STEP-BY-STEP GUIDE

# How to Set Up a Personal AI Assistant

## Your own Jarvis: voice, memory and a face

Talk to your AI out loud and hear it answer back.
It remembers everything in an Obsidian vault.
The voice runs free, on your own machine.

Intermediate · 60-90 min · Claude + Obsidian · Claude Pro $20/mo

**@learnwithseb**

One system at a time. Built, not theorised.

# Contents

01 The Game Plan
02 What You Need
03 Give Your AI a Memory First
04 Install the Voice
05 Point the Voice at Your Assistant
06 Put a Face on It
07 Build Your Own Face
08 Talk to It Every Day
09 Validation
10 Keep Building
11 Quick Reference

# The Game Plan

You are going to build a personal AI assistant you can talk to out loud. You hold a key, speak, let go, and it answers in a real voice about a second later. It remembers who you are, because it reads and writes a folder of notes on your own computer. And it has a face on screen that reacts while it listens, thinks and speaks.

This is three separate pieces bolted together. Each one works on its own, so you can stop after any step and still have something useful.

## What you are actually building

- The memory: a folder of plain text notes in Obsidian that your AI reads at the start of every conversation and updates as you work.
- The voice: a small program that turns your speech into text, sends it to Claude, and speaks the reply out loud.
- The face: a web page that animates in time with the voice, so you can see what it is doing at a glance.

## Who this is for

Anyone comfortable copying a command into a terminal. You do not need to code. You will be telling an AI to do the setup for you, and checking its work. Expect an hour, and expect one thing

© 2026 Learnwithseb — Your Own Jarvis

to go wrong. That is normal, and this guide covers the ones that actually happened.

**NOTE**

Both tools in this guide are open-source projects by Jared Rhodenizer (@jaredrhod). The voice is github.com/jaredrhod/backtalk and the face is github.com/jaredrhod/ai-visualizer. This guide rebuilds the setup as a step-by-step reference in plain English, adds the Windows fixes that are not in the documentation, and shows you how to build your own face.

**START HERE IF YOU ARE NEW**

This guide continues from the earlier one, Build an AI That Never Forgets. If you have not set up an Obsidian vault with Claude yet, Step 01 gives you the short version. Do that part first. A voice with no memory is a stranger every morning.

# What You Need

One paid subscription, and nothing else. Everything that makes the voice work runs on your own machine for free.

| What | What it does | Cost |
|---|---|---|
| Claude Pro | The thinking. Every reply comes from here | $20/month (about GBP 18) |
| Obsidian | The memory vault your AI reads and writes | Free |
| backtalk | Hearing and speaking | Free, open source |
| ai-visualizer | The face on screen | Free, open source |
| Kokoro voice model | Speaks the replies, on your machine | Free |
| Whisper model | Turns your speech into text, on your machine | Free |

You also need a microphone, speakers or headphones, and about 2GB of free disk space for the two speech models.

## Why the voice costs nothing

Most voice assistants send your audio to a company that charges per word. This one does not. The model that hears you (Whisper) and the model that speaks (Kokoro) both download once and then run entirely on your computer. No API key, no per-word bill, and your audio never leaves the machine.

**THE HONEST COST**

The voice is free. The thinking is not. Every spoken turn uses your Claude Pro allowance exactly like a typed one. Talking is not cheaper than typing, and it is much easier to ramble out loud. Say "usage report" mid-conversation to hear what you have spent.

STEP 01

# Give Your AI a Memory First

Skip this step if you already have an Obsidian vault wired to Claude. If not, do it now. The voice will be pointed at this folder, and whatever lives here is who your assistant becomes.

Install Obsidian from obsidian.md and create a new vault. Then install Claude Code, which is the desktop app that lets Claude read and write files on your computer. Open it in the folder that will hold your assistant.

You need two folders side by side. One is the vault, which holds the memory. The other is the assistant folder, which holds its personality. Keeping them separate means you can rebuild one without touching the other.

```
Documents/
  My Vault/          <- Obsidian vault, the memory
  My Assistant/      <- personality lives here
```

Open Claude in the assistant folder and paste this.

