SAKSHAM RAI

(408) 513-5080 | raisaksham2001@gmail.com | Portfolio - 🔗 | LinkedIn - 🔗 | GitHub - 🔗

Education

University of California San Diego | *La Jolla,* *California* *October 2020 – June 2024*

- **Degree:** BS. Mathematics-Computer Science, Double Minor – Cognitive Science (ML Focus), Innovation and Entrepreneurship
- **Academic Achievements:** Departmental Provost Honors (x6), Sophomores Honors Program (x1), Blackstone Launchpad Program.
- **Associations**: CSForEach, Engineers for Exploration, UCSD Speech and Debate, The Basement Entrepreneurs Program and UCSD Cricket Team.
- **Relevant Coursework:** Web-Client Technologies, Software Engineering, Algorithms and Data Structures, Object-Oriented Design, Computation, Systems Programming, Unsupervised Machine Learning, AI Algorithms, Product Innovation and Venture Finance
- **Coursework GPA**: 3.7

Skills

Languages: Python, JavaScript, Java, C, Typescript, PHP, CSS and HTML

Frameworks/Databases/Tools: React, Next.js, Vue.js, Flask, MySQL, PostgreSQL, MongoDB, Firebase, Docker and Git/Github

Experience

Full-Stack Engineer | *County of San Diego* | React, Node.js, Postgres, Js, MySQL, PHP *Sept 2024 – Present*

- Engineered a real-time drug retrieval portal**, parsing & ingesting 5,000 + county records** into PostgreSQL**, reducing query latency by 20%**: Link
- Automated backend data synchronization via Google Sheets API & CRON jobs**, ensuring real-time updates.**
- Developed a Naloxone distribution platform**,** integrating JWT authentication & unique ID assignment, **securely tracking 5,000+ citizens' opioid access**: Link

Full-Stack Engineer | *Los Angeles County* | React, Node.js, MongoDB, Js *Oct 2024 – Present*

- Developing a secure opioid distribution API, enabling **8,000+ users** to access opioid kits via **21 vending machines**.
- Building MongoDB-backed user authentication with JWT, ensuring **privacy-compliant access control.**
- **I**ntegrated real-time inventory tracking by automating **unique ID generation, vending machine stock updates, and** Google Maps API **location tracking.**

AI Software Engineer Intern | *AfyaChat* | Flask, Python, FastAPI, Twilio, GPT 3.5, Llama Index *March 2024 – April 2024*

- Built an AI-powered eConsultation system, **reducing specialist response times by 50%** using GPT-3.5 & Flask.
- Built a Retrieval-Augmented Generation (RAG) pipeline, optimizing **7+ years of historical medical data** for real-time AI-based specialist recommendations.
- Engineered microservices architecture using FastAPI, ensuring **scalable AI model execution with low-latency inference**.
- Integrated real-time physician notifications via Twilio APIs, **reducing average patient response time** by **35%**.

Impact – Driven Projects

***GymScout* | Node.js, React, MongoDB, Typescript, Redis | Link** *Jan 2025*

- Developed a full-stack gym discovery platform, processing **1,000+ daily location-based searches** with Google Maps API & MongoDB geospatial queries**.**
- Optimized backend throughput, **reducing unnecessary API calls by 30%** via Redis caching and batched request handling**.**
- Implemented a dynamic review system, ensuring **real-time search ranking** based on **user engagement and location relevance**.
- Designed a responsive UI by implementing a dynamic React frontend with TypeScript and Tailwind CSS**.**

***Talk – To – Abel, SDX AI Hackathon* | Next.js, Firebase, WebSockets, ONNX, OpenAI, Whisper, VAD | Link** *August 2024*

- Developed a real-time AI-driven voice assistant**, processing 6,000+ live conversations** via WebSockets & ONNX runtime**.**
- Built a speech-to-text pipeline with Whisper AI & VAD (Voice Activity Detection), **reducing transcription latency** to **under 200ms**.
- Integrated streaming GPT-4 responses, using exponential backoff & retry logic, **decreasing API request failures by 30%**.
- Implemented Firebase authentication with Firestore-based session persistence, enabling **seamless multi-turn AI conversations.**
- Deployed on a scalable Next.js + Docker setup, ensuring cross-platform compatibility **with WebRTC-based low-latency interactions**.