

Shaokang Jiang

(858)-319-7385 | shj002@ucsd.edu | resume.shaokang.me | github.com/ShaoKangJiang

EDUCATION

University of California San Diego

M.S. in Computer Science and Engineering, Advised by Michael Coblenz

La Jolla, CA

2022-2024 (expected)

University of Wisconsin-Madison

B.S. in Computer Science, GPA: 4.00/4.00, graduated with distinction in the major

Madison, WI

Graduated 2021

EXPERIENCE

Software Engineer

June 2024 – now

Pullscription

- Worked closely with Jago, the founder and CEO, to discuss, suggest, design, and implement the system structure for the organization's blog and shopping website, including both the customer and supplier end. The aim was to create an easy-to-maintain structure that is fast for users to access and optimized for SEO.
- Utilized Nginx and Docker to deploy the system on a server with CI/CD pipeline supported by GitHub Actions.
- Deployed and integrated the Matomo tracking system with a focus on privacy concerns and wrote unit tests.
- Collaborated closely with the team and provided guidance in implementing the website.
- Integrated Stripe for payment processing and Auth0 with JWT for authentication into the website, and wrote tests.
- Worked with the design team to create and implement the blog site, embedding the abstract content into the main website and iOS app. Wrote unit tests for them.

Researcher

April 2024 – now

Independent

- Worked with Jimmy Koppel on comparing and analyzing the advantages and disadvantages of using tutorials over the Cody system.
- Discuss, design, and implement an analysis plan to examine the data and write a paper.

Graduate Student Researcher

January 2023 – now

University of California San Diego

- Worked with Michael Coblenz on the usability analysis of autocomplete, designed, executed an experiment with 32 participants and published on the Foundations of Software Engineering.
- Worked with Michael Coblenz on the usability analysis of Copilot. Designed an experiment and executing it now.
- Designed and implemented a consumer-level universe eye-tracking platform for both web and native based VSCode, a Chrome extension and a VSCode extension to collect data.
- Worked with Kristen Vaccaro and Deepak Kumar, and collaborated with two other group members to assess political bias in YouTube's recommendation system.
- Developed a method to match long and short items using a newly developed matching algorithm, and devised a mixed design to measure, collect, and evaluate outcome variables.
- Worked with Jude Abishek Rayan under the guidance of Steven Dow on improving conference communication protocols for projects CueHub and conference management.
- Designed, implemented, tested, and deployed software using React, Firebase, Flask, GitHub Actions, Cloudflare, and Oracle Cloud, based on the weekly discussion results. Improved the product in a rapid process and assisted in designing the entire research workflow.

Teaching Assistant

January 2023 – now

University of California San Diego

- Major TA for Java and Data Structure courses (DSC30) with Soohyun Liao for two quarters.
- Sole TA for the Usability of Programming Languages course (CSE291) with Michael Coblenz for one quarter and TA for the same course for another quarter.
- Sole TA for Compilers (CSE 131) with Yufei Ding for one quarter, and Introduction to Data Visualization (DSC 106) with Sam Lau for another quarter.
- Guided group projects on programming languages and software engineering topics, and proposed, designed, and developed innovative course assignments.
- Managed course logistics, graded assignments, and collaborated with the professor and team members to refine course content. Conducted office hours and discussion sections to assist students and deepen their understanding.
- Received an 'Excellent' rating from the instructor, and over 90% of the students gave a 'recommendation' rating in the course evaluations.

TECHNICAL SKILLS

- **Languages:** Java, Python, Rust, C, C++, C#, HTML/CSS/JavaScript, TypeScript, Matlab, L^AT_EX, GAMS, Markdown, Haskell, SQL, Embedded JavaScript, Blogger Template Language, Shell
- **Frameworks:** React, Node.js, Vue, Jest, Vitest, JavaServer Pages, Flask, Expo (React Native), JUnit, Svelte, VitePress, Hexo, Electron, Chrome Extensions, VSCode Extensions, Puppeteer, Tampermonkey, JavaFX, Servlets
- **Platforms/Tools:** Git, Docker, GitHub, Google Cloud, Azure, Cloudflare (Workers and KV), Firebase, Oracle Cloud, VS Code, Visual Studio, IntelliJ, Eclipse, Nginx
- **Libraries/Services:** Auth0, Stripe, Matomo, MailJet, Daily.js, Socket.io, TensorFlow, WebGL, NLP.js, echarts, Matplotlib and more

PROJECTS

Check out more small projects on shaokang.me/projects or on my GitHub.

Pullscription websites | *Vue, Typescript, Vitest, Auth0, Docker, Nginx, Matomo, Stripe, Blogger* 2024

- Architected and implemented various components of a modern comic book shopping site, including the shopper and supplier interfaces, and a blog. Configured CI/CD pipelines using GitHub Actions and a cloud server with Nginx.
- Integrated Stripe payment solutions and Auth0, and deployed and integrated Matomo tracking for analytics. Collaborated with the design team to develop the blog using Blogger's template language and embedded its content into both the website and iOS app. Built tests using Vitest and updated Docker for the entire system.

