danimart.ai

FREE GUIDE · JARVIS SETUP

Build your own Claude OS

The setup from the video: one assistant that runs on Claude or Codex, talks with you by voice, sends every request to the right model and keeps everything in your Obsidian vault. Here are the pieces, how they connect and how to build each one with Claude Code.

01
Engines
Claude Code and Codex

02
The Bridge
one brain for every app

03
The Router
right model per request

04
Voice
Whisper and Kokoro, free

05
The Vault
synced with Obsidian

06
Your interface
stats, calendar, skills

danimart.aiClaude and Claude Code are trademarks of Anthropic. Codex is a trademark of OpenAI. Not affiliated with Anthropic, OpenAI or Obsidian.

UNDER THE HOOD

How it works

Two apps on the outside, one brain in the middle. Whatever you type or say goes to the same place, gets sorted, is answered by the right engine and ends up written in your vault.

Obsidian plugin
lives inside the vault

Speech
Whisper listens · Kokoro talks

Jarvis HUD
web app in the browser

The Bridge
one brain for both apps · a small server on your computer

Router
sorts every request

TIER 1
Instant
no AI, saved data

TIER 2
Quick answer
small fast model

TIER 3
Real work
Claude Code or Codex

The Vault · shared memory
markdown files · daily notes · reports · every request

Piece	What it does	Where to get it
Claude Code	Main engine for real work, and the tool you use to build everything else.	code.claude.com/docs
Codex CLI	Second engine. Same requests, OpenAI models.	github.com/openai/codex
Whisper	Speech to text. Open source, runs on your computer.	github.com/openai/whisper
Kokoro	Text to speech. Open weights, Apache license, free.	github.com/hexgrad/kokoro
Obsidian	Your vault: plain markdown files, and a dashboard when you don't want the web app.	obsidian.md
Skills	Your library of reusable instructions for Claude Code.	Skills docs · anthropics/skills

STEP 01

Install the engines

Claude Code. Download the Claude app at claude.com/download and open the Code tab, or install it in the terminal (see the docs). It needs a paid Claude plan.

Codex. Install the CLI and sign in with your ChatGPT account or an API key (Codex CLI docs):

```
npm install -g @openai/codex
```

The trick that makes the OS possible: both engines can run **without opening a chat**, so your own app can call them.

```
claude -p "Summarise today's notes in 5 bullets"
codex exec "Summarise today's notes in 5 bullets"
```

Docs: Claude Code headless mode · Codex non-interactive mode. Pick the engine per request, or keep one as a fallback when the other hits its usage limit.

STEP 02

Build the Bridge

The Bridge is a small server that runs only on your computer. Every app talks to it, never to the engines directly. That's what gives you **one brain for both apps**: same memory, same skills, same history.

- One endpoint receives a request (text, plus where it came from).
- It asks the Router which tier to use, runs it and returns the answer.
- It writes the request and the answer into the vault.

You don't write it by hand. Open Claude Code in a new folder and paste:

```
Build a small local server called "bridge" (Python or Node, your choice).
POST /ask {text, source} -> {answer, tier, engine}.
It must: 1) classify the request with router rules I'll give you,
2) answer tier 1 from local JSON files, tier 2 with
`claude -p --model haiku`, tier 3 with `claude -p` or `codex exec`
in my projects folder, 3) append every request and answer to
[VAULT]/Daily/YYYY-MM-DD.md. Listen on localhost only. Add a README.
```

STEP 03

The Router: right model for each request

Most requests don't need your most powerful model. The Router sorts each one into a tier before anything runs, so simple things are instant and your plan lasts longer.

Tier	Example	Who answers
1 · Instant	"What's on my calendar today?" "How many followers did I get this week?"	No AI. Reads data the OS already saved.
2 · Quick answer	"What are my priorities?" "Rewrite this caption."	A small fast model, e.g. Claude Haiku.
3 · Real work	"Build me a landing page." "Fix the bug in my app."	Claude Code or Codex with files and tools.

Start with simple rules (keywords and length), then let the small model decide when the rules aren't sure. Log every decision so you can tune it.