**PROMPT — COPY & PASTE INTO CLAUDE**

*Set this folder up as my personal AI assistant. Create a CLAUDE.md here that gives you a name, a personality and a set of operating rules, and interview me first so it actually sounds like someone I want to talk to. Point your memory at my Obsidian vault next door: create memory/ for durable facts about me (one fact per note) and daily/ for a note per working day recording what we did, decisions made, and open loops. Add a startup sequence to CLAUDE.md so that at the beginning of every session you read the memory index and the most recent daily note before doing anything else.*

© 2026 Learnwithseb — Your Own Jarvis

**TIP**

Give it a name you will not feel silly saying out loud twenty times a day. You will be speaking this name to end conversations, so pick something short and distinct from ordinary words.

STEP 02

# Install the Voice

backtalk is the voice line. It listens on a key press, transcribes what you said, hands it to your assistant, and speaks the answer sentence by sentence as it arrives.

Put it beside your vault, not inside it. If you clone it into the vault, Obsidian will try to index a gigabyte of program files and slow to a crawl.

```
cd Documents
git clone https://github.com/jaredrhod/backtalk.git
```

On Mac or Linux, run the installer that comes with it. It sets up the environment and downloads the two speech models, which is about a gigabyte on first run only.

```
cd backtalk
./install.sh
```

On Windows there is no installer, so let Claude do it. Open Claude in the backtalk folder and paste this.

**PROMPT — COPY & PASTE INTO CLAUDE**

*Install this project on Windows. The shell scripts here are Mac and Linux only, so do the equivalent natively. Install uv and espeak-ng using winget rather than piping a script from the internet into a shell. Create a virtual environment with Python 3.12, install this package into it, then download the speech models. Tell me what each step is doing as you go, and stop and show me the error if anything fails rather than working around it silently.*

© 2026 Learnwithseb — Your Own Jarvis

**WINDOWS: THE ERROR EVERYONE HITS**

On Windows the install stops with "Microsoft Visual C++ 14.0 or greater is required". One dependency, webrtcvad, has no ready-made Windows version and needs a compiler. You have three choices: install Microsoft C++ Build Tools (2-3GB), or ask Claude to swap in the webrtcvad-wheels package, or ask Claude to remove the dependency entirely. The third option is free and keeps push-to-talk working perfectly. It only costs you hands-free listening, which most people should not use anyway.

**IF YOU HAVE AN NVIDIA GRAPHICS CARD**

Open backtalk.json and set "stt_device" to "cpu". Left on the default, it tries to use your graphics card and fails with a missing library the moment you speak. The program starts perfectly and then simply never answers, which looks exactly like a broken microphone.

STEP 03

# Point the Voice at Your Assistant

backtalk has no personality of its own. It borrows your assistant's. All the configuration lives in one small file called backtalk.json in the backtalk folder.

The two settings that matter most are agent_dir, which points at the folder holding your assistant's CLAUDE.md, and extra_dirs, which gives it permission to reach your vault.

```
{
  "agent_dir": "C:/Users/you/Documents/My Assistant",
  "name": "Jarvis",
  "extra_dirs": ["C:/Users/you/Documents/My Vault"],
  "permission_mode": "ask",
  "mic_mode": "ptt",
  "ptt_key": "right_ctrl",
  "voice": "bm_lewis",
  "stt_device": "cpu"
}
```

You never need to edit this by hand. Ask Claude to do it, in the same session you used to install.

© 2026 Learnwithseb — Your Own Jarvis

**PROMPT — COPY & PASTE INTO CLAUDE**

*Configure backtalk.json for me. Point agent_dir at my assistant folder and add my Obsidian vault to extra_dirs so it can reach its memory. Set the name to match my assistant. Keep permission_mode on ask so it checks with me before doing anything that changes files. Use push-to-talk, not hands-free. Set the talk key to a key I never type with, and tell me which one you chose. Then read the config back to me in plain English so I can confirm it is right.*

## Choosing a talk key

The default is the Home key. Plenty of keyboards do not have one. Right Ctrl works on almost every keyboard and nobody presses it on its own. Avoid right Alt on Windows: it registers as a different key internally, so it silently never fires and looks like a broken install.

