

Muhammad Abdullah

mabdullahashfaq433@gmail.com | +923334523377 | 03/04/2003 | Lahore, Pakistan. | github.com/R4333 | linkedin.com/in/muhammad-abdullah-795672110 | abdullah-portfolio-gamma.vercel.app

PROFILE

Highly motivated Full Stack Software professional building AI Powered products end-to-end. I have worked on everything from the frontend down to deployment. With a strong foundation in programming languages such as C++, Python, Typescript, and JavaScript, as well as experience with popular web frameworks and libraries like Nextjs, Reactjs, Expressjs and Nodejs. Lately my professional focus has been on developing AI infused Full Stack applications. I am eager to contribute my skills to your projects and gain valuable experience from it.

PROFESSIONAL EXPERIENCE

Full Stack Engineer

06/2025 – Present

Webify Ventures - Idraak Labs

Lahore, Pakistan

- I worked on a wide variety of projects from several clients and took on a vast amount of roles ranging from Full Stack Development to DevOps.
- A few projects that I worked on:
- ConversaIQ - Conversation intelligence for sales and support agents.
- Thalamus - The Enterprise RAG Solution for Document intelligence.
- Aletheia - An AI tool to help Legal professional analyze Court Depositions through Video and Audio recordings.
- First Rule Contract Manager - Construction Contract review automation for construction teams with help of AI.
- Tabsy App - Productivity and note-taking application that leverages Local AI Integration and integrates with popular apps like Slack and Discord.
- ResumeCraft - A Resume creation app with AI Integration.
- AI Voice Assistant for Even Realities G2 Smart Glasses.
- Lyfe in Color - An image generation app that turns family photos into coloring books.

Full Stack Web Developer

09/2023 – 01/2024

Freelance Web Developer

Lahore, Pakistan

- Developed Indian Runners, a NEXT.JS based fitness web application.

MERN Stack Intern

07/2023 – 08/2023

Coding Pixel

Lahore, Pakistan

EDUCATION

Bachelor of Science (Computer Science)

2021 – 2025

FAST NUCES Lahore

Lahore, Pakistan

A-Levels

2019 – 2021

LGS Paragon

Lahore, Pakistan

- Maths, Physics, Chemistry, Computer Science.

O-Levels

2017 – 2019

Laurelbank Public School

Lahore, Pakistan

- Maths, Physics, Chemistry, Computer Science.

SKILLS

Languages: C++, C, JavaScript, CSS, HTML, Python, Typescript, SQL

Frameworks: NEXT.JS, ExpressJS, Node.js, REACT

Databases: Postgres, SQLite, MongoDB

Libraries/Tools: Raylib, SFML, ImGui, SDL2, VSCODE, Visual Studio, MAKE/CMAKE, Linux, WSL, Jupyter Notebook, Docker, Proxmox

LANGUAGES

Languages: Urdu, Punjabi, English

Resume Intel

- Full-stack AI-powered resume and cover-letter platform that parses uploaded resumes, analyzes job postings, and tailors content using LLMs, with server-side PDF export and chat-driven DOCX editing.
- Built a Next.js 16 (App Router) web app with React 19 for AI-assisted resume building, cover-letter generation, and DOCX tailoring.
- Integrated OpenAI GPT-4.1 via the Vercel AI SDK for resume parsing, job-description analysis, AI rewriting, and multi-turn chat editing.
- Implemented server-side, print-grade PDF export using Puppeteer + headless Chromium, and DOCX parsing/editing with Mammoth, JSZip, and xmldom.
- Designed a PostgreSQL data layer with Drizzle ORM and migrations; added email authentication, password reset, and rate limiting via Better Auth and Resend.
- Engineered public shareable resume pages, template system, completion scoring, and user dashboard with version history.
- **Technologies:** TypeScript, Next.js 16 (App Router, Server Actions), React 19, Tailwind CSS v4, Radix UI/shadcn, PostgreSQL, Drizzle ORM, Vercel AI SDK, OpenAI (GPT-4.1), Puppeteer/Chromium, Mammoth/JSZip, Better Auth, Resend, React Query, React Hook Form, Zod.

An Internal RAG and Document Search project for Deloitte

- Built an enterprise RAG platform that lets consultants search and chat across internal knowledge silos (SharePoint, Confluence, Azure files, ServiceNow) with natural language.
- Created an ingestion pipeline that chunks documents, generates embeddings (text-embedding-3-large), and maintains a vector index in Azure AI Search.
- Integrated Azure OpenAI (GPT-4.1) for grounded answers with citations, Microsoft Entra ID for per-document access control, and a streaming React/TypeScript chat UI.
- Deployed serverlessly on Azure Container Apps, fully defined as IaC (Bicep).
- **Technologies:** Python, FastAPI, LangChain, Azure OpenAI, Azure AI Search, Microsoft Entra ID, SharePoint REST API, Microsoft Graph, React, TypeScript, Docker, Azure Container Apps, Bicep.

