structure and folder READMEs.
4. Download the fake Fathom-style transcript from OneDrive and save it inside the vault.
5. Paste Prompt 2 and point Claude Cowork to the saved transcript file path.
6. Open Graph View and show how people, companies, meetings, projects, workflows, and systems are connected.
7. Paste Prompt 3 to archive the example content while keeping the READMEs and structure.
8. Use Prompt 4 to have Claude interview the student and extract their real workflow.
9. Use Prompt 5 as homework after students pick their real workflow.

Example transcript download: https://netorg7032938-my.sharepoint.com/:t:/g/personal/brian_hatch-capital_com/IQAOeLXITjt2TICkffkuDKKEAUTexnQbHdAxl4nW4YT8dCk

Why This Matters

The company brain is the beginning of every future workflow. A workflow cannot become valuable until the agent has the context needed to understand it.

Students will eventually connect tools, but tool access alone is not understanding. A CRM gives the agent a person/pipeline slice. A Google Drive SOP gives the agent a process slice. A PMS, Slack channel, spreadsheet, or project board gives another slice. Obsidian is where those slices become connected through wiki links.

The goal is not to duplicate the entire business. The goal is to pull the context required for the selected workflow into one linked place:

- people and relationship context from the CRM;
- SOPs, templates, and examples from Drive;
- project and ownership context from project tools;
- source-system notes describing where work happens now;
- human judgment rules and stop signs from the operator;
- output examples and receipts that show what good work looks like.

Recommended Starter Vault Structure

```
AI Bootcamp Company Brain/
  00-Inbox/
    README.md
  10-Meetings/
    README.md
  20-People/
    README.md
  30-Projects/
    README.md
  40-Workflows-and-SOPs/
    README.md
  50-Source-Systems/
    README.md
  60-Content-and-Outputs/
    README.md
  70-Prompts/
    README.md
```

> ```
> 80-Receipts/
>   README.md
> 99-Archive/
>   README.md
> README.md
> ```

This is intentionally generic. Students can rename, add, or delete folders later. The important part is that each folder has a README that tells the agent what belongs there and how to document it.

README Standards To Bake Into The Vault

Prompt 1 should create practical READMEs, not placeholder documents. Each README should include:

- **Purpose:** what this folder is for.
- **What belongs here:** the file types, notes, and business context that should live here.
- **What does not belong here:** guardrails so the vault does not become a junk drawer.
- **Naming convention:** a simple filename pattern.
- **Required sections:** the headings every note in that folder should usually include.
- **Wiki-link rules:** which related notes should be linked.
- **Agent instructions:** how Claude Cowork should update that folder later.
- **Open Questions rule:** uncertain facts go under Open Questions, not into invented conclusions.

Folder-specific README direction:

- **00-Inbox:** raw captures, downloaded files, voice-note transcripts, screenshots, messy notes, temporary inputs waiting to be processed.
- **10-Meetings:** one note per meeting or transcript. Preserve the native transcript, then add metadata, attendees, summary, decisions, action items, extracted entities, and backlinks.
- **20-People:** one note per person. Link company, role, source meeting, related projects, related workflows, action items, and open questions.
- **30-Projects:** one note per project or initiative. Track goal, owner, related people, related meetings, decisions, open questions, next actions, and outputs.
- **40-Workflows-and-SOPs:** one note per workflow/SOP. Track trigger, owner, inputs, source systems, current steps, exceptions, human review

gates, stop signs, outputs, receipts, and missing information.

- **50-Source-Systems:** one note per tool/system. Track what data lives there, what workflow uses it, what the agent may need to read/write/trigger/verify, and access questions for later.
- **60-Content-and-Outputs:** briefs, reports, checklists, drafts, templates, deliverables, and examples of good output.
- **70-Prompts:** reusable prompts, extraction prompts, agent instructions, and workflow-specific prompt snippets.
- **80-Receipts:** proof artifacts: screenshots, sent items, exports, logs, confirmation pages, reviewed files, and evidence that work happened.
- **99-Archive:** old, superseded, or fake training data that should stay searchable but not compete with active business context.

Prompt 1 - Create The Starter Vault Structure

```
You are pointed at my fresh Obsidian vault folder.

Create the starter company-brain folder structure inside this
vault.
Create README.md files inside every folder.
The README files are agent instructions. They should tell future
agents what belongs in each folder, what does not belong there,
how notes should be named, how wiki links should be used, and what
template to follow.

Create this structure:

00-Inbox/
10-Meetings/
20-People/
30-Projects/
40-Workflows-and-SOPs/
50-Source-Systems/
60-Content-and-Outputs/
70-Prompts/
80-Receipts/
99-Archive/

Also create a main README.md at the vault root.

README requirements:
```

