George Ma

(661) 513-3350 | georgema2020@gmail.com | itsgeorgema | in/ggeorgema | ggeorgema.com

Education

University of California, San Diego — La Jolla, CA

***B.S. in Computer Science**, Minor in Business Analytics; GPA: 3.9* — *Expected June 2027*

Relevant Courses: Data Structures & Algorithms, Object-Oriented Programming, Discrete Math, Systems Programming

Technical Skills

Programming Languages: Java, Python, TypeScript, JavaScript, C/C++, Swift, SQL, Bash, Groovy, HTML, CSS
Libraries/Frameworks: React, Next.js, Node.js, Express.js, FastAPI, Flask, React Native, JUnit, PyTorch, scikit-learn
Technologies & Tools: AWS, Docker, Git, MySQL, PostgreSQL, REST API, Firebase, Gradle, CUDA, CI/CD

Experience

Praxie AI — San Francisco, CA

Software Engineer Intern — *April 2025 – Present*

- Built and shipped **15+** mobile-responsive UI components with **React Native/TypeScript**, serving **300+ users**
- Optimized tournament searches to perform in **sub-1s** by redesigning Firestore query nesting with pagination
- Engineered a data migration pipeline to **denormalize** complex structures, boosting data-fetching speeds by **20%**
- Refactored legacy code into modular hooks and components, reducing onboarding time for new features by **43%**

Alpha Kappa Psi @ UC San Diego — La Jolla, CA

Webmaster/Lead Developer — *December 2024 – Present*

- Spearheaded the full-stack migration of the chapter website from Wix to **Next.js**, **Tailwind CSS**, and **Supabase**, resulting in a more scalable platform with **60% faster** page load times for **2000+ monthly active users**
- Led a team of **4 developers** using **Git** and **Agile** methodologies to manage code reviews and feature tracking
- Designed a relational **PostgreSQL** schema to handle dynamic content, achieving **sub-1s** latency for all requests
- Authored comprehensive technical documentation to streamline the handover process for future webmasters

Solana Center for Environmental Innovation — Encinitas, CA

Data Engineer (Consultant) — *March 2025 – June 2025*

- Developed an automated Python **ETL** pipeline with **Pandas** and **NumPy** for feature engineering, standardization, and imputation on the client's waste collection program datasets, eliminating **15+ hours** of manual data cleaning
- Integrated predictive models (**SARIMA**, **Prophet**, **XGBoost**) into the pipeline with automated cross-validation and residual diagnostics to forecast trends with **88.2%** accuracy, helping the client optimize resource allocation
- Built a **Streamlit** dashboard to visualize program KPIs and trends, supporting the client's operational decisions

Projects

Watchdog | *Demo* — *August 2025 – September 2025*

- Built an **AI-powered CI/CD** automation for Github PR reviews, linting, and security scans for **6+** languages
- Engineered 2 **dockerized MCP microservices** using **FastAPI** and **Express.js** on **AWS ECS** and **Fargate**
- Implemented a parallelized smart-chunking algorithm for accurate, in-depth LLM reviews of complex code diffs
- Orchestrated a language-agnostic CI system to conditionally execute **13+ toolchains** for resource-optimized full-stack linting, formatting, and security scans, reducing workflow duration from 10 mins to **under 2 mins**
- Enhanced user security and reliability by implementing graceful degradation and least-privilege token scoping

Spotify Mood Player | *Demo* | Website — *April 2025 – August 2025*

- Developed a full-stack **React**/TypeScript web app that categorizes and plays music by mood using audio feature analysis through a **PostgreSQL** database and **Flask**/Python REST API deployed via CI/CD on **AWS Lambda**
- Engineered a first-party proxy managing Spotify **OAuth 2.0** flow and session state for cross-browser compatibility
- Achieved **92.6% accuracy** in track classification by designing a dockerized **MCP pipeline** leveraging OpenAI, fine-tuned with lyrics from the Genius API and audio features extracted from the iTunes API using Librosa
- Optimized API response time by parallelizing computations with a **ThreadPoolExecutor** per Gunicorn worker

Pokemon Generator | *Demo* | Website — *April 2025 – July 2025*

- Created a full-stack, dockerized web app generating custom Pokemon using **Flask**, Tailwind CSS, and **PyTorch**
- Designed a **RESTful inference API** that serves a custom **PyTorch Conditional GAN**, utilizing **CUDA** GPU acceleration to generate unique 256x256 pixel Pokemon images via user inputs mapped to latent condition vectors
- Implemented a relational **PostgreSQL** schema with **SQLAlchemy ORM** to map and store user's creations
- Developed a prediction service using **scikit-learn** RandomForestRegressors to simulate balanced game attributes