STEP 04

Voice mode with Whisper and Kokoro

Whisper turns your voice into text. **Kokoro** turns the answer back into a natural voice. Both run on your own computer and are free.

```
pip install -U openai-whisper          # needs ffmpeg installed
pip install "kokoro>=0.9.4" soundfile  # needs espeak-ng installed
whisper question.wav --model small     # quick test from the terminal
```

```
from kokoro import KPipeline
import soundfile as sf
tts = KPipeline(lang_code='a')          # 'a' = American English
for i, (_, _, audio) in enumerate(tts("Good morning. You have 3 tasks today.",
voice='af_heart')):
    sf.write(f'answer_{i}.wav', audio, 24000)
```

The loop: press a key or say a wake word, record, Whisper transcribes, the Bridge answers, Kokoro speaks. Ask Claude Code to wire it up as a "voice" endpoint on the Bridge.

Smaller Whisper models (`base`, `small`) are faster on a laptop. Model list and voices: Whisper README · Kokoro-82M on Hugging Face.

STEP 05

Sync it with Obsidian

An Obsidian vault is just a folder of markdown files, which is why it works so well as shared memory: Claude Code, Codex and you can all read and write it.

- **Daily notes:** every request and answer of the day, written by the Bridge.
- **Reports:** weekly summaries, stats snapshots, finished tasks.
- **Dashboard note:** today, priorities, pipeline. Open it in Obsidian when you don't want the web app.

For a button inside Obsidian, build a small plugin that sends the current note or a question to the Bridge. Start from the official template: obsidian-sample-plugin (how to build a plugin).

> Point Claude Code at the vault in your instructions (CLAUDE.md) so it reads your notes before answering: "My memory lives in [VAULT]. Read the Dashboard and today's daily note first."

STEP 06

Your interface: the Jarvis HUD

A web app in your browser that talks to the Bridge. It's fully yours, so show what matters to you. Mine has:

- **Social media stats:** followers and post results, pulled on a schedule from each platform's API (for Instagram: the Instagram Platform, with a professional account) and saved for tier 1.
- **Calendar:** today's events from the Google Calendar API or the calendar you already use.
- **Voice button:** the Whisper and Kokoro loop from step 4.
- **Skill library:** every skill in `~/.claude/skills`, one click to run it through the Bridge. Give Codex the same rules through an `AGENTS.md` file.

```
Build a single-page web app "Jarvis HUD" that talks to my bridge on localhost.
Cards: social stats, today's calendar, priorities from my vault Dashboard,
skill library (list ~/.claude/skills). A chat box and a push-to-talk
voice button. Dark, clean, mobile friendly. Show me the layout first.
```

BUILD ORDER

Do it in this order

- **Day 1:** engines working from the terminal, plus a vault folder with a Dashboard note.
- **Day 2:** the Bridge with tier 3 only, writing to the daily note.
- **Day 3:** the Router with tiers 1 and 2.
- **Day 4:** the HUD with chat and calendar. Then voice. Then stats and the Obsidian plugin.

One piece at a time, tested before the next. A working small OS beats a half-built big one.

Rules that keep it safe

- **Localhost only.** The Bridge can run commands on your computer. Don't expose it to the internet.
- **Keys in environment variables.** Never inside the HUD code or in your vault.
- **Separate projects folder.** Tier 3 runs inside it, not in your whole disk.
- **Ask before acting.** Anything that sends, posts or deletes waits for your yes.

Prompts for later

Morning brief

```
Read my Dashboard, today's calendar and yesterday's daily note. Give me my top 3
priorities and read them out loud with Kokoro.
```

Tune the Router

```
Read the router log of the last 7 days. Which requests went to tier 3 but could have
been tier 1 or 2? Suggest new rules.
```

Weekly report

```
Write a weekly report in [VAULT]/Reports with what I did, social stats vs last week
and open tasks.
```

Want more setups like this? Join the danimart.ai Telegram group for daily tips, new skills and every new post: t.me/+V8Z1IbLHZgVjYzdk