CASE STUDY

Portfolio Website

I designed this site and built it with two AI coding agents, Claude Code and OpenAI Codex. This is how the work was split, which tools the agents were connected to, and the problems we found and fixed along the way.



The homepage. A portrait in a gilded frame above a brass and enamel astrolabe that was modeled in Blender. The birds, pines and sea in the painting move.

LIVE SITE	bhavportfolio.vercel.app
TIMELINE	April to October 2026, about 110 commits
BUILT WITH	Claude Code running Claude Opus, and OpenAI Codex
STACK	Next.js 16, React 19, TypeScript, Three.js, Blender, Python
MY ROLE	Direction, every design decision, review and approval of each release
ABOUT THIS PDF	Written by AI (Claude Code), reviewed and approved by Bhavya

How the work was split

I did not write this site's code by hand. I decided what it should be, judged every result, and gave feedback until it matched what I had in mind. Two agents did the building.

ME

Set the direction and supplied the reference images.

Chose between the options the agents offered.

Reviewed every change on my own machine and approved each release.

OPENAI CODEX

Wrote the plans and quality checklists for the Poetry and Books rooms.

Generated the painted artwork: the coastal landscape and the Books background layers.

Modeled the poetry scroll end caps in Blender, and started the Projects stone archive.

CLAUDE CODE

Built most of the site: the Blender astrolabe, the Poetry and Books rooms, the stone archive and all of the animation.

Composed the music, fixed performance problems, wrote the tests and shipped each release.

The loop

Brief I describe the goal and share references › **Plan** Codex writes the plan and makes the artwork › **Build** Claude Code works on a separate branch › **Review** I check it on my machine › **Release** "Push it", and Vercel deploys

Review usually sends it back to Build several times before a release.

One shared memory

The two agents never talk to each other. They share two files in the repository. **AGENTS.md** holds the rules: how to work, what not to touch, what to run before finishing. **HANDOFF.md** holds the current state: what is live, what is half done, and why each decision was made. Every session ends by updating it.

This mattered in practice. Codex planned the Poetry room and started building it, then ran out of tokens. Claude read the plan and the handoff, kept the parts Codex had finished, and completed the room without starting over. It happened again with the Projects archive: Codex built its model pipeline, ran out of usage, and Claude finished it from the same notes.

Rules I set for the agents

- Nothing reaches the live site until I say "push it".
- Ask every question up front when I am about to step away.
- Keep the test suite small: a test has to guard a real risk.

Tools the agents were connected to

The agents did more than write code. They ran other programs, looked at the results and adjusted, much like a person would.

TOOL	WHAT IT WAS USED FOR
Blender 5, driven by Python	The astrolabe, the poetry scrolls and the stone archive are built by scripts. The agent writes the script, runs Blender in the background, inspects the renders and adjusts the geometry or materials.
Image generation in Codex	Painted artwork in one consistent style: the homepage coast and the separate layers of the Books background.
Python with NumPy and Pillow	Textures for the astrolabe, cutting animation layers out of the paintings, image compression, and synthesizing all of the music note by note.
Playwright and headless Chrome	Screenshots for my review, frame by frame recordings to find animation bugs, timing measurements, and the automated browser tests.
GitHub and the gh command line	Branches, commits, and checking whether each deployment succeeded.
Vercel	Every push to the main branch deploys the live site. Pushing a branch gives a private preview link.
Sound tools	macOS afconvert encodes the music so each track loops without a gap. Recorded effects come from Pixabay: the agent built an audition page, I picked, and a small script cut each clip without re-encoding.
Claude desktop app	Most of the work happened in its Code tab, with a built-in browser pane, so the agent could check pages without taking over my screen.

Judging the result against the concept



Concept painting on the left, the finished 3D dial on the right. The agent produced side by side sheets like this for every part of the dial, so I could judge it piece by piece and point at exactly what was off.

Homepage

A portrait in a gilded frame above an astrolabe. Choosing a year turns the dial, and the painting changes to that year.

The astrolabe

It is a real 3D model, not an image. Every ring, gear, star and numeral is generated by Python inside Blender from a concept painting, then exported for the browser. The rings turn at different speeds and in different directions, and the chosen year flashes and then settles into a steady glow.