Thalamus | *Enterprise RAG for Document Intelligence*

- Built the Python backend for a multi-tenant RAG platform that answers natural-language questions from uploaded document sets with full citation.
- Engineered a retrieval pipeline using LlamaIndex for ingestion, chunking, embedding (OpenAI), hybrid vector search (Pinecone), and cross-encoder re-ranking.
- Integrated multiple LLM providers (GPT-4.1, Claude) via a unified factory layer with structured output parsing for tables, key-value extraction, and report generation.
- Designed an async document ingestion pipeline (PDF, DOCX, OCR for scans) with incremental indexing, metadata extraction, and dead-link pruning.
- Exposed a REST API (FastAPI) consumed by the Next.js frontend; streamed chat responses token-by-token and enforced tenant/role-based access.
- Deployed on AWS ECS with S3 for document storage.
- **Technologies:** Python, FastAPI, LlamaIndex, OpenAI, Anthropic, Pinecone, PostgreSQL, Redis, Docker, AWS (ECS, S3).

Aletheia | *AI-powered legal deposition analysis platform*

- AI-powered legal deposition analysis platform that processes video/audio to detect deception, inconsistencies, and behavioral cues.
- Built an async multi-stage AI pipeline analyzing deposition video/audio for tone, response quality, consistency contradictions, and evasiveness indicators.
- Engineered audio prosody feature extraction (pitch, RMS, zero-crossing rate, spectral centroid, speaking rate, filler-word frequency) using librosa/numpy, with per-speaker baseline tone classification.
- Integrated multiple LLM/AI providers – OpenAI GPT-5.2 (structured JSON outputs via Pydantic schemas), Google Gemini (multimodal video analysis: facial expressions, gaze, head pose), and AssemblyAI (diarized transcription & speaker roles).
- Architected a FastAPI REST API with Celery + Redis background workers, supporting job queuing, stage-by-stage progress tracking, streamed logs, and run-artifact persistence.
- Built media processing via ffmpeg (audio extraction, chunking) and a vanilla JS web UI for monitoring jobs.

- **Technologies:** Python, FastAPI, Celery, Redis, Uvicorn, OpenAI API, Google Gemini (genai), AssemblyAI, librosa, NumPy, Pandas, Pydantic, ffmpeg, REST, JavaScript/CSS/HTML, uv.

Lyfe in Color

- Full-stack web app that converts user photos into personalized, printable coloring-book pages and cover art, then lets users arrange, review, and order finished books.
- Built a React + TypeScript SPA with a guided multi-step creation workflow (upload, theme selection, streaming generation progress, failure recovery, cart, and checkout).
- Designed an Express/Node backend integrating the OpenAI image API with streaming (SSE), batch concurrency, retry/backoff logic, safety-error classification, and per-job latency metrics.
- Implemented Supabase authentication and data layer, plus an admin dashboard for orders, themes, and galleries.
- Deployed on Railway via Nixpacks.
- **Technologies:** React, TypeScript, Vite, Tailwind CSS, shadcn/ui (Radix UI), TanStack Query, React Router, React Hook Form, Zod, Framer Motion, Recharts, dnd-kit, Node.js, Express, OpenAI API (streaming/batch), Supabase, Railway, Vitest.

ConversaIQ | *Cross-platform desktop application*

- A cross-platform desktop application that listens to live sales/customer calls, transcribes them, and provides real-time AI-driven coaching, deal analysis, and conversation summaries.
- Live transcription of both microphone and system audio (speaker) with speaker diarization.
- Real-time AI co-pilot chat – ask questions mid-call and get suggestions.
- Live conversation analysis – sentiment, objection handling, intent detection, and a "close probability" score.
- Auto-generated summaries with TL;DR, bullet points, and action items.
- Session history with searchable, renameable past conversations.
- Customer attachment – link conversations to customer records (CRM-style).
- Drag-and-drop resizable panel UI (transcript / analysis / chat layout).
- Multi-provider support with pluggable AI factory: OpenAI, Google Gemini, Deepgram, FireworksAI (LLM + STT).
- macOS system audio capture via System Audio Dump (feat/MacOS_support).
- Autonomous AI Voice Agent mode with bidirectional WebSocket audio streaming to Deepgram Voice Agent API and virtual audio routing (feat/Auto_Agent).
- Cross-platform audio engine (24 kHz PCM, chunked IPC streaming).
- Offline-first local persistence with migrations + sync-state tracking.
- **Technologies:** Electron, React 18, TypeScript, Vite, Tailwind CSS, shadcn/ui, Radix UI, Recharts, React DnD, react-resizable-panels, TanStack React Query, React Router (hash router), react-hook-form + Zod, better-sqlite3 (Repository Pattern), electron-store, OpenAI SDK, @google/genai, Deepgram STT/Voice Agent, FireworksAI, ws (WebSockets), Electron IPC secure bridge (context isolation, no nodeIntegration), electron-builder, ESLint, marked + DOMPurify (safe markdown rendering).

