Soham Rajesh Kayal

linkedin.com/in/sohamkayal/ | kayalsoham61@gmail.com | 831-346-8532 | github.com/Sohamkayal4103 | LeetCode Profile | Portfolio

EDUCATION

University of California, Santa Cruz *06/2024 - 06/2026*

Master's Degree in Computer Science **GPA: 3.96 / 4**

- Artificial Intelligence **(A+)**, ML, Analysis of Algorithms (DSA) **(A+) - topper** (Been a TA), Cryptography **(A+)** (TA & Research), Advanced Operating Systems (A), Approximation Algorithms (A), Computational Models(A)

University of Mumbai *08/2020 - 06/2024*

Bachelor's Degree in Engineering (***Major*** - *Information Technology,* ***Minor*** - *Blockchain Technology*) **GPA: 3.7 / 4**

- Adv. Data Structures & Algorithms (A), Operating Systems, Computer Networks, App Development, DevOps, Big Data, Cybersec, System Design, Data Science, Big Data, Data Mining

EXPERIENCE

Microsoft – Microsoft AI Development Acceleration Program *New England, 06/2025 - 09/2025*

Software Engineer

TypeScript, JavaScript, Azure, Python, Rust

- Working on improving agentic worflows in Microsoft Teams. Currently copilot answers using the complete meeting trasncript sent for grounding which uses a lot of tokens.
- Converting the meeting into the knowledge graph and feeding it to copilot for grounding.
- Resulted in improved precision nearing 100% and Recall upto 93% which is better then the trasncript approach and resulted in 73% of COGS savings (per 20 mins meeting and the savings scale as the meeitng time increases).

Amazon *Seattle, 06/2025 - 09/2025*

Software Development Engineer Intern

TypeScript, React, Java, jUnit, Spring Web MVC, AWS CDK, CloudWatch, Spring Boot, Redux, Apollo, Horizonte, Cline, Docker

- Single-strategy manual allocation in Apply Funds workflow in Amazon Business caused inefficiencies and high customer tickets.
- Designed and shipped a cross-service payments API; secured critical paths via custom interceptors to prevent CSRF vulnerabilities **(P99 <200ms)**.
- Automated single-strategy manual allocations; reduced support tickets by **50%** and saved **6-8 weeks** of developer time by decoupling workflows.
- Revamped frontend architecture using **TypeScript, React, and Redux**; simplified state management and reduced UI regressions.
- Conducted rigorours testing **(unit, integration, stress, beta, bar-raiser checks, etc.)**; implemented system health monitoring (CDK metrics, Alarms, Dashboards) in AWS CloudWatch to ensure high availability and **faster rollbacks** for on-call engineer during Sev3/Sev2/Sev1.

University of California, Santa Cruz *Santa Cruz, 01/2025 - Present*

Graduate Student Researcher (Cloud Security)

C++, Golang, Node.js, Google Cloud APIs, React, Supabase (SQL), Google APIs (OAuth, Google Drive APIs)

- Cloud encryption schemes used everywhere like **AES offer only computational security** and remain vulnerable to **quantum attacks**.
- Developed an encryption algorithm immune to quantum attacks based on the research paper (Link); optimized and scaled it for real-time encryption.
- Developed a C++ service for high-entropy initial key generation and encryption; also built a go microservice for parallel final key generation.
- Built a **Node.js/Express** backend orchestrating frontend with all the services and user and document metadata management.
- Used React and CSS in frontend; supabase for storing user and document metadata; achieved the speed of **less than 15ms for 150kb of files**.

Star Union Dai Ichi Life Insurance Company Limited *03/2023 - 12/2023*

Software Engineer Intern

Flask (Python), TypeScript, Next.js, React.js, Tailwind CSS, SMTP, MySQL Workbench, GitHub Actions, Docker

- The existing platform for managing procurement needs was old, inefficient and buggy; worked on developing a new procurement platform.
- Developed **scalable REST APIs** using Flask and SQLAlchemy; modeled complex workflows for insurance marketing teams.
- **Optimized MySQL database schemas and indexing; improved query performance** for analytics dashboards; **automated CI/CD pipelines via GitHub Actions**; established rigorous tests to prevent production regressions; migrated legacy views to Next.js, improving page load speeds **(approx. 30%)** and user experience for internal teams. The platform was adpoted for all zonal offices across India **(5000+ orders per week).**