The painting comes alive

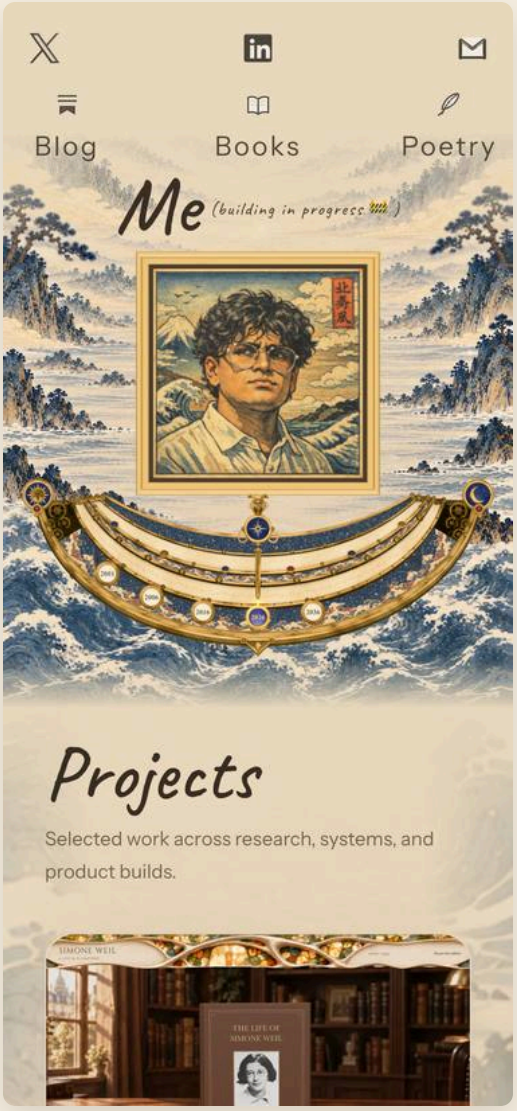
The coastal painting behind it was made with image generation. Its birds now glide, the pines sway and the sea swells. The painting itself never changes. A light layer on top covers each painted bird with a patch of plain sky and redraws it in flight. Only the water is cut out to move, so the rocks stay still.

Staying smooth

The motion pauses when the top of the page scrolls out of view, when the tab is hidden, and while you drag the dial, so the 3D model always gets the full frame. Visitors who ask their device for reduced motion see the still painting.

The bio

Clicking the 2026 portrait reveals a short handwritten note. The same text is also in the page's hidden data, so search engines and AI assistants can read it without clicking.



On a phone the same composition stacks, and the sea keeps moving below the dial.

Poetry

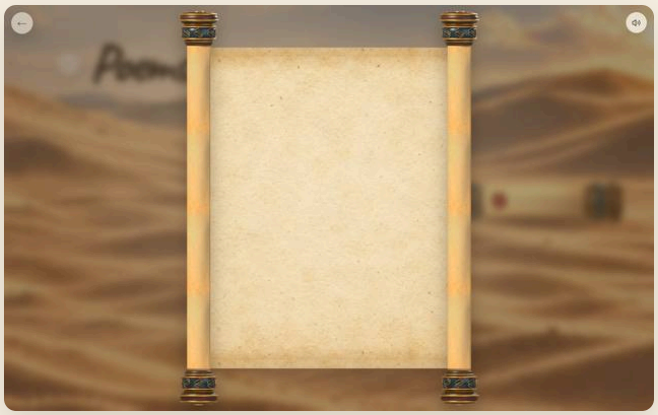
Poems live as scrolls lying in a desert. Opening one lifts it off the sand, turns it upright and unrolls it.



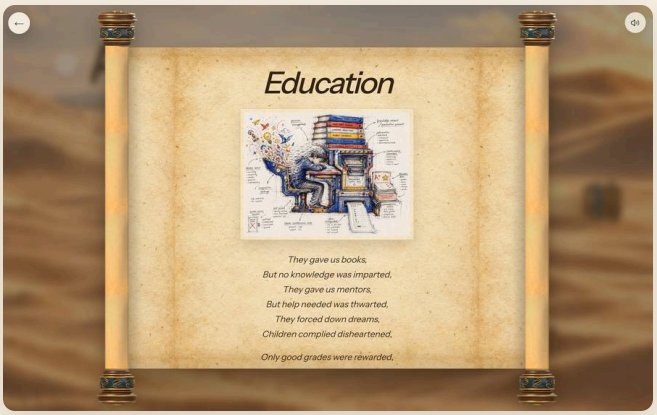
The concept I chose.



The finished page.



Mid opening: the rods part and the paper unrolls.



The poem, as real text on the paper.

- The scrolls, rods and wax seal are Blender renders. The render script also records where the paper band sits in each image, so the titles are real text placed exactly on the paper.
- The poem is text, not a picture. It stays sharp, screen readers can read it, and each poem has its own address that works even without JavaScript.
- The blowing sand is a small particle simulation: grains hop along the ground, thin sand snakes wriggle across it, and gusts tear spray off the dune crests.
- The music is an original piece in an Arabic mode, maqam Hijaz, synthesized in Python: a breathy ney, a plucked oud, a low drone and a soft frame drum. It starts only after the visitor clicks, and it remembers if they turn it off.

Books

My recommended books float in a spiral in painted space.

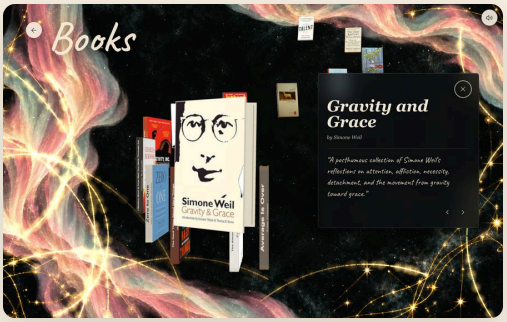


The concept I chose.