**PUSH-TO-TALK IS THE RIGHT DEFAULT**

Hands-free listening sounds better than it is. With an open microphone, a video playing, music with vocals, or someone else in the room can all trigger your assistant. With push-to-talk the microphone is shut unless you are holding the key, so nothing records in the background.

**IF IT CANNOT HEAR YOU**

backtalk records from whatever Windows has set as the default microphone. If you have an audio interface, a webcam and a capture card, the default is often not the microphone you actually speak into. Ask Claude to list your input devices and add a setting to record from the right one.

STEP 04

# Put a Face on It

The face is a web page that opens full screen and animates while your assistant works. It is genuinely useful, not just decoration: you can see at a glance whether it is idle, hearing you, thinking or speaking, without staring at a terminal.

It needs nothing installed. It is one small Python file and some web pages, with no dependencies at all.

© 2026 Learnwithseb — Your Own Jarvis

```
cd Documents
git clone https://github.com/jaredrhod/ai-visualizer.git
cd ai-visualizer
python server.py
```

Your browser opens on the face. Four designs ship with it, and you can click between them in the gallery at the root address.

### How the two pieces talk to each other

This is the clever bit, and it is much simpler than you would expect. The voice writes its current state into a tiny file. The face reads that file eight times a second. That is the whole connection. No network, no database, no configuration protocol.

```
.voice_state       idle | listening | thinking | speaking
.voice_waveform    the shape of the sound being spoken
```

You just need to tell one side where the other one lives. Point backtalk's signals_dir at the ai-visualizer folder, or point the visualiser's bus_dir at the backtalk folder. One direction, not both.

**PROMPT — COPY & PASTE INTO CLAUDE**

*Wire the visualiser to my voice line. Set signals_dir in backtalk.json to my ai-visualizer folder so the voice writes its state there, and leave the visualiser reading its own folder. Then prove it works without me having to talk: write a test state to the file and check the server reports it back correctly, and tell me what you saw.*

**TIP**

Run the visualiser on its own with the demo flag before you connect anything. It plays a scripted loop of all four states, so you can pick a face you like before worrying about whether the wiring works.

STEP 05

# Build Your Own Face

The four faces that ship are examples, not limits. Adding your own is deliberately easy: create a folder inside faces/, put an index.html in it, and it appears in the gallery automatically. That is the entire plugin system.

© 2026 Learnwithseb — Your Own Jarvis

```
ai-visualizer/
  faces/
    board/
    radial/
    my-face/          <- yours
      index.html
      face.json
```

face.json is just a title and a one-line description for the gallery.

```
{
  "title": "My First Face",
  "tagline": "A ring that breathes with the voice."
}
```

## The five values you draw from

Your page loads core.js, which handles all the plumbing. It gives you a handful of live values that update every frame. Draw whatever you like from them.

| Value | What it gives you |
|---|---|
| AV.state | idle, listening, thinking or speaking |
| AV.env | 0 to 1, how loud the voice is right now, smoothed |
| AV.level | 0 to 1, the raw loudness, jumpier |
| AV.samples | 64 numbers describing the shape of the sound |
| AV.micLevel | 0 to 1, your own microphone, if you ask for it |

A complete working face is about twenty lines. This one draws a circle that grows with the voice and changes colour when speaking.

