Jason Xu

https://jasonxu.dev • https://github.com/JasonXu314
Xinwen31415@gmail.com • (314) 359-1208 • https://www.linkedin.com/in/jx6pc

Education

Missouri University of Science and Technology — May 2025
M.S. Computer Science — **GPA: 3.922**

Skill Set Summary

- Skillful in software design and development using OO principles.
- Familiar with multiple computer languages including C++, Java, JavaScript/TypeScript, Python, SQL, HTML, CSS.
- Familiar with multiple database systems including MySQL, MongoDB, PostgreSQL.
- Experience with multiple operating systems such as Windows, MacOS and Linux (Ubuntu, Kali, Fedora).
- Working experience with DevOps tools such as Git, NPM, Make, Maven, Gradle, Docker.

Selected Skills

Fluent:

JavaScript/TypeScript (6+ years)

- **React**, **Angular** as primary professional frameworks for internships and user-facing projects:
  - Built many personal projects in React, choosing **Next.js** for integrated full-stack support and SSR capabilities.
- **Nest.js** as primary MVC backend framework:
  - Worked with both NoSQL- and SQL-style databases in numerous projects.
    - Worked with MySQL and PostgreSQL in both managed deployments and self-hosted via Knex and Prisma ORM.
    - Worked with MongoDB both via MongoDB Atlas and self-hosted.
  - Created custom Vite plugin/mini-framework to support **Svelte** integration with both SSR and client-side hydration capabilities.
- Very familiar with **Auth0** for authentication and authorization on both backend and frontend.
  - Currently building own auth system from scratch, utilizing similar techniques as Auth0.
- **BABYLON.js** and **THREE.js** for 3D rendering
- **Svelte/SvelteKit** for personal, class, and hackathon projects
- Very familiar with fundamental DOM APIs including WebSockets, Canvas API, Fetch API, Promises, WASM, Custom Elements & Shadow DOM API, WebXR.
- Worked with Jest, Jasmine for unit testing, Cypress for end-to-end testing.
- 6+ years experience with CSS and design implementation.

C++ (3 years)

- First learned through coursework, used in several personal projects and programming competitions
- Very familiar with both OOP patterns and various low-level APIs
  - POSIX Socket I/O operations
  - Threading and interrupt handling
  - Syscalls and inotify API
  - Zlib & compression
- Wrote HTTP parser/server/client mini-frameworks from scratch (https://github.com/JasonXu314/cpp-express)
- Working on package management/build system

Deployment and Hosting (6+ years)

- Vercel
  - Primary deployment platform for full stack applications.
  - Utilized when speed is a priority and serverless environment is not a limitation.
- Divio
  - Primary deployment platform for stateful backend servers such as WebSocket servers, image manipulation or other calculation-heavy backends.
  - Familiarity with Docker container deployments.
- Self-hosting via Docker containers on own server with Nginx

Familiar:

Java (5 years)

- Used for school, programming competitions, and in early personal projects.
- Familiar with standard library classes and APIs and techniques for HTTP and multithreading.
- Some experience with Swing library for creating GUI applications.

Python (4 years)

- Used for some school assignments, particularly for statistical modeling and networking.
- Familiar with pip package manager and ecosystem.
- Familiar with Anaconda 3 and Jupyter Notebook
- Familiar with popular libraries including Flask, Django, Numpy, Pandas, SciPy, ScaPy, and MatPlotLib.

Other:

- Have some experience with Material UI, PrimeNG/PrimeReact, Mantine UI, etc. design systems.
- Limited experience with Unity and Godot basic concepts.
- Worked with both Lerna and Turborepo in some monorepo exploration projects.
- Some Rust development experience.
- Experimented with some basic fullstack C# with Blazor apps.

Working Experience

Hurricane Beryl — Houston, TX
Software Engineer Intern — July 2024

- Survived

Patterson UTI Drilling Co. — Houston, TX
Software Engineer Intern — May 2024 – August 2024

- Enhanced functionality on Tour Sheet/CMR viewer to allow arbitrary date selection
- Developed mobile app from scratch to mirror functionality of web portal, with a focus on user experience and app flow:
    - Worked with existing APIs to re-engineer and optimize data fetching and display
    - Designed, implemented, and verified interface and layout changes to minimize wasted navigation time
    - Implemented diffing/caching algorithms to minimize network usage and reduce latency
- Tech stack:
    - Angular/Ionic + Material UI for user interface
    - Capacitor for mobile interface
    - MQTT for real-time data stream access

Parsons Services Inc. — St. Louis, MO
Software Engineer Intern — June 2023 – August 2023

- Enhanced existing modeling software to provide additional data for model verification and iteration.
- Created software to interpret camera positioning data to recreate capture angles.
- Worked with existing libraries to transform data into correct coordinate systems.
- Tech stack:
    - Express.js for serving client data.
    - Potree + THREE.js for rendering point clouds and camera shots.
    - Proj4.js for performing mathematical transformations.

J.B. Hunt Transportation Services, Inc. — Fayetteville, AR
Software Engineer Intern — June 2022 – August 2022

- Built and deployed IT request tool and mileage calculator tool to production.
    - Used by other employees to request and receive support for technical issues on JB Hunt software platforms.
- Managed API communications and application logic within tools.
- Utilized Agile techniques to streamline software development with other interns on team.
- Collaborated with other engineers to improve and modernize outdated interfaces and optimize calculations to improve user experience.
- Tech stack:
    - Angular, Typescript, PrimeNG for UI components.