AI Voice Assistant for Even Realities G2 Smart Glasses

- Built a full-stack AI voice assistant streaming live glasses-microphone audio to OpenAI and rendering answers on a HUD.
- Engineered a realtime voice pipeline (WebSockets, custom PCM 16kHz 24kHz resampling) bridging the glasses mic to the OpenAI Realtime API with streamed transcripts/answers.
- Integrated OpenAI Responses, Transcriptions, and Realtime APIs with session continuity.
- Built the gesture-driven HUD UI (Up/Down/Click navigation, per-screen layouts) via the Even toolkit glasses router.
- Developed a React companion app and a hardware-in-the-loop simulator launcher.
- **Technologies:** React 19, TypeScript, Vite, Tailwind CSS, Node.js, Express 5, WebSockets, OpenAI SDK (Realtime/Responses/Transcriptions APIs), PCM audio processing, Even Realities G2 SDK & toolkit.

AI Voice Customer Agent Platform

- A full-stack platform that powers real-time, phone-based AI voice agents. Incoming PSTN calls are bridged to a realtime LLM (OpenAI/Deepgram) with live speech-to-text, text-to-speech, and voice activity detection – plus an admin dashboard to manage calls, transcripts, prompts, tools, and settings.
- Built a realtime voice-agent platform bridging Twilio phone calls to OpenAI Realtime / Deepgram over WebSockets, handling bidirectional μ -law audio streaming and G.711 transcoding.

- Designed a pluggable provider layer (OpenAI, Deepgram) with configurable VAD (server/semantic), TTS voices, and multi-language support.
- Implemented an admin dashboard for live call monitoring, transcripts, recordings, prompt templates, and MCP tool integration for function-calling.
- Engineered the backend with Fastify + WebSockets, Drizzle ORM/SQLite persistence for calls/transcripts/agents/settings, and encrypted API-key storage.
- Set up streaming media orchestration between Twilio Media Streams and AI providers with audio listener fan-out for recording/monitoring.
- **Tech stack:** React 19, TypeScript, Vite, TailwindCSS v4, Radix UI, React Router, Sonner, Node.js, Fastify, WebSockets (ws), Drizzle ORM, SQLite/libSQL, Pino logging, tsx, OpenAI Realtime API, Deepgram, Twilio (Voice/Media Streams/Recordings), MCP (Model Context Protocol), VAD, pnpm workspaces (monorepo), Docker, Nixpacks, ngrok, ESLint.

MagarMunch: An Online Food Ordering and Delivery System

- Uses NEXT.JS with REACT components for Restaurant Website.
- Backend REST API implemented using Node.js, ExpressJS, and MSSQL.

Building Healthcare Solutions

- A full-stack marketing & e-commerce platform for an indoor-air-quality company (Saniuvent vent-stack treatment technology). Built the customer-facing site, secure checkout, admin order management, and a transactional-email infrastructure.
- Developed a full-stack web platform for Building Healthcare Solutions, a clean-air technology company.
- Built a marketing site (product catalog, interactive US installer map, testimonials) with lazy-loaded routes.
- A secure e-commerce checkout processing credit-card and ACH payments.
- A role-based admin dashboard for order management.
- A serverless email system with queue processing, unsubscribe, and suppression-list handling.
- Designed the Postgres schema and Supabase Edge Functions for payment authorization and email automation.
- **Technologies:** React 18, TypeScript, Vite, Tailwind CSS, shadcn/ui (Radix UI), React Router, TanStack React Query, React Hook Form, Zod, Supabase (Postgres, Auth, Edge Functions/Deno), Authorize.net (Accept.js, eCheck ACH), Recharts, Embla Carousel, Lucide, Vitest, Testing Library, ESLint.

