

# Pratham Kode

Language Models from Scratch | PyTorch | Distributed Training

[kodopratham@gmail.com](mailto:kodopratham@gmail.com) | [huggingface.co/prathamkode](https://huggingface.co/prathamkode) | [github.com/kodopratham](https://github.com/kodopratham) |  
[linkedin.com/in/pratham-kode-94291a282](https://linkedin.com/in/pratham-kode-94291a282) | [x.com/prathamkode](https://x.com/prathamkode)

## EDUCATION

### Vishwakarma Institute of Technology

Pune, India

*B.Tech in Computer Science and Engineering*

*Expected May 2028*

- Coursework: Data Structures & Algorithms, Operating Systems, Computer Networks, Distributed Systems, DBMS, OOP

## EXPERIENCE

### Particle

Remote

*Founder — from-scratch language models ([particle-1.0](#), [1.6](#), [2.0](#))*

*2026 – Present*

- Built a 109.5M Llama-style decoder from random initialization (12 layers, 768 hidden, 12 heads; RoPE, SwiGLU, RMSNorm, tied embeddings) — not a fine-tune of a public checkpoint.
- Trained a custom 32k byte-level BPE tokenizer on FineWeb-Edu, then pretrained on ~2B FineWeb-Edu tokens (context 2048, bf16).
- Ran the full pipeline on AWS Spot with SkyPilot: tokenize on CPU Spot, pretrain and SFT on `g6e.12xlarge` (4× L40S) with FSDP, AdamW, cosine LR, and S3 checkpoints; end-to-end train budget ~\$40–70.
- Supervised-fine-tuned for chat, converted weights to Hugging Face `LlamaForCausalLM`, and released MIT `bf16` checkpoints on the Hub.
- Shipped [particle-2.0](#): continued pretraining plus SFT on a new data mix over the same 109.5M architecture; the most-downloaded Particle release.

### Cantiffin

Remote

*Founder & Lead Full Stack Developer*

*June 2025 – August 2025*

- Shipped [cantiffin.com](https://cantiffin.com), a food-ordering platform used by 70+ people, with Node.js, PostgreSQL, and WebSocket order tracking to cut queue wait.
- Set up GitHub Actions → Vercel CI/CD for the web app.

## TECHNICAL SKILLS

**Languages:** Python, C++, TypeScript, JavaScript, SQL, Bash, Solidity

**ML / LLMs:** PyTorch, Transformers, tokenizers, FSDP, SFT, RoPE, SwiGLU, RMSNorm, ONNX Runtime, RAG, LangChain

**Training infra:** AWS (Spot EC2, S3), SkyPilot, Docker, GitHub Actions, Hugging Face Hub

**Software:** React, Next.js, Node.js, PostgreSQL, REST APIs, WebSockets, Git

## PROJECTS

**Smartwatch-LM** | *PyTorch, BPE, ONNX Runtime* | [v0.1](#), [v0.2](#)

2026

- Trained tiny GPT-style LMs from scratch for on-device smartwatch assistants: `smartwatch-lm-0.1` (6 layers, ~12M params) and `smartwatch-lm-0.2` (~15.4M params, 5.5k BPE vocab, best val. loss 0.324).
- The model replies in natural language and emits parseable intent tags (35 intents, e.g. `<INTENT:GET_STEPS>`) plus slot placeholders that the host app fills from sensors; exported to ONNX (opset 17, ~60 MB) with a 39-prompt golden benchmark.
- Paper accepted at **NLPiR 2026** (Nara, Japan; Springer LNNS).

**DolphinAI** | *Python, LangChain, Pinecone / Supabase*

2025

- RAG chatbot over marine data; AI Genesis finalist (top 10 / 200+). Sub-200ms retrieval on a vector index.

**TruePaper** | *Solidity, Web3.js, IPFS*

2025

- Ethereum dApp for immutable academic certificates (on-chain hash + IPFS payload).

## ACHIEVEMENTS

---

**Uncanny Hackathon** winner — Delhi, 2026.

**AI Genesis** finalist — DolphinAI, top 10 of 200+ teams.

**Monad Blitz** runner-up — Monoken, a blockchain ticketing app.

Published a Unity/C# game to Google Play at 15.