© 2026 Learnwithseb — Your Own Jarvis

```
<canvas id="stage"></canvas>
<script src="../../core.js"></script>
<script>
AV.init({});
const cv = document.getElementById("stage");
const ctx = cv.getContext("2d");

function frame(dt) {
  AV.tick(dt);
  cv.width = innerWidth;
  cv.height = innerHeight;
  ctx.fillStyle = "#050907";
  ctx.fillRect(0, 0, cv.width, cv.height);

  const r = 80 + AV.env * 160;
  ctx.strokeStyle =
    AV.state === "speaking" ? "#4af07a" : "#2a6b45";
  ctx.lineWidth = 3;
  ctx.beginPath();
  ctx.arc(cv.width / 2, cv.height / 2, r, 0, 6.29);
  ctx.stroke();
}

let last = performance.now();
(function loop(now) {
  frame(now - last);
  last = now;
  requestAnimationFrame(loop);
})(performance.now());
</script>
```

In practice you will not write this yourself. Describe the face you want and let Claude build it, then refine it by telling it what looks wrong.

**PROMPT — COPY & PASTE INTO CLAUDE**

*Build me a new face for the ai-visualizer in faces/my-face/. Read core.js first so you use the real values it exposes, and copy the structure of an existing face rather than inventing your own. I want a slowly rotating wireframe globe on a near-black background that breathes gently while idle, tints amber while it is listening to me, spins faster while thinking, and pulses in time with the voice while speaking. Everything must be drawn on a canvas with no external libraries, because this project deliberately has no dependencies. Add a face.json with a title and tagline, then tell me how to preview it in demo mode.*

Preview it without needing to talk by adding ?demo=1 to the address. The face runs through a scripted loop of every state, so you can check all four look right in about thirty seconds.

© 2026 Learnwithseb — Your Own Jarvis

**REFINING IT**

Describe the problem, not the fix. "The pulse is too twitchy on quiet words" gets you a better result than "change the smoothing to 0.4". You are the one with eyes on it; let Claude work out which number to move.

**KEEP IT CHEAP TO DRAW**

This runs sixty times a second, often on a second monitor all day. Thousands of individually drawn shapes will heat up your machine and spin the fans. If your laptop gets loud, tell Claude to reduce the number of things being drawn per frame.

STEP 06

# Talk to It Every Day

Start the voice line, hold your talk key, speak, let go. The reply comes back in a second or two. Press the key again while it is talking to interrupt it, which you will do constantly once you get used to it.

You can also type into the same window. It is the same conversation and you still get a spoken reply, which is handy for anything fiddly to say out loud like a file path.

## Phrases that control the session

A few exact phrases, spoken on their own, control the conversation itself rather than being sent to your assistant.

| Say this | What happens |
|---|---|
| usage report | Tells you the turns and tokens you have spent |
| clear the session | Fresh start, forgets this conversation |
| switch to the deep model | Smarter, noticeably slower |
| back to the fast model | Returns to normal speed |
| goodbye [name] | Ends the session |

## Things worth asking it

The point of the memory vault is that these questions work on day thirty, not just day one.

© 2026 Learnwithseb — Your Own Jarvis

**PROMPT — COPY & PASTE INTO YOUR ASSISTANT, OUT LOUD**

*What did we work on yesterday, and what did I leave unfinished?*

**PROMPT — COPY & PASTE INTO YOUR ASSISTANT, OUT LOUD**

*I have had an idea I do not want to lose. Save it to my vault with today's date, link it to anything related you already have, and read the summary back to me so I know you got it right.*

**PROMPT — COPY & PASTE INTO YOUR ASSISTANT, OUT LOUD**

*Remember this from now on: I prefer short answers unless I ask you to go deeper. Write it to your memory folder so it survives after this conversation ends.*

**THE BEST USE OF A VOICE ASSISTANT**

It is at its best when your hands are busy and your brain is not. Walking, washing up, driving, or pacing around the room working through a problem. Sitting at a desk, typing is usually still faster.

**IT WILL ASK PERMISSION**

With permission_mode on ask, your assistant checks out loud before writing files or running commands. Say "yes" to approve. Anything else denies it, and your actual words are passed back as the reason, so "no, put it in drafts instead" works as a redirect rather than a flat refusal. Silence for 75 seconds counts as no.

# Validation

Work down this list in order. Each item rules out a whole category of problem, so when something fails you know roughly where to look.