3XP - SKALE Ecosystem ***01/2024 - 03/2024***

Founding Engineer

Next.js/React, TypeScript, Node.js/Express, Solidity, AWS, Supabase (SQL)

- Developed the MVP for SKALE ecosystem within three months; built a full-stack app using Next.js, Express.js for backend APIs, and Supabase (SQL) as database; deployed using AWS.
- Engineered cost-efficient smart contract interactions by batching multi-step blockchain calls into **single-click flows**, reducing gas fees and improving UX for thousands of metaverse gamers.
- Responsible for bug-fixing, API optimization, and **scaling the platform** to handle concurrent users across the decentralized ecosystem.

RealmCast - Built at YC Hackathon (Turn Real life videos into a VideoGame) ***GitHub***

React, Vite, TypeScript, Node.js/Express, Google Gemini API, Three.js, Unreal Engine 5.7 (C++), Python, Pixel Streaming, Tailwind CSS

- Converting real-world spaces into playable 3D environments requires separate capture, modeling, level design, scripting, and engine integration, in short its slow and expensive.
- Built an AI system that turns room photos, video, or text prompts into fully playable 3D game scenes in **one pipeline**.
- Backend extracts frames, runs visual segmentation, and uses **Gemini** to synthesize structured scenes (layout, objects, NPCs, lighting, missions, interactions).
- **Three.js browser runtime (working end-to-end)**: room in, playable world out. **Unreal Engine 5.7 path (in progress)**: runtime spawning of props/NPCs/vehicles, third-person character control, and Pixel Streaming for high-fidelity browser viewing.
- One pipeline replaces the entire capture → modeling → level design → scripting → engine integration chain.

Mergers & Acquisitions Due Diligence Copilot - AI-Native Deal Intelligence System ***MCP Server || GitHub***

TypeScript, MCP Server (mcp-use), React, Python, OpenAI APIs, Supabase (SQL), Node.js/Express, Next.js

- Due diligence in mergers and acquisitions is expensive, fragmented across emails/spreadsheets/data rooms, and risks are often missed or surfaced too late.
- Built an AI-native copilot that creates governed workspaces for acquisition reviews; ingests evidence across **finance, legal, HR, tax, ops, security**.
- Uses **OpenAI** for full-document risk identification, executive-facing findings, and cross-source contradiction detection when documents make conflicting claims.
- Adds **deterministic checks** for governance changes, employee retention risk, and sector revenue decline in CSV data; scores risk by probability, impact, confidence, and diligence tower.
- Enforces **human approval gates** for high-severity findings; generates approved-findings reports, visual risk outputs, provenance chains, audit logs, and investment committee packages.
- Created an MCP server and integarted it with the official ChatGPT website as a tool available for public use.

PromoPilot - Multi-Modal Agentic System ***Link***

Model Context Protocol (MCP), Claude APIs, AWS (Bedrock, RDS, Lambda), LangGraph, LangChain, Next.js (TypeScript), Python

- Early stage startups do not have resources for hiring a complete marketing team to advertise their product.
- Built MarketGenie, an AI marketing agent to help small businesses generate ready-to-post social media campaigns.
- Orchestrated 4 AI agents via **Model Context Protocol (MCP)** and AWS Bedrock to automate marketing campaign planning.
- **Agent 1**: gathers & analyzes competitor data; **Agent 2**: analyzes customer sentiments by reading product feedback; **Agent 3**:generates audio/video based ads using **AWS Nova and Polly**; **Agent 4**: posts on social media and replies to comments and improves in next campaign iteration
- Developed frontend using Next/React and CSS, used Node/Express for backend, **AWS models and Claude APIs for intelligence**, Auth0 for authentication, **AWS RDS as database**, twitter apis for posting/commenting and **reduced cost and campaigning time from days to a few hours**.