Video matching and leaning rating system | *HTML, JS, Node.js, Oracle cloud, Worker* 2024

- Developed a text-based algorithm in Node.js to identify matched pairs between any two videos, paired with a concurrent pool management tool to fully leverage a multicore cloud server, surpassing any existing video matching algorithms in internal tests.
- Created an anonymous rating system with a front end built using pure HTML and JS, and a RESTful API backend deployed on Cloudflare Workers as an intermediary server for data recording. With the rating system, users can manually compare two randomly selected matched videos from a pool, manually label any bias for analysis.

Workshop Manager | *Electron, React, Socket.io, HTTPS, DDNS, GPTs, Worker, MailJet* 2024

- Designed and developed a 4-component workshop management system for real-time user engagement to foster idea generation and improve communication during meetings, capable of hosting at least 50 concurrent participants.
- Entire build To be used as a component in the Collective Intelligence-2025 conference.
- Pre-conference component: Created a registration system utilizing a RESTful API backend with Worker for user verification and sending invitation emails via Mailjet. The frontend was developed using a simple HTML page.
- In-conference Management Component: Uses DDNS to dynamically bind to a subdomain of shaokang.me. Developed with Electron as the main platform, with React for the frontend and a JavaScript backend running an HTTPS WebSocket server. The management system will communicate with and broadcast guidance to all participants with the help of GPT services for hint generation, while also updating the main screen with large-screen guidance.
- In-conference Client-side Ideation and Discussion Components: Both components allow participants to submit ideas in real-time during small group discussions and facilitate large group discussions on the submitted ideas in a subsequent phase. They were developed using React and Socket Client for real-time communication..

Cross platform Markdown app | *Electron, JavaScript* 2024

- An Electron-based markdown editor designed with basic operations like reading from and writing to disk, and a floating menu for easy access to titles and editing them, packaged for multiple platforms
- Frontend developed with React, featuring a custom-built lexer and parser for markdown; backend implemented in JavaScript to handle file reading and writing from local storage.

Cue Hub | *React, Firebase, Python, Flask, Daily.js, GPTs* 2024

- Updated, redesigned, and deployed an online meeting platform with support for user communication and ideation. Provided separate interfaces for facilitators (monitors) and participants (ideators), using React for the frontend.
- Integrated Firebase Realtime Database for data storage, real-time cue publishing from facilitators end and message syncing between ideators. Used Daily.js for real-time meetings and transcription, and utilized prompt-engineered GPTs to provide real-time cues.
- Built and deployed both serverless and server-based backend services using Python with Flask and Javascript to handle audio creation as needed by the frontend.

- Fixed several critical bugs left by previous development teams that could exceed Firebase quota limits, improved the overall user experience, and enhanced the robustness of the entire program.
- Successfully conducted 60 meetings on the platform without any errors.

HealthCare Chatbot | *Expo, Worker, GitHub, Jest, Vitest, Figma* 2024

- Led a team of five to create a chatbot with CI/CD pipelines using Agile methodologies. Self-delegate leadership roles when necessary. Consulted with experts in health care to understand their needs and requirements.
- Developed a chatbot using Expo to provide healthcare information and answer questions via a fine-tuned GPT service. Backend services were built with JavaScript and deployed on Cloudflare Workers to handle user information management, session storage, authentication and token management, using salted hashes for password storage.
- Learned methods for managing disputes and mediation conflicts within a team, including documenting divergence points and finding common ground. Realized the importance of communication in a team.
- Integrated Material Design principles and conducted unit tests using Jest for the Expo app and Vitest for the middleware server. Utilized GitHub Actions for the CI/CD pipeline.

Online interactive shell | *Java Server Page, Servlets, Java, Azure* 2023

- Developed an interactive shell that allows users to run Java source code online in a fully Java environment without accessing the actual source code, ideal for demonstrating command-line Java programs.
- The frontend was built using Java Server Page and Servlets, while the backend was developed in Java, featuring a custom-built pool management utility to manage connections, preserving resources by terminate connection at the right time. This allows multiple users to interact with the system at the same time.
- The entire tool was deployed on Azure under a free plan, successfully supported 30 users simultaneously.

Eyetracker Monitoring Platform | *C#, JS, Tampermonkey, Chrome extension, WebSockets* 2023, 2024

- Developed a universal platform for tracking eye movement in VSCode, integrating the Tobii Eye Tracker 5.
- The 2023 edition tracks specific window appearances and user interactions in the web-based version of VSCode. The frontend is built with JavaScript and deployed as a Chrome extension, while the backend is developed in C#. Both ends communicate using Chrome native messaging. The system successfully ran for 80 hours of user testing with 33 different users without any issues.
- The 2024 edition tracks the entire window arrangement and user interactions in VSCode, with the ability to define and track additional events as needed. The frontend is built with JavaScript and deployed as a script in VSCode, while the backend is developed in C#. Both ends communicate using WebSockets.