- Every README must include Purpose, What Belongs Here, What Does Not Belong Here, Naming Convention, Required Sections, Wiki-Link Rules, Agent Instructions, and Open Questions.
- The main README.md should explain the whole company-brain structure and tell future agents to read folder READMEs before updating the vault.
- 00-Inbox README should say raw captures and downloaded files land there before processing.
- 10-Meetings README should say to preserve native transcripts inside meeting notes, then extract metadata, attendees, summary, decisions, action items, entities, and backlinks.
- 20-People README should say one note per person with company, role, first/source meeting, related projects, related workflows, action items, meeting history, and open questions.
- 30-Projects README should say one note per project with goal, owner, related people, related meetings, workflows, decisions, open questions, next actions, outputs, and receipts.
- 40-Workflows-and-SOPs README should say one note per workflow with trigger, owner, inputs, current steps, source systems, exceptions, human review gates, stop signs, expected output, proof receipts, and missing information.
- 50-Source-Systems README should say one note per tool/system with what data it stores, what workflows use it, what an agent may need to read/write/trigger/verify later, and access questions.
- 60-Content-and-Outputs README should cover briefs, reports, templates, checklists, drafts, and examples of good output.
- 70-Prompts README should cover reusable prompts, extraction prompts, workflow prompts, and agent instructions.
- 80-Receipts README should cover screenshots, exports, logs, sent items, confirmation pages, reviewed outputs, and proof that work happened.
- 99-Archive README should explain how to archive fake training data and old material without deleting it.

Rules:
- Keep the structure simple and beginner-friendly.
- Do not create fake business notes yet.
- Do not invent any real client facts.
- Use placeholders only in the README templates.
- Use Obsidian wiki links like [[Person Name]], [[Project Name]], and [[Workflow Name]] in the templates.

```
- Make the READMEs practical enough that you can read them later
before updating the vault.

After creating the files, summarize the folder structure and tell
me what to do next.
```

Prompt 2 - Ingest The Example Meeting Artifact From A Saved File

```
You are pointed at my Obsidian company-brain vault.

First, read the main README.md and every folder README.md so you
understand the vault rules.

Then read this saved fake Fathom-style meeting transcript file:

[PASTE THE LOCAL FILE PATH HERE]

Your job is to update the vault based on that file.

Important:
- This is EXAMPLE information only.
- Put example notes under the right folders.
- Use Obsidian wiki links everywhere useful.
- Connect people to companies, meetings, projects, workflows,
source systems, and action items.
- Preserve the native transcript inside the meeting note in 10-
Meetings/.
- Create clean extracted notes from the transcript.
- Do not invent facts beyond the transcript.
- If a fact is unclear, add it to an Open Questions section.

Create these files as Markdown:

1. 10-Meetings/[meeting-title].md
   Include metadata, attendees, summary, decisions, action items,
transcript link/backlink, and extracted entities.

2. 20-People/[person-name].md for every person
```

```
   Include:
   - name
   - company
   - role
   - email if available
   - first contact / source meeting
   - related projects
   - related workflows
   - action items
   - notes from this meeting

3. 30-Projects/[project-name].md for every project
   Include:
   - project goal
   - owner
   - related people
   - related company
   - related meetings
   - related workflows
   - decisions
   - open questions
   - next actions

4. 40-Workflows-and-SOPs/[workflow-name].md for each workflow
   Include:
   - workflow goal
   - trigger
   - inputs
   - current steps
   - source systems
   - human review gates
   - stop signs
   - expected output
   - proof receipts
   - missing information

5. 50-Source-Systems/[system-name].md for each tool/system
   Include:
   - what it is used for
   - what data it stores
   - what workflow uses it
```

```
  - known access method if mentioned
  - open questions about access

6. 60-Content-and-Outputs/[output-name].md for any drafts, briefs, reports, or content artifacts mentioned

7. 80-Receipts/[receipt-name].md for any proof or verification artifact mentioned

After creating the notes, give me:
- a list of files created
- a graph explanation showing which notes should connect to each other
- 5 questions I should ask next before turning this into an agent workflow
```

Prompt 3 - Archive Example Data But Keep The Structure

```
The notes in this vault are from a fake training example.

Archive the example information, but keep the company-brain structure and the README instructions.

Do this:
- Move all example meeting, people, project, workflow, source-system, output, and receipt notes into 99-Archive/Example-Training-Data/.
- Keep all folder README.md files in place.
- Keep the main README.md in place.
- Create 99-Archive/Example-Training-Data/README.md explaining that the archived notes are fake training data.
- Create a clean 00-Inbox/README.md reminder that future real transcripts, call notes, screenshots, and source docs can be dropped there for processing.
- Do not delete the example data unless I explicitly ask.

Return:
- what moved
```

```
- what stayed
- what I should do next with real business information
```