The finished page. The books are real 3D objects you can scroll through.

- Codex painted the background as separate layers: a starfield, the ribbons and a gold star chart. I wanted the gold arcs fuller, like the concept, so Claude wrote a script that paints matching orbit lines and constellations onto Codex's layer.
- The ribbons wave, light travels along the gold, and shooting stars fall. It is all drawn on a separate flat layer behind the 3D books, so it never gets in their way, and it holds still while you read.
- The music, "Drift", took three rounds. The first options sounded too much like a synthesizer and too intense. The final one uses soft breathy pads and a muted felt piano in bright, calm chords.
- On phones the art is rearranged rather than cropped: the ribbons stay pinned to the edges and the middle stays calm for the books.



Selecting a book brings it forward with a short summary.

Projects: the stone archive

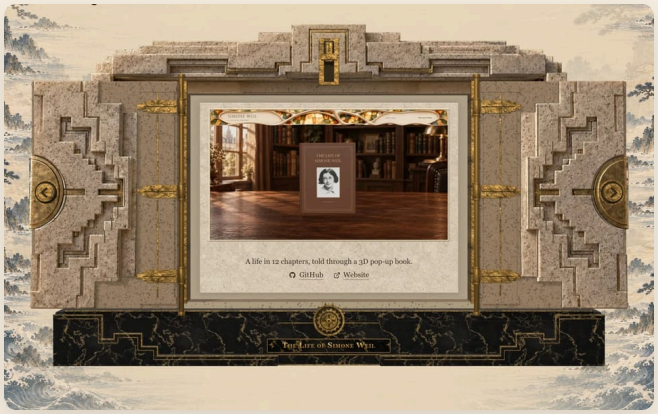
My projects sit in a carved stone archive, one at a time. The gold buttons move through them; the latch at the top swings the doors shut and locks them.



The concept I chose for the closed archive.



The finished archive, closed. Its four reliefs come from the concept.



Open: one project at a time, its name on a plate that slides in the plinth.



Closing: the carved doors swing towards you on their hinges.

- Codex projected the approved painting onto real carved geometry in Blender. After my feedback, Claude made the carved wings themselves the doors: they swing shut, the two gold half-circles meet as a seal, which turns its gold strip as brass pistons drive steel rods into it.
- The buttons sink when pressed; a gear on the plinth turns the way you travel and slides the name plates along their track. All the metal shares one aged-brass texture the agent generated; the plinth is gold-veined black marble.
- For sound, the agent gathered 20 recorded candidates on an audition page and I picked one per moment: a brass click, a ratchet, stone grinding, a thud and a vault lock.

Problems we found and fixed

The most useful part of the process was not the first build. It was noticing what felt wrong and finding the real cause. Each fix started with a measurement or a frame by frame recording.

PROBLEM	CAUSE	FIX AND RESULT
The first swipe on Books jumped instead of gliding	The 3D scene only redraws when something changes. After sitting idle, the first new frame counted several seconds as one step, so the books jumped to the end.	Treat a long idle gap as one normal frame. The motion now spreads over about 45 frames instead of 1.
One swipe sometimes moved two books	A trackpad keeps sending scroll events for over a second after your fingers lift, longer than the old 0.65 second lock.	Step on the strength of the scrolling, using each event's own timestamp. A flick with 1.5 seconds of momentum now moves exactly one book.
The first poem opened with a stutter	That first click also started the music, and creating the browser's audio engine freezes the page for about 130 ms.	Create the audio engine quietly while the page loads, so the click only switches it on. No slow frames remain.
Books loaded slowly from the homepage	One book cover was 929 KB, more than all the others together, and images only started downloading after the scripts ran.	Resize that cover, compress the art and start downloads with the page. 0.4 seconds instead of 1.2 to 2.3 on my machine.
Letters clipped on the scroll titles	The title box hid anything outside it, cutting off the tail of the g and the slant of the italic n.	Give the letters room without letting long titles overflow. Checked at nine screen widths.
Projects flashed the old list on every first visit	For about 4 seconds, while 3.3 MB of 3D model loaded, the page showed the plain list, then jumped.	Show a still of the closed archive at its final size, and halve the model. Ready in 1.4 seconds, and nothing moves.
The new astrolabe sound was silent on a laptop	96% of its sound was below 150 Hz, which laptop speakers barely play.	Soften the deepest rumble and lift the part small speakers can play. Audible everywhere, without booming on headphones.
The test suite grew too large	While building, it reached 20 unit tests and 45 browser tests across four browsers.	Once things were stable, keep only tests that guard real risks. Today: 12 unit tests and 8 browser tests.

See it live: bhavportfolio.vercel.app

Source: private GitHub repository, available on request

Contact: bhavskhimavat@gmail.com

This case study was written by AI (Claude Code) from the project's history and notes, and reviewed and approved by Bhavya Avinash Khimavat.