Agentic Event Orchestrator - Sponsor Prize Winner at Agentic Orchestration Hack (SF) ***Link***

Model Context Protocol(MCP), RAG, Node.js, Python, React (TypeScript), Anthropic Claude API, Google Calendar API, LangGraph, LangChain

- Being in Bay Area means drowning in tech events and hackathons and you need to know which events are worth your time.
- Built an agentic system **(3 AI agents) using Claude APIs for intelligence**, and a **MCP server for orchestrating the agents** to solve the problem.
- Implemented **Retrieval-Augmented Generation (RAG)** to synthesize insights from Luma/Devpost and cross-referenced **Google Calendar APIs** for scheduling. **The strategy agent** analyzes sponsor prizes for a hackathon against **GitHub & LinkedIn** and generates tailored project ideas.
- **The prototyping agent** reads sponsor documentation and auto-generates working boilerplate code integrating multiple SDKs.
- **Saved several days of effort and cost (approx. 2-3 weeks, winnings of 300+$)** as the agent gives me winning boilerplate projects **within minutes.**

Fake Currency Detection - (Computer Vision) ***Link***

React.js, Flask, OpenCV, TensorFlow, Keras, ImageNet, YOLOv8

- Developed a **machine learning model** using **TensorFlow** and **Keras** to detect counterfeit currency from uploaded images, leveraging **transfer learning** with the **VGG16 architecture** pre-trained on ImageNet.
- Implemented **image preprocessing** techniques using **OpenCV**, and applied advanced methods including **data augmentation**, **early stopping**, and **dropout layers** to enhance model accuracy and prevent overfitting.
- **Deployed the model**, created a full-stack app with a **React.js** and **Python backend**; enabled real-time inference, imroved UX with **<500ms** latency.

Hotel Management System using Spring Boot as backend- **Github**

Java, HTML, CSS, JavaScript, Spring Boot, Spring Security, Hibernate, Thymeleaf, MySQL

- Developed a comprehensive Hotel Management System featuring room booking, pricing display, review system, and room type showcasing, along with **secure login functionality** for administrators and customers.
- Implemented a full-stack solution using **HTML, CSS, and JavaScript** for the **front-end**, coupled with **Spring Boot, Spring Security, and Hibernate** for the **back-end**, ensuring **robust performance and security**.
- **Integrated MySQL Workbench** for efficient data management and utilized **Thymeleaf** for **seamless server-side templating**, resulting in a dynamic and responsive user interface.

BondsBy: A Bonds Trading Website **Vercel Link || GitHub**

MERN (MongoDB, Express.js, React, Node.js), Auth0, Stripe API, nodemailer, JSON, Chakra UI

- Built trading platform in MERN; handled **100+ daily transactions** with clean API boundaries.
- Added portfolio tracking, real-time views, and risk tools; boosted engagement **25%**.
- Improved onboarding with Auth0; reduced login errors **25%** via smoother flows.
- Standardized UI with Chakra components; sped feature delivery **50%** while keeping UX consistent.
- Integrated Stripe payments + email automation; increased payment success **20%**.

Privacy-Preserving Graph Processing on Hardware Enclaves (Intel SGX) ***GitHub Link***

C++, Pointers, DSA (AVL Tree, Graphs, Unordered Map), Intel SGX, Linux (Unix), Git

- Prevented pattern leakage from AES-GCM outsourced graphs by running BFS, DFS, Dijkstra inside hardware enclave (Intel SGX).
- Exposed only $|V|$, $|E|$, query type, and response size; enforced deterministic traces via fixed memory flows.
- Used doubly-oblivious layouts and an Omix++ map; kept access patterns stable under queries.
- Achieved near-linear $\mathcal{O}\big(C\,|E|\log^2|E|\log\log|E|\big)$ with **<15%** overhead; extended to MST via Obfuscuro + Omix++.

Hardware Hierarchical Dynamic Structure (HHDS) - Associated with MASC-UCSC Research Group ***GitHub Link***

C, C++, Smart Pointers, Ownership & Lifetime Design, Concurrency (Thread Pools), Bazel, Bit-Manipulation, Packed Data Structures, Cache Optimization

