

Na Yoon Kang

310-922-2854 | nykang17@gmail.com | [linkedin.com/in/na-yoon-kang-b89376330/](https://www.linkedin.com/in/na-yoon-kang-b89376330/) | github.com/nykang17

EDUCATION

University of California, Los Angeles

Los Angeles, CA

Bachelor of Science in Computer Science

Mar. 2026

GPA: 3.865/4.0

Relevant Coursework: Software Construction, Software Engineering, Programming Languages, Operating Systems Principles, Data Structures & Algorithms, Data Science, Computer Network, Artificial Intelligence, Machine Learning, Deep Learning for Computer Vision, Human-Computer Interaction

TECHNICAL SKILLS

Programming Languages: JavaScript, TypeScript, Java, Python, C/C++, C#, Bash, SQL (Postgres), HTML/CSS

Machine Learning: TensorFlow, PyTorch, scikit-learn, NumPy, Pandas

Frameworks: React, Node.js, Express, Figma, Firebase, Svelte, Unity

Developer Tools: Git, GitHub, Docker, VS Code, Visual Studio, Linux/Unix

Languages: English, Korean

EXPERIENCE

Software & AI Engineer Intern

June 2025 – Aug. 2025

Twelverays | Javascript, n8n, Google Calendar API, Slack API, OpenRouter API, OpenAI API

Remote

- Built an AI-powered research assistant by integrating the OpenRouter API into n8n workflows; automatically generated company research briefs from upcoming Google Calendar meetings and delivered them via Slack, saving interview teams 30+ minutes of manual prep per lead
- Led the full lifecycle of an internal HR chatbot project (requirements → design → deployment), enabling 100% of employees to quickly access HR policy and benefits information through Slack, reducing routine HR inquiries and improving response efficiency
- Optimized automation pipelines by integrating APIs (Google Calendar, Slack, OpenAI, OpenRouter), resulting in more reliable and scalable workflows used by multiple cross-functional teams
- Enhanced AI-driven automation by fine-tuning prompts for two workflows, boosting response accuracy to ~85–90% and reducing the need for manual corrections

Software Engineer Intern, Main Sites Team

Oct. 2025 – Mar. 2026 (Expected)

Daily Bruin | React, Node.js, Typescript, Next.js, Python, Kubernetes

Los Angeles, CA

- Develop and maintain full-stack features for dailybruin.com serving 60,000+ weekly readers using React and Node.js
- Collaborate with cross-functional team in weekly meetings to design, implement, and deploy new projects for the main news site
- Work with deployment infrastructure including Kubernetes, servers and databases to scale production applications
- Refactor and upgrade existing codebase to improve site design, layout, and performance for student's readership

PROJECTS

Bruin Market | *React, CSS, Javascript, Node.js, Firebase*

Jan. 2024 – Mar. 2024

- Built a full-stack e-commerce platform for UCLA students to upload, sell, and purchase items with integrated real-time chat
- Implemented Google OAuth2.0 authentication with Firebase and structured NoSQL database for secure, scalable user management
- Optimized chat performance, cutting message load time by 15%, which improved user retention during beta testing

Gartic Band | *Svelte, Node.js, Express, Socket.io, Tailwind CSS, Postgres, Javascript*

Sep. 2024 – Dec. 2024

- Launched a multiplayer music composition game with real-time collaboration powered by Socket.io
- Designed sequencer interface and animations that increased peer play-test ratings from 3.8 → 4.5/5 on usability surveys
- Reduced sound playback lag by 20% with custom audio compression and optimized database schema

3D Tank Trouble | *TinyGraphics, Javascript*

Sep. 2024 – Dec. 2024

- Recreated Tank Trouble in 3D using TinyGraphics, implementing keyboard movement, collision detection, camera controls, and sound effects
- Engineered custom 3D shapes through vector manipulation, enhancing game realism and performance
- Conducted usability tests with classmates; 70% reported smoother gameplay vs. baseline demo version

Grow-a-gatchi | *React Native, Expo, TypeScript, Firebase, Raspberry Pi 5*

Sep. 2025 – Dec. 2025

- Designed and built a cross-platform mobile app (iOS/Android/Web) using React Native + Expo and TypeScript that gamifies indoor plant care by linking real-world watering events to a virtual Tamagotchi-style plant sprite with animations, leveling, and a watering streak calendar
- Integrated a Raspberry Pi 5 hardware system to physically detect and log watering sessions in real time, syncing data to Firebase for persistent state across devices
- Introduced custom plant animations that measurably improved user positivity ratings in usability testing by 30%
- Leveraged Firebase Realtime Database to achieve 300–400ms end-to-end sync latency between a physical watering event detected by a Raspberry Pi 5 + IMU sensor and the mobile app, enabling a near-instantaneous feedback experience