# Parker Throneberry

Software Developer · Austin, TX

parkert.dev Résumé Email LinkedIn GitHub GitLab

## Work Experience

### Pushnami

Software Developer · Austin, TX · July 2026 - Present

Building cloud infrastructure and distributed services for high-volume push notifications and advertiser-facing marketing data products.

- Built a popunder that opens at configurable points across Pushnami's sites and added about $320K in annualized revenue.
- Found and fixed user misclassification in a legacy system, recovering $168,000 in annual revenue.
- Led research validating safe removal of a third-party API, then removed it to save $102,000 per year.
- Saved $40,000 per year by optimizing the primary algorithm to use one-seventh as many database calls.
- Saved $10,000 per year by keeping transient data in DynamoDB with TTLs and permanent records in S3.
- Automated the replacement of flagged domains to preserve service availability and reduce manual intervention.
- Designed agentic development workflows that coordinate specialized subagents, run scheduled tasks, package reusable tools and skills, and produce Grafana dashboards after delivery to measure project outcomes.

### Paycom

Software Developer · Irving, TX · Nov 2024 - July 2026

Built core scheduling software and APIs, along with iWant, an LLM-powered experience integrated directly into scheduling workflows within a high-availability payroll platform.

- Built and maintained core scheduling software using React, PHP, and MySQL, powering workflows used by 37,500+ client organizations.
- Built iWant, an LLM-powered assistant using GraphQL APIs, enabling natural-language scheduling workflows and improving user satisfaction by ~30%.
- Built backend services and REST APIs in C# and PHP supporting high-traffic scheduling workflows.
- Developed external APIs enabling client integrations and extension of scheduling functionality.
- Contributed to migration from a monolithic system to service-based architecture while improving performance.
- Delivered production features in a regulated Agile environment and mentored interns while expanding test coverage.
- Built reusable skills, agents, and agentic workflows adopted by developers across the company, turning successful AI-assisted engineering patterns into repeatable tools for broader teams.

### Printforgers

Software Developer (Independent Project) · Dallas, TX · Sep 2023 - Present

A full-stack 3D printing platform with automated quoting, real-time pricing, and a unified checkout system.

- Built the platform with React, TypeScript, Node.js, and PostgreSQL for file uploads, automated pricing, and checkout workflows.
- Developed an automated pipeline to analyze uploaded models, generate previews, and produce real-time quotes.
- Designed a server-authoritative pricing and order system with snapshot-based persistence for consistent, auditable checkout.
- Built asynchronous backend workflows for file processing and quote generation.
- Containerized services with Docker, fronted with Nginx, and deployed via GitLab CI/CD.
- Built a generative workflow where customers upload reference images and a model running on my own server produces 3D assets that can move into Printforgers quoting and production.

## Technical Projects

### PocketShare

Marketplace platform · Open beta · Sep 2026

A marketplace for renting real Android phones by the minute. Hosts connect spare phones to a PocketShare hub, and developers and QA teams rent the exact model they need, then use it through adb and Android Studio or straight from the browser.

- One `pocketshare connect` puts the rented phone in `adb devices`, so Android Studio, scrcpy, Espresso, Appium, and Maestro use it like a phone on the desk.
- In the browser, the live screen takes taps, swipes, and typing, dropping an APK installs it, and a shell, Logcat, and file transfer sit next to the screen.
- Every CLI command takes `--json` and returns a checkable exit code; `pocketshare run --max-spend` rents, connects, runs a test command, and always ends the rental.
- AI agents rent through an API key and the CLI, or the MCP server at pocketshare.io/mcp, under a daily spending limit.
- Hosts plug phones into a hub over USB; a lease cuts access the moment a rental ends, and every phone is wiped and checked before the next one.
- New accounts start with $1 of credit, and a separate site at getpocketshare.com walks phone owners through hosting.

Developer site  Host site  View on GitLab

### HostForge

Game server hosting · Live · Jun 2026

Game server hosting on my own hardware. Customers choose a Minecraft: Java Edition or Terraria plan, pay through Stripe, and get a provisioned server with a join address to share.

- One dashboard per server: live console, file uploads and editing, game settings, backups and restore, and sharing with friends.
- Paid orders provision containers on enrolled worker nodes through Pelican Panel and Wings, with no per-order operator work.
- Each server gets a public join address through DNS-only SRV and CNAME records published to Cloudflare.
- Releases go through a protected GitLab pipeline, and rollback is a plain snapshot of the databases, protected configuration, and previous images.
- The domain previously hosted HostForge Atlas, a Minecraft server-sizing calculator and provider comparison.