GRPO-MATH LLM

- Qwen 2.5 3B trained using Deepseek's GRPO Reinforcement Learning technique to turn it into a Reasoning model for Math Problems.

FactFinderLLM

- R&D Project to reduce hallucinations in LLM responses on a given dataset using a custom multistage RAG pipeline and model finetuning. Demonstrated on Wikileaks and FOIA request data and involved creating a Synthetic Dataset using LLMs.

Tabsy App | *AI-Powered Note-Taking with Local LLM & RAG*

- A cross-platform Electron desktop app that combines rich note-taking with on-device AI and integrates seamlessly into Slack and Discord workflows.
- Integrated local AI via llama.cpp, allowing users to download and run any Hugging Face model (GGUF) for private summarisation, chat, and idea expansion directly inside notes.
- Built a local RAG pipeline that indexes all user notes into a vector store (Chroma) using Nomic Embed Text v2 for embeddings, enabling semantic search and retrieval-augmented answers across the entire knowledge base.
- Added Slack & Discord integration: import threads as structured notes, export meeting summaries, and trigger note creation with slash commands using the respective APIs.
- Implemented a drag-and-drop resizable panel UI (notes, AI chat, integrations) with real-time markdown preview and full-text search backed by SQLite.
- Designed secure Electron IPC (contextBridge) to isolate AI processing and file system access from the renderer; packaged with electron-builder for macOS/Windows.
- **Technologies:** Electron, React, TypeScript, Tailwind CSS, SQLite, Chroma, llama.cpp, Nomic Embed Text v2, Slack Bolt API, Discord.js, Node.js, electron-builder.

Costco Tracker Control Tower

- Web app replacing a manual Excel/VBA workflow for sales order tracking and multi-artifact reporting.

- Built a Next.js web app that replaced an Excel macro workflow, letting users enter an order once and auto-generating 5 downstream artifacts (Sales WIP, Photoshoot Log, Promo Tracker, Global Program Tracker, Biz Recap decks).
- Implemented canonical data model + projection logic to derive tracker-specific views from one source of truth (Zod validation).
- Added an approval inbox + notification event log with Power Automate-ready contracts.
- Engineered Excel/PPTX export pipelines (SheetJS xlsx, pptxgenjs) writing to a versioned output store.
- Designed Microsoft Power Platform handoff assets – Dataverse schema, SharePoint library contract, and Power Automate flow templates for production migration.
- **Technologies:** Next.js 16, React 19, TypeScript, Vitest, ESLint, Zod, SheetJS (xlsx), pptxgenjs, date-fns, Microsoft Power Platform (Dataverse, SharePoint, Power Automate, Entra ID), Office Scripts.

Simple Simple C Programming Language

- Custom Compiler written with FLEX, Bison, and LLVM.
- Compiles code to C++ and LLVM IR. Language had many advanced features added to it; mostly for Array manipulation.

Traffic Simulator

- Implemented using C++, SFML and ImGui.
- Tile based editor to allow users to create custom maps with simulated traffic signals and associated mechanics.
- Implemented path finding using A* algorithm.
- Viewport with zoom scaling.

SMART POS | *A NEXTjs based Point of Sale System*

- Implemented with Typescript and TailwindCSS.
- Postgres and Drizzle ORM.
- Authentication with Lucia.

Hide Stream | *A Steganography Tool*

- Lets you hide files in audio files and pictures using several different techniques.

Pac-Man in Assembly

- Implemented in pure x86 NASM assembly with interrupt handlers, custom ISRs and interrupt chaining for hooking timer and keyboard input. Load maps from files and interfaces with video memory.

AI-based Subtitle Maker

- Uses ffmpeg and Faster-Whisper Speech to Text Model.

Frontend for OpenAI GPT-3

- Personal Project. A frontend I implemented for OpenAI GPT-3 API when I got access to the developer preview. Primarily written as an internal content creation tool for a short-lived personal entrepreneurial venture by the name of Wordiosa.org. We provided copywriting and written content creation services to our clients. We closed the business due to lack of time on our part and then soon ChatGPT came along.

Process Scheduling Simulation | *Project for Operating Systems Course*

- Implemented in pure JS, HTML and CSS using Bootstrap and Chart.js libraries.
- Rate Monotonic, and Earliest Deadline First Algorithms.

2048 Game

- Implemented in C++ in CLI.

Website for Wordiosa.org | *A personal copywriting and content creation venture*

- A simple site made using WordPress and Elementor.

INTERESTS

Gaming Modding Coding Radios/RTL-SDR IoT SciTech Silicon Enthusiast