VSCode Experiment helper | *Typescript, Fiddler* 2022, 2024

- Designed and developed two VSCode extensions to monitor user interactions, upload collected data, and enhance the development experience using pure VSCode without requiring viewing any outside window.
- The 2022 edition introduced a CodeLens feature above each method signature in Java, enabling users to run provided test cases with a click and debug programs. 33 real users tested it, and there were no bugs or complaints.
- The 2024 edition modified the existed browser plugin on VSCode to support navigating between pages. Integrated Fiddler to intercept and modify requests, making it possible to browse websites within VSCode while coding.

An economical computer | *Node.js, GAMS, Puppeteer* 2022

- Automatically scraped computer components' performance data weekly using Puppeteer, with GitHub Actions and a cron scheduler, by simulating user behavior.
- Used optimization with preset LP problem-solving to find the best configuration.

Wise grader | *JavaScript, TensorFlow, Tampermonkey* 2021

- Developed and deployed a Tampermonkey script to expedite grading by suggesting scores based on length, N-gram analysis, and topic relevance for dictation questions, using the TensorFlow. The entire system operates locally.

Covid data report | *Node.js, Puppeteer* 2021

- Automatically scraped COVID data from various official sites, analyzed each site to extract relevant data, and delivered filtered results to subscribed users daily. This process was automated using Node.js with Puppeteer on a server, simulating user behavior when visiting websites, with the help of GitHub Actions and a cron scheduler.

Financial Organization App | *Expo, NLP.js* 2020

- Developed an app to help individuals organize expenses and income using voice interaction and simple chat, utilizing a local natural language understanding engine for Capital One.
- Built with Expo and packaged for both Android and iOS. Users can import and export their data in JSON format, allowing for migration or synchronization between devices. The app does not upload any data, addressing privacy concerns.

Financial Tutoring website | *HTML, Embedded JavaScript, Worker, JavaScript, echarts*

2020

- Designed and led a team to develop a static financial education website for Capital One. The site includes interactive learning tools, such as quizzes and a self-built, stateless learning progress tracking system. It also has various utilities, including a yearly budgeting tool.
- Built front end using embedded JavaScript for the main site and HTML with JavaScript for utility tools. Developed backend with JavaScript and deployed on Cloudflare Workers to handle requests, including the encryption and decryption of learning stage information using AES and crypto and a dynamically rendered quiz system. Users can download their encrypted learning progress and upload it to decode and resume learning, or use the same browser to continue learning. No data is stored on the server.
- Created detailed implementation guidance for group members and a user manual for the website and learning content. Integrated everything to automatically build and deploy using GitHub Actions.
- Followed Agile methodologies and communicated with stakeholders to understand their needs.
- Interacted with mentors from Capital One to discuss and understand the software development stages.

Energy Simulator | *HTML, WebGL, JS, Java, JavaFX, Three.js, echarts*

2020

- Designed and developed a web-based energy simulation tool using jsLPSolver to optimize energy sources based on historical data and project future trends using linear regression. Demonstrated the results with a pure static website built with HTML and JavaScript, showcasing the energy usage of a hypothetical energy system in a 3D environment using WebGL and JavaScript, with full support for modern browsers and partial support for IE 11.
- Supports interactive 3D models using A-Frame with Three.js, AR functionalities via QR codes, dynamic data visualization, and lightweight localization.
- Reimplemented most functionalities in JavaFX without 3D representation.

AutoCommenter | *Java, JFrame*

2019

- Designed and developed a local utility program in Java with JFrame to analyze Java code based on preset rules and semi-automatically add comments.
- Integrates user input through typing or voice commands when necessary, with voice recognition powered by the iFlyTek API and translation supported by the Baidu Translation API.

HateFate Website | *HTML, Java Server Page, SQL, Java*

2017

- Led a four-person team in developing a social networking website that connects users based on shared dislikes, with features including login, email-verified signup, and user pairing. Utilized Material design to create the interface.
- Developed the frontend using HTML and JavaServer Pages (JSP) and built the backend with JSP and SQL for data storage. Deployed on the Azure cloud with an Azure SQL database.
- Developed multiple login methods and utilized cookies to store user information, enabling automatic session recovery. Designed and integrated a custom mail system to handle email verification, password resets, and new match notifications using SMTP.
- Secured the website with SSL encryption and employed a combination of MD5, public/private key encryption, and AES for data security during page forwarding and email verification.
- Achieved first place in the class project demo.

PUBLICATIONS

-
- Shaokang Jiang and Michael Coblenz. **An Analysis of the Costs and Benefits of Autocomplete in IDEs.** Proceedings of the ACM on Software Engineering (FSE 2024) (25% acceptance rate)