View live  View on GitLab

### Tools For Free

File converter · Live · Sep 2026

A free file converter and toolkit with 2,900+ conversions across 180+ formats, plus tools to compress, merge, trim, OCR, and password-protect files. No accounts, watermarks, or cookies, and files sent to the server are deleted automatically after a few minutes.

- Tools are data and engines are containers: a new tool is a registry row, and every conversion route is derived from engine capabilities, so no page exists for a conversion nothing implements.
- Engines parse hostile input with no network, a read-only root, every capability dropped, and a Unix socket as the only way in or out.
- PNG, JPG, and WebP conversions, font and e-book conversions, and tools such as merge PDF, images to PDF, and QR codes run entirely in the visitor’s browser.
- Everything is reachable without a browser too: a keyless public API with an OpenAPI document, an llms.txt catalog, and a remote MCP server.
- The deploy job brings the stack up and drives a real conversion through every engine before it reports success.

View live  View on GitLab

### Austin Tech Recycling

Local business platform · Live · Sep 2026

The website and operations app for Austin Tech Recycling, a local Austin business that picks up electronics, buys back business IT, and clears out offices. Customers describe their equipment without an account, and one operator workspace takes each request from inventory review to a released proposal.

- Guest intake saves the original submission before any integration runs and gives the customer a secure status view.
- Parsing organizes rough equipment lists with a deterministic fallback, but never decides money, access, timing, or outcomes.
- Offers come from deterministic economics and comparable-sale evidence, and nothing reaches the customer until an operator releases it.
- Operator alerts for new requests and replies carry no contact details or message contents.
- Playwright covers guest intake, sign-in, accessibility, and the operator proposal flow at phone, tablet, and desktop widths.

View live  View on GitLab

### Homelab Network Stack

Network infrastructure · Operating · Feb 2026

The DNS and edge layer every other service in the lab runs on. AdGuard Home filters client queries, CoreDNS answers the internal .lan zone, and Unbound resolves from root hints with DNSSEC validation. Nginx terminates TLS for public hostnames, and Cloudflare DDNS keeps their records current.

- Unbound resolves recursively and validates DNSSEC rather than forwarding to a public provider.
- Internal services are addressed by .lan hostname, never by IP.
- A new public app ships in three ordered merges (app, then Nginx edge, then DDNS), so no record points at an upstream or certificate that is not ready.
- Each service keeps its own repository, CI, and deploy; the meta repository only pins versions and never rewrites their history.
- Prometheus and Grafana watch three Proxmox hypervisors (node, API, and ZFS exporters) alongside LAN service metrics.

View on GitLab

### Trace Zero

Privacy suite · In development · Aug 2026

A privacy suite with two deliberately separate halves: Device explains what the apps on a phone tell the internet using local network metadata only, and Remove runs authorized opt-out and deletion workflows against supported public sources.

- Device keeps everything local: app-private SQLite with versioned migrations, user-controlled retention, confirmed deletion, and user-initiated JSON and CSV export. No account, and no TLS interception.
- Destination classification and an explainable exposure score that reports honest unknowns instead of guessing.
- Remove depends on PostgreSQL alone: commands atomically write business state and durable jobs, and separately gated workers lease their own queue with expiring leases, delayed rescheduling, and poison-job failure.
- Queue rows deliberately carry no profile values, receipts, URLs, or raw errors, and malformed jobs or broker results fail closed.
- The Android VpnService foundation and metadata parser are scaffolded with packet forwarding switched off until connectivity tests pass.

View on GitLab

### Weather API

Public API · Live · Mar 2026

A free public weather API with no accounts or API keys. It normalizes US National Weather Service and MET Norway data into one consistent JSON format, keeping each source separate with its attribution and upstream timestamps.

- A partial result returns 200 with explicit per-source errors, and a failed alert source returns an error, never a fabricated empty alert list.
- Fair use is 60 requests per minute per IPv4 address or IPv6 /64, with RateLimit headers and 429 responses that carry Retry-After.
- Upstream calls share a bounded queue, a concurrency limit, request coalescing, cooldowns after failures, and a bounded cache that honors provider cache headers.
- Offline regression tests mock every upstream service.

View live  View on GitLab

### Texas Tech Bio App

Biolab automation · Live demo · Jul 2024

Built for the Texas Tech biology lab to match DNA sequences against its primer inventory, reducing a repetitive lab processing workflow by more than 90%.