- ✓ The voice line starts and speaks a greeting out loud
- ✓ You hold the key, say something, and it answers within a couple of seconds
- ✓ Pressing the key mid-reply stops it within a syllable
- ✓ Interrupt it, immediately ask something different, and the answer matches the NEW question. Do this three times
- ✓ Typing in the window gets a spoken reply too

© 2026 Learnwithseb — Your Own Jarvis

- ✓ Ask it something that needs a file, and it asks permission out loud first
- ✓ Say "usage report" and hear real numbers back
- ✓ The face reacts as it listens, thinks and speaks
- ✓ Say "goodbye [name]" and it signs off cleanly

### When something goes wrong

There is a log file at logs/backtalk.log inside the backtalk folder, and it answers most questions faster than guessing does.

| What the log shows | What it means |
|---|---|
| Nothing at all when you press the key | The talk key is not registering. Choose a different key |
| A line starting [you] with your words | It heard you correctly. The problem is further down |
| [you] but no reply line after it | Your Claude login has expired. Sign in again |
| A reply line but no sound | It is speaking to the wrong audio output |

**THE FAILURE THAT FOOLS EVERYONE**

If your Claude login expires, the voice line still starts, still hears you, and still transcribes you perfectly. It simply never answers, with no error anywhere on screen. It looks exactly like broken hardware and it is not. Check the log for a reply line before you touch anything else.

# Keep Building

You now have the three pieces working together. Here is where it goes next.

- Teach it recurring jobs. Ask it to write down how you like your weekly review done, then say "run my weekly review" and let it follow its own instructions.
- Audition other voices. Around sixty ship free with Kokoro, across several languages. Change one line in the config and listen to a few before settling.
- Build a second face for a different mood. A calm one for working, a busier one for recording.
- Put the face on a spare monitor or an old tablet. It costs almost nothing to run and turns the assistant into something you glance at rather than open.
- Keep a genuine daily note. The memory is only as good as what goes into it, and the daily note is the habit that makes the difference.

### One honest word on safety

Both projects are open source, and both are worth reading before you install them. Anything that listens to your microphone and can write files on your computer deserves five minutes of scrutiny. You do not have to read code to do this: ask Claude to clone the repository and audit it before you run anything.

**PROMPT — COPY & PASTE INTO CLAUDE**

*Before I install this, review it for safety. Clone the repository somewhere temporary and read every source file, the commit history and any setup files written for an AI agent to execute. Tell me what it sends over the network, what permissions it needs and why, and anything that looks obfuscated or out of place. Give me a plain-English verdict, and be specific about what you did NOT check.*

**WHAT I FOUND WHEN I DID THIS**

Both repositories came back clean. No hidden network calls, no obfuscated code, no tracking. The permission system in particular is carefully built. That is a snapshot of one point in time, though, not a permanent guarantee, and it is worth repeating the check on any project you hand access to your files.

# Quick Reference

| Thing | Where it is |
|---|---|
| The voice | github.com/jaredrhod/backtalk |
| The face | github.com/jaredrhod/ai-visualizer |
| Your settings | backtalk/backtalk.json |
| The log when things break | backtalk/logs/backtalk.log |
| Your own faces go here | ai-visualizer/faces/your-face/ |
| Preview a face with no voice line | Add ?demo=1 to the address |
| Talk key that works on most keyboards | right_ctrl |
| Fix for NVIDIA machines | Set stt_device to cpu |

## The settings you will actually change

| Setting | What it does |
|---|---|
| agent_dir | The folder holding your assistant's personality |
| extra_dirs | Other folders it may read, such as your vault |
| name | Its name, used to end the session |
| ptt_key | The key you hold to talk |
| voice | Which voice speaks. bm_lewis is British male |
| permission_mode | Keep on ask so it checks before acting |
| stt_device | Set to cpu if the replies never come |

**CREDIT**

backtalk and ai-visualizer are built and maintained by Jared Rhodenizer (@jaredrhod) and released under the AGPL-3.0 licence. This guide is an independent walkthrough, not an official manual. If the tools are useful to you, the projects are worth supporting.


© 2026 Learnwithseb. All rights reserved.