Utsav Khairnar

Bangalore, India +91 7226008635 utsavkhairnar14@gmail.com github.com/utsav1033 linkedin.com/in/utsav-khairnar

Experience

Engineering Intern | Alma (Memfold AI), backed by Peak XV *Jul 2026 – Present*
Bangalore, On-site

- Replaced the computer-use agent's 72k-character LLM planner with a one-pass classifier over **27 tools**: task time **33.2 s** → **19.2 s**, p50 decision latency **765** → **388 ms**, **5k fewer tokens** per turn
- Solved fast/slow model sync: a fast model acts on screen while the main model writes its plan. Built the sync layer that skips repeated steps, drops plans made stale by wrong steps, and verifies every send or delete before it runs. Fastest end-to-end task **27 s** → **19 s**
- Built long-horizon memory for multi-step tasks in creative and CAD apps: working context in memory, task state checkpointed to disk, so tasks survive context-window limits
- Ported the multi-agent layer into a dictation app currently used by 30 beta users, as a standalone **Rust** binary, optimized for speed and disk footprint

ML Engineer Intern | Reliance Industries Ltd. *Dec 2025 – Mar 2026*
Navi Mumbai, On-site

- Fixed the PDF-parsing microservice (OCR + local LLM) for the internal Letter-of-Credit (trade finance) scrutiny tool: **72 s** → **23 s** per document, **3,000+ documents a day** for **230 sales executives** across India
- Salt production forecasting (XGBoost) on **10 years** of data across **6 regions**: **27% more accurate** than baseline, served via FastAPI with per-region dashboards and alerts for production planning

Full-Stack Developer (Freelance) | Brydgework *Apr 2026 – May 2026*
New Delhi, Remote

- Co-built a donation platform sized for **4k monthly active users**: Next.js, JWT auth, payment gateway
- Owned deployment: Dockerized on a VPS behind Nginx (reverse proxy, SSL termination)

Projects

qwendoom: the real Doom, running inside a language model *github.com/utsav1033/qwendoom*

- Ran the real Doom engine on a RISC-V CPU compiled into the weights of a Qwen3.5-style model: every generated token is one CPU step, and the whole game lives in the model's memory. No OS, no disk
- Grew the machine's memory **32 KB** → **12 MB** (11 → 328 layers) and ported Doom to it with a custom libc; a C emulator of the same machine matches native Doom pixel for pixel
- Wrote an exact C stepper that skips weights proven unable to change the next token: **79** → **3,700 tokens/s (47×)**. The model booted Doom and drew **41 frames** over **76M tokens**, **0 mismatches** against the reference CPU

bytesized: a 2,946-byte LLM engine *github.com/utsav1033/bytesized*

- C engine with no libc that runs SmolLM2-135M from the original Hugging Face files; the binary is **2,946 bytes**, small enough to fit in one QR code
- Output matches Hugging Face transformers **token for token** (3 prompts × 100 tokens); own BPE tokenizer, RoPE, grouped-query attention and KV cache
- Shrunk **67,976** → **2,946 bytes (23×)** in 18 steps, each re-verified: raw syscalls, custom linker script, and a self-unpacking binary with a 348-byte arm64 assembly unpacker

carmen: AI-written GPU kernels, proven by a judge *github.com/utsav1033/carmen*

- An LLM writes Apple Metal kernels; a deterministic judge checks them against a float64 answer key on hundreds of inputs (incl. hidden ones) and times them against MLX
- Fused 4-bit MLP kernels: **8 MLX kernels** → **2** per layer of Qwen 2.5 0.5B, decode **3% faster** (95% CI 2–5%, 30 paired runs) with identical output; up to **1.82×** MLX on single kernels
- Judge caught **75/75** planted bugs vs **40/75** for a KernelBench-style one-shape check, and exposed 2 of my own false speedups

Skills

Languages	Python, C, Rust, Swift, TypeScript, Java, SQL
AI / Inference	Metal, MLX, vLLM, llama.cpp, Hugging Face Transformers
Backend	FastAPI, Spring Boot, Next.js, Angular
ML	PyTorch, TensorFlow, NumPy, scikit-learn, XGBoost, OCR
Databases	SQLite, Postgres, MySQL, Redis
Infra	Linux, arm64 / RISC-V, Docker, AWS, Azure DevOps, GitHub Actions, Nginx, CI/CD

Education

B.Tech in Computer Engineering *2022 – 2026*
Pillai College of Engineering, University of Mumbai

Publications

Akshara: Contextual Transliteration of Brahmi Using Deep Learning *ieeexplore.ieee.org/document/11437445*
ICNTE 2026, IEEE

- Brahmi → Devanagari → English: OCR, a deep learning transliterator trained on **22,000+** samples of **272 characters**, and a Bi-LSTM translator. Presented Jan 2026, published in IEEE Xplore