- Designed and built a comparative benchmark suite in **C++23** evaluating the HHDS graph library against **Boost.Graph** and **LiveHD** across 20+ operations—node, pin, and edge insertion/deletion/lookup plus 9 traversal variants—scaling inputs from **1K to 5M nodes**.
- Characterized HHDS's three-tier edge storage and 32-byte cache-aligned node layout, measuring $\sim$**7**$\times$ **faster edge insertion** and near-memory-bandwidth traversal (**9 ns/op**) versus Boost.Graph.
- Empirically isolated an $O(N^2)$ **bottleneck** in Boost.Graph's adjacency-list node deletion, quantifying a **20,992**$\times$ **HHDS speedup** at 500K nodes (1.7 μs/op vs. 35 ms/op)—a result featured in the thesis's Future Work.
- Benchmarked HHDS's persistent, incrementally-patched topological cache (Kahn's algorithm), demonstrating **55–142**$\times$ **speedups over LiveHD** on hierarchical forward/backward traversal.
- Cut benchmark wall-time from $\sim$**70 hours to** $\sim$**40 minutes** by parallelizing the three-library suite and tuning input sizes, preserving per-operation ($\geq$1 s/cell) measurement validity.
- Authored a **14-page technical thesis** (LaTeX) with 10 figure groups and 33 vector plots; drove the report through **4 review revisions** with the faculty advisor.

AugmentFS: User-Space Integrity File System ***GitHub Link***

C++, FUSE, SQLite, Systems Programming, Data Integrity, ACID Transactions

- Built FUSE file system with embedded SQLite; added queryable metadata on top of legacy FS (ext4).
- Enforced WORM audit logs; used FNV-1a hashes (file + 4KB blocks) to detect silent corruption (bit rot).
- Streamed hashing to decouple DB transactions from write path; cut sequential overhead **2.4x to 1.4x**.
- Ensured crash consistency via WAL and copy-on-write (COW) updates; improved safe recovery on restart.

Decentralized Healthcare System **Demo || Paper**

TypeScript, Solidity, React.js, Ether.js, Particle Auth, Push Protocol, Huddle SDK, Chakra UI

- Built Web3 system for records, consultations, and appointments; reduced fragmented healthcare workflows.
- Added KYC onboarding for doctors; ensured verified experts and safer interactions.
- Shipped Solidity contracts; deployed to Sepolia and Scroll Alpha for testing and iteration.

DAOConnect: A DAO Management Platform **Demo || GitHub**

Solidity, Ethers.js, Next.js, ParticleAuth, Chakra UI, Optimism, Wagmi

- Built DAO governance + tokenomics platform; streamlined proposals, voting, and community workflows.
- Deployed Solidity on Optimism Sepolia; enabled low-cost on-chain actions for iteration.
- Integrated Particle Auth with ERC-4337; delivered Web2-style onboarding for wallets.
- Added quadratic voting + multi-token governance; improved fairness and richer governance models.

Aave-Anonify **Demo || GitHub**

TypeScript, Next.js, Chakra UI, Solidity, Ether.js, RainbowKit, Wagmi, NFTPort, GraphQL, Apollo Client, The Graph, Anon Aadhaar

- Built lending flow for artists using stablecoins (GHO, DAI) with NFTs as collateral.
- Integrated Aave v3; enabled efficient borrowing power and interest-aware asset handling.
- Added NFT marketplace for investor lending; aligned incentives with flexible repayments.

TechV3nt / TechX: Event Management Platform

React, Solidity, Ether.js, RainbowKit, Wagmi, NFTPort

- Built event platform with NFT ticketing; enabled verifiable ownership and transferability.
- Implemented wallet auth and on-chain mint/verify flows using Ethers.js, RainbowKit, Wagmi, NFTPort.

Netflix Clone iOS App (Native iOS) ***GitHub Link***

Swift, SwiftUI, UIKit, MVVM, REST APIs, Xcode, YouTube APIs, Gemini APIs, GCP

- Built iOS app for streaming in SwiftUI; used MVVM to isolate UI, state, and networking.
- Integrated TMDB + YouTube APIs; added trailers, search, and structured browsing flows.
- Used Swift Concurrency for responsive fetch and rendering; handled retries, caching, and errors.
- Added Gemini APIs for spoiler-free summaries and fan theories inside the app.