Prompt 4 - Interview Me To Extract My Real Workflow

```
You are going to interview me step by step to extract a real
business workflow and make it agent-ready.

My workflow is:

[WORKFLOW NAME]

Do not give me a giant form all at once. Interview me one section
at a time.

Your job is to help me produce a complete workflow note for my
Obsidian company brain.

Interview me in this order:

1. Workflow overview
   - What is the workflow called?
   - What outcome should it produce?
   - Why does it matter?

2. Trigger
   - What starts the workflow?
   - Is the trigger manual, scheduled, event-based, email-based,
form-based, system-based, or something else?
   - What tells us the workflow should start now?

3. Owner and participants
   - Who owns the workflow today?
   - Who else is involved?
   - Which people should have notes in 20-People/?

4. Inputs and source systems
   - What information is needed to start?
```

```
   - What systems does the information live in?
   - Is it in CRM, email, Google Drive, PMS, Monday.com, Slack, spreadsheet, calendar, database, or another tool?
   - What should become a note in 50-Source-Systems/?

5. Current steps
   - Walk me through the process in order.
   - For each step, ask what tool is used, what data is read, what data is written, and what output is created.

6. Decisions and human judgment
   - Where does a human make a judgment call?
   - What rules guide the decision?
   - When should an agent stop and ask instead of acting?

7. Exceptions and edge cases
   - What can go wrong?
   - What happens if a required field is missing?
   - What happens if a name, email, file, reservation, lead, or record already exists?
   - What should the fallback be?
   - Example: if the preferred new email address is already taken, what backup naming rule should the agent use?

8. Outputs
   - What should the workflow create or update?
   - What does good output look like?
   - Do we have examples that should go in 60-Content-and-Outputs/?

9. Review gates
   - Who reviews the output?
   - What must be reviewed before anything is sent, published, changed, or marked complete?

10. Receipts
   - What proves the work happened?
   - Screenshot, sent item, exported file, updated record, confirmation page, log, calendar event, or reviewed output?
   - What should go in 80-Receipts/?
```

```
11. Wiki-link plan
   - Which people, projects, meetings, workflows, source systems, outputs, and receipts should link to each other?

After the interview, produce:
- a Markdown workflow note for 40-Workflows-and-SOPs/
- a list of People notes to create or update
- a list of Project notes to create or update
- a list of Source System notes to create or update
- a list of Output examples needed
- a list of Receipt examples needed
- open questions before this can become an agentic workflow

Do not invent facts. If I do not know something, put it under Open Questions.
```

Prompt 5 - Homework: Gather Context For My Real Workflow

```
I selected this workflow for my first agent-ready workflow:

[WORKFLOW NAME]

Help me identify the context I need to move into my Obsidian company brain before next week's build/tool-test session.

Organize the answer into:

1. People context
   - which people matter
   - what CRM/contact information I need
   - what relationship/history details matter
   - which notes should link to each person

2. Process/SOP context
   - what SOPs, checklists, templates, examples, or Google Drive docs I should bring in
   - what should be summarized into Obsidian
   - what should stay as a linked source document
```

```
3. Project context
   - which projects or client/account records this workflow affects
   - what project status, goals, open questions, or next actions matter

4. Source-system context
   - what systems the workflow touches
   - what information each system stores
   - what the agent may need to read, write, trigger, or verify later
   - what access questions I should bring to Week 3

5. Prompt/context notes
   - what the agent needs to know before doing the work
   - what stop signs and human review gates matter
   - what good output looks like
   - what receipt proves the work happened

6. Wiki-link plan
   - which notes should link together
   - which people connect to which projects
   - which workflows connect to which source systems
   - which outputs connect to which receipts

Return a checklist I can complete before next week.
```

People README Example

Use this as the sample folder README to show the class what “agent instructions inside a vault” means.

```
# People README

This folder stores one note per important person.

## What Belongs Here

- clients
```

```markdown
- prospects
- team members
- vendors
- partners
- recurring collaborators
- stakeholders mentioned in meetings

## What Does Not Belong Here

- one-off names with no action or relationship
- companies without a named person
- raw transcripts
- long project plans

## Person Note Template

---
tags: [person]
status: active
---

# Person Name

## Snapshot

- **Company:** [[Company Name]]
- **Role:**
- **Email:**
- **First contact:** [[Meeting Note]]
- **Related projects:** [[Project Name]]
- **Related workflows:** [[Workflow Name]]

## Context

Short summary of who this person is and why they matter.

## Meeting History

- [[Meeting Note]] - short note about what happened.

## Action Items
```

```
- [ ] Owner -> action item.

## Open Questions

- Question that needs follow-up.

## Agent Instructions

When updating a person note:
- link their company using [[Company Name]]
- link any meeting where they appeared using [[Meeting Title]]
- link any project or workflow they own or influence
- keep facts short and source-backed
- do not invent contact details or decisions
- add uncertain items under Open Questions
```

Teaching Notes For Brian

The point of this exercise is not that everyone should have the same vault forever.

The point is:

- a vault is just a structured folder of Markdown files;
- agents can create the first structure;
- agents can turn a messy meeting into connected business memory;
- wiki links create a graph of people, meetings, projects, workflows, and systems;
- that graph is useful because future agents can retrieve the right context instead of starting from zero;
- example data should be archived before students use the vault for real business work.

Best line:

> "You are not documenting because documentation is cute. You are creating the memory layer your future agent will use to understand the business."