

Abishek Puri

🏠 San Marcos, TX | 📞 (512) 667-2072 | ✉️ abishekpuri77@gmail.com | 🌐 LinkedIn | 🐙 GitHub

Education

Bachelor of Science in Computer Science & Data Analytics

Class of 2028

Texas State University

GPA: 3.7, Dean's List, Honors College

Relevant Coursework: Data Structures & Algorithms, OOP, Computer Architecture, Software Engineering, Applied Linear Algebra

Technical Skills

Languages: Python, Java, C++, JavaScript, TypeScript, HTML/CSS

Frameworks & Libraries: FastAPI, Next.js, React, Node.js, Flutter, Three.js

AI / ML: OpenAI API, RAG, FAISS, OpenCV, sentence-transformers, Pandas

Databases & Cloud: PostgreSQL (Supabase), Firebase Auth, Vercel, Render, Git/GitHub

Concepts: System Design, Backend Development, API Design, CI/CD, Agile

Experience

Research IT Assistant

Nov 2025—Present

Texas State Division of Research

- Built and deployed a full-stack FAQ chatbot (FastAPI + JavaScript widget) handling 50+ live research queries, embedded site-wide via a single `<script>` tag.
- Architected REST API endpoints with input validation and structured error handling to mitigate injection-style vulnerabilities and improve response reliability.
- Owned the full release cycle across backend (Render) and frontend (Vercel) with configurable CORS and API base URL for multi-department reuse.

Projects

Astralecture—Interactive 3D Solar System

Live

Three.js, JavaScript, Vite, Web Audio API, Speech Synthesis API

- Built a real-time 3D web app rendering all 8 planets with accurate orbital motion at 60 FPS and a custom-shaded Sun.
- Engineered a cinematic tour mode with Speech Synthesis API narration and Earth-comparison overlays; deployed to Vercel.

PuriPy—Time-Travel Debugger for Python

GitHub

Python, sys.settrace, Textual, msgpack, Gemini API

- Built a record-and-replay debugger that captures every variable mutation, function call, and return value during Python execution using `sys.settrace`; implemented delta encoding to store only changed variables per frame, reducing trace size by 9x vs naive serialization.
- Designed a serialization pipeline handling arbitrary Python objects via pickle with `repr()` fallback, and pre-computed a line index and call-depth table enabling O(1) frame navigation across the full execution history.
- Integrated a natural-language query layer using Gemini API to answer questions like "why is this variable None?" by extracting relevant frames from the trace as context.

RAG Knowledge Chatbot—Document Q&A System

GitHub

Python, OpenAI Embeddings, FAISS, Streamlit

- Built a RAG pipeline ingesting PDF/TXT/MD documents, embedding chunks with `text-embedding-3-small` into a FAISS vector index for semantic search.
- Implemented similarity-threshold gating and source-chunk previews for full answer auditability, reducing hallucinated responses.