JARVIS Mobile - Built at Cactus + Gemma 4 Voice Agents Hackathon (YC HQ): On-Device Social Memory Agent for iOS ***GitHub***

Expo, React Native, Python, TypeScript, cactus-react-native (on-device inference), Gemma 4 e2b-it INT4, WeSpeaker ResNet34-LM INT4, Parakeet CTC, Segmentation 3.0, expo-sqlite, expo-audio, expo-file-system, expo-speech

- Remembering people - names, prior conversations, open follow-ups, what matters to them - is fragmented across notes, contacts, and messages, and existing AI memory tools require sending private audio to the cloud.
- Built a **mobile-first, fully on-device** social-intelligence agent: downloads **Gemma 4 (4.36 GB)** onto the phone, enrolls a primary-user voiceprint, captures live conversations, and builds a persistent local profile of every recurring person. **Runs with WiFi off - zero cloud dependency, zero raw audio leaves the device.**
- **Audio + perception pipeline:** 16 kHz mono PCM WAV captured in **8-second rolling chunks (5-minute cap)** via expo-audio; routed through Cactus React Native bindings to **Parakeet CTC** STT + **Segmentation 3.0** diarization + **WeSpeaker ResNet34-LM** speaker embeddings, all running on **Apple Neural Engine**.
- **Drift-correcting voiceprints:** cosine match against **speaker_profiles** table with tuned thresholds (**0.74** known-person / **0.70** session bucket); on each confirmed match the stored embedding is replaced by a **running mean of L2-normalized samples**, so identity sharpens with every interaction without requiring re-enrollment.
- **Gemma 4 memory loop:** prior summary + open follow-ups + recent relationship signals (tone, affinity 1-5, topics) are loaded into the analysis prompt; Gemma returns structured **JSON** (session summary, facts, follow-ups, signals) that is **merged, not overwritten**, into a **5-table SQLite schema** (people, speaker_profiles, sessions, session_turns, person_memories). **Answer Mode** supports normal/rizz/joke/roast personas; same on-device contract ports unchanged to native iOS / Android via Cactus.swift and Cactus.kt bindings.

LiveNPC - Built at YC Voice Agents Hackathon (Voice-Agent Framework for Game NPCs) ***Demo || GitHub***

Python, TypeScript, Pipecat (Pipecat Cloud), NVIDIA Nemotron-3-Super-120B (LLM) + Nemotron Streaming STT, Gradium (TTS), vLLM, Cekura, Twilio, AWS, Daily/WebRTC, RTVI, React Three Fiber, Three.js, Zustand, Vite, FastAPI, Pydantic, Docker, MCP

- In games like GTA 5, NPCs have fixed, pre-scripted dialog that loops forever, cannot react to player actions, and can never break out of the game (e.g., place a real phone call).
- Built **LiveNPC**, an **engine-agnostic framework** (Unity, Unreal, Three.js, Godot) that gives game NPCs real-time **voice conversations**, live scene reasoning, and in-world actions - solo, in **one day**.
- Engineered a streaming voice pipeline - **NVIDIA Nemotron-3-Super-120B** (LLM) + **Nemotron** streaming STT + **Gradium** TTS via **vLLM** - orchestrated with **Pipecat** over **WebRTC**; exposed NPC actions (draw weapon, step forward, flee, de-escalate) as **LLM tool calls** streamed back through the **RTVI** data channel.
- Designed a language-agnostic client/agent protocol (**TypeScript + Python/Pydantic**) and injected live game state into the LLM context each turn for action-aware, non-scripted dialogue; built the demo game in **React Three Fiber**.
- Deployed to **Pipecat Cloud** as one **Docker** bundle serving **WebRTC, Daily, and Twilio** from a single `create_transport` entrypoint; integrated **Twilio** to place **real outbound phone calls** to real numbers from inside the game.
- Built an automated **evaluation + self-improvement loop** with **Cekura** (8 metrics, 8 simulated-player scenarios; **6/6 in-character, 3/3 weapon-discipline** pass) and auto-tuned the agent prompt from eval failures; shipped a **<60s live demo**.