- Enter a DNA strand, upload one or more primer CSV exports, and get a table of every matching primer with its serial number, name, date, investigator, and concentration.
- Each primer is checked against the strand and its complement; RNA input is normalized to DNA, and primers marked as reverse are flipped before matching.
- Packaged with electron-builder as a Windows installer, so the lab could run it without a server.
- The hosted demo ships example RNA sequence data, so it can be tried without the lab’s files.

View live  View on GitLab

### GPU Performance Visualizer

Community tool · Live demo · Mar 2024

Built a crowdsourced GPU comparison tool that a mining community actively used over an extended period to evaluate hardware and tune configurations.

- Collected community-submitted benchmark results and turned them into filterable, comparable hardware records.
- Used by a mining community over an extended period to choose hardware and tune farming configurations.
- The hosted build is now a frontend-only viewer of the collected dataset, kept online without its original MySQL backend.

View live  View on GitLab

### Grand Cherokee openpilot Research

Vehicle research · Offline research · Oct 2026

Offline groundwork toward an openpilot port for my own Jeep Grand Cherokee, built on comma.ai existing tooling rather than a new driving model. Everything so far runs on local files: no vehicle captures, no validated signals, and no control results yet.

- Seven offline commands that only ever read local files, so nothing in the repository can open vehicle hardware.
- A dated source ledger, synthetic fixtures, and a fully pinned dependency lock so the toolchain reproduces without network access.
- Imported DBC values stay marked as hypotheses until validated, and unmatched or invalid frames stay visible instead of being silently dropped.
- Capture, harness, and Raspberry Pi deployment are documented designs only, gated behind an explicit evidence milestone. Nothing has been verified on the vehicle.
- Independent work with no comma endorsement, and no claim of openpilot compatibility for this vehicle.

View on GitLab

### 3D Printing Web Store

Printforgers precursor · Archived demo · Sep 2023

My first extensive product build: model uploads, generated previews, print configuration, real-time quotes, and checkout. It became the precursor to Printforgers.

- Users upload an STL and pick a print quality; the server slices it with PrusaSlicer to produce gcode and an estimated price.
- Established the upload, quoting, visualization, and commerce workflows later rebuilt at a larger scale as Printforgers.
- The hosted build is an archived demo: Stripe is removed, no payment is processed, and uploads are size- and complexity-capped so they cannot overload the small host.

View live  View on GitLab

### Fitbit Chia Pool Stat Viewer

Wearable app · Published to the Fitbit gallery · Nov 2023

Built a Fitbit companion app that brought Chia farming statistics from a phone-side API integration to Sense and Versa watches.

- Published to the Fitbit App Gallery for Sense and Versa devices.
- The paired phone app calls the SpaceFarmers API and forwards a compact payload to the watch, which renders it for a glanceable on-wrist view.
- The farm identifier is entered once in the phone settings, so the watch app needs no network access of its own.

View live  Parker Throneberry's Fitbit App Store Page  View on GitLab

### H.265 Video Converter

Weekend build · Live demo · Aug 2023

A weekend React and Node.js project that wrapped H.265 transcoding in a straightforward upload, progress, and download workflow.

- The Node.js backend drives HandBrake to re-encode uploads to HEVC and streams the finished file back.
- The interface reports original and converted file sizes, so you can see whether compression actually helped.

View live  View on GitLab

### Capstone Fitness App

College capstone · Read-only showcase · Feb 2024

Led a six-person capstone team, taught five teammates React, and built the complete backend, database, and most of the frontend for a fitness tracker.

- Led a six-person team and taught five teammates React while owning the Node.js, Express, and MySQL architecture.
- Accounts, fitness goals, weight-based nutrition recommendations, and workout logging with reps, sets, and weight.
- The hosted build is a read-only showcase: the original MySQL account system was retired in favour of demo data.

View live  View on GitLab

### Homepage App

First React project · Still in daily use · Aug 2023

My first React project: a personal browser homepage with weather, custom backgrounds, saved settings, and Google search.

- My first React project, still deployed and still in daily use as a browser start page.
- Search, custom backgrounds, and saved settings, plus a ZIP-based weather widget.
- The weather widget now reads from a separate homelab weather service that prefers recent station observations and falls back to a clearly labelled forecast, with per-ZIP caching and request coalescing.

View live  View on GitLab

## Education

**University of North Texas at Dallas**
B.S. in Information Technology · Aug 2021 - May 2024