DevGraph - Built at YC AI Growth Hackathon (AI Growth Team for Developer Tools) ***Live || Demo || GitHub***

TypeScript, Next.js 16, React, Convex (reactive backend), Clerk, OpenAI (GPT-5 + web_search), Fiber AI, Node.js, GitHub API/SBOM, Devpost, Vercel

- Developer-tool companies and hackathon sponsors struggle to discover, acquire, and retain adopters - signups spike for a day or two, then most churn and rarely convert to paying customers.
- Built (solo) a **multi-tenant SaaS** that runs the entire growth loop end-to-end - discovery, research, acquisition, deals, retention, and turning feedback into shipped features - grounded in real, cited data (no generic LLM output).
- Engineered an **8-stage research pipeline run in parallel** on a **Convex** reactive backend (background actions streaming results live); grounded web stages with the **OpenAI Responses API web_search tool** so every claim carries real source URLs, cutting a **10+ min** sequential run to ∼**2 min** (under Convex's 600s action cap).
- Resolved real people and companies via **Fiber AI** (GitHub→LinkedIn identity + real-time Tracker signals for expansion/churn); detected SDK adoption from code via GitHub SBOM + Devpost scraping; scored leads **0-100**.
- Shipped **AI-generated offers + outreach** (button-only push to an Orange Slice webhook), a **feedback→features** loop for engineering/GTM teams, and a **Viral Launch-in-a-box** generator (multi-channel assets + transparent reach→CAC projection); deployed on Vercel.

Project Showcase ***GitHub Link***

React.js, Django, Material UI, React Bootstrap, Firebase

- Built searchable project discovery by department and mentor; simplified selection and navigation.
- Added interactive Q&A and filters; improved engagement and faster project matching.
- Planned scalability with predictive recommendations; structured data for future ranking models.
- Shipped React frontend + Django backend; used Firebase for auth and data storage.

BMI Calculator (iOS) ***GitHub Link***

Swift 5, Xcode, UIKit, Auto Layout, Core Graphics, Composable Architecture

- Built UIKit app in Swift; applied MVC structure for readable, maintainable code.
- Implemented Auto Layout + UILabel results; color-coded BMI categories for clarity.
- Added input validation and safe optional handling; reduced incorrect outputs and crashes.
- Used Core Graphics for custom UI elements and simple animations.

Quiz App (iOS) ***GitHub Link***

Swift, Xcode, UIKit, Core Data, JSON parsing, Concurrency

- Built quiz UI in UIKit; structured question bank with Codable and JSON updates.
- Persisted progress and high scores using Core Data; enabled durable local state.
- Implemented safe state transitions and validation; prevented corrupted scoring paths.

SKILLS

Programming Languages: Swift, Objective-C, C, C++, Go, Java, Python, TypeScript, JavaScript, Solidity, Rust
Libraries & Frameworks: Spring Boot, Spring Web MVC, Node.js, Express.js, React.js, Next.js, Flask, Django, Angular, Tailwind CSS, Chakra UI, UIKit, Redux
Database Management: MySQL (SQL), PostgreSQL, MongoDB (NoSQL), Firebase, Supabase, Redis
Tools: Xcode, Git, GitHub, Amazon Web Services (AWS), Google Cloud Platform (GCP), Docker, Kubernetes, Terraform, AWS CDK, CloudWatch, GitHub Actions, Linux/Unix
Other Technical Skills: Data Structures & Algorithms (DSA), Problem-Solving, System Design, Web Development, REST API, GraphQL, SDLC, CI/CD, Agile, Test-Driven Development, DevOps, Web3, OpenZeppelin, token-based accounts, account abstraction (ERC-4337), EIPs, DeFi, Uniswap, Chainlink oracles, cross-chain messaging (CCIP), Safe Multisig, Gelato relay (gasless), MetaMask, Aave, Synthetix, Zero-Knowledge Proofs (ZK), OmniChain, Data Visualization, Power BI

CERTIFICATIONS & ACHIEVEMENTS

- **Knight Badge on LeetCode (Highest Rating: 1873).** Among the **top 5.16%** of LeetCoders; solved **800+ DSA problems**.
- Achieved a **global rank of 1285** in Google Kickstart **(Link)**.
- Won Silver Medal in **ICPC Regionals, California, 2025** (2nd in California, 7th in NorthWest US Region, Div 2)
- Achieved first place at IIT Bombay's OPHacks, resulting in a spot in Optimism's accelerator program **(Link)**.
- Won **15 hackathons** (MLH and ETHGlobal), including Hack This Fall, ETHSF (500+ participants per event). **Latest win: Agent Orchestration Hack**
- **Certification:** Completed Google Cloud Facilitator Program (Qwiklabs Profile).
- **Publication:** Presented & published Decentralized Healthcare System paper (ICTIS 2024) - **(Link to the paper)**.