Washington University in St. Louis — St. Louis, MO
Software Engineer – Dr. Ting Wang Lab — May 2019 – September 2021

- Constructed and managed the database and backend systems for the 3D Epigenome Browser.
- Implemented host-centered real-time collaboration system with transferrable controls.
- Designed and implemented data serialization and compression schema to ease storage and transfer of data between front- and backend.
- Developed 3D modeling software to allow researchers to explore and visualize chromosome features.
- Led technical feasibility evaluations and idea generation, published in *Nucleic Acids Research* (https://pubmed.ncbi.nlm.nih.gov/35412637/)
- Tech stack:
    - Svelte + sapper for frontend controls & routing, SCSS for styling, Babylon.js for 3D rendering.
    - Express + MongoDB backend to persist model data and host real-time collaborative rooms.
    - Puppeteer to create snapshots of models for preview images.
    - Hosted on Divio via Docker container for persistent servers.

Missouri University of Science and Technology Rolla, MO

Introduction to C++ (CS1570) Grader Jan 2022 – May 2022, Aug 2023 - Current

- Created reference implementations to help students understand computational problem solving techniques.
- Performed code reviews on student work and gave detailed feedback regarding program correctness and best practices.
- Wrote software to automate repetitive grader tasks such as feedback pushing and grade submission.

Based Capital St. Louis, MO

Chief Technical Officer April 2021 – Present

- Computer system setup including hardware and software installation and optimization.
- Developed company website and profitability monitoring software.
- Provided technical consulting and feasibility/reliability assessments.
- Tech stack:
    - React + Next.js for frontend UI, SCSS for styling.
    - Mailjet & Zoho mail for email services.

Selected Side Projects

Personal:

ER Diagram Tool (https://cstk.jasonxu.dev/er)

- Designed and implemented GUI-based drag-and-drop editor from scratch to facilitate user-friendly editing of Entity-Relationship diagrams.
- Used by many students in databases class for completing homework and for designing project structures.
    - Dynamically sized shapes for entities, attributes, relationships, and EER constraints.
    - Toggles for weak entities, key/partial key/multivalued/derived attributes, and identifying relationships.
    - Support for freely positioned labels.
    - Support for saving and sharing diagrams among group members.
    - Built using Svelte, SvelteUI for frontend controls; canvas API with custom render/control engine for drawing diagrams; diagrams serialized on frontend and uploaded to Nest.js backend for storage with MongoDB.

Relational Diagram Tool (https://cstk.jasonxu.dev/rel)

- Designed and implemented GUI-based drag-and-drop editor from scratch to facilitate user-friendly editing of Relational diagrams.
- Used by students in databases class for completing homework assignments.
    - Dynamically sized tables and column positioning.
    - Toggles for table keys.
    - Automatic drawing of relationship lines between table columns.
- Tech stack:
    - Svelte, SvelteUI for frontend controls.
    - Canvas API with custom render/control engine for drawing diagrams.

JSFlap (https://cstk.jasonxu.dev/jsflap)

- Cloned functionality of JFlap, a finite state machine editor used in Theory of Computer Science class, with enhancements for user experience and additional features for quality of life.
    - Support for deterministic finite automata, nondeterministic finite automata, push-down automata, and Turing machines.

    - Support for step-by-step and instantaneous evaluation of automata with single and multiple input strings.
    - Validation of automaton construction and parameters.
    - Configuration for desired alphabet(s).
- Tech stack:
    - Svelte, SvelteUI for frontend controls.
    - Canvas API with custom render/control engine for drawing FSMs and evaluating inputs.

G—

- C++ compiler assistant with strong defaults and incremental compilation, designed from the ground up to simplify and streamline C++ code building and iteration.
    - Smart compilation will automatically find needed source files and include them for compilation.
    - Watch mode to automatically recompile code when changes are detected.
    - Included run, debug, valgrind, and gcov commands to run compiled executable by itself or in a debugging environment like gdb or valgrind, or to check code coverage with unit tests.
    - Support for alternative linkers, command line argument passthrough to g++ compiler, argument passthrough to running executable, compiling with debug symbols, and more.
- Tech stack:
    - Pure C++ for source analysis, CLI parsing, and command composition.
    - Used g++, ld, mold, gdb, valgrind, etc. under the hood.
    - Used inotify API to listen for file changes in watch mode.

FeauxS (https://feaux-s.vercel.app)

- Final project from OS class containing CPU simulation, exploration of various scheduling algorithms memory allocation, and integrated assembler.
- Features:
    - Semi-realistic step-based CPU simulation with flags and general-purpose registers based off of x86 assembly.
    - Simulation of I/O-bound operations and interrupt control structure to facilitate parallelism.
    - Process-based operating system handling scheduling, I/O delegation, and memory allocation, written in C++ and running in browser via WebAssembly.
    - Exhibition of multiple different scheduling algorithms, including First-In-First-Out, Shortest Job First, Shortest Remaining Time, and Multi-Level Feedback.
    - Support for writing and submitting programs to the computer on the fly with built-in assembler to load pseudo-x86 assembly into WASM memory for execution.
        - Included basic arithmetic operations with general purpose registers.
        - Included testing and setting of flag register bits.
        - Included conditional jumps and labels.
    - Ability to tune simulation parameters and simulated system specifications.
- Tech stack:
    - C++ for operating system code and CPU/IO device simulation, compiled to WebAssembly using Emscripten.
    - Canvas API with custom render/control engine based on the ones used for CSTK projects, modified to interact with WebAssembly.
    - Next.js for toolbar UI with PrimeReact for UI components and styles.

Open-Source Contributor:

SvelteUI (https://github.com/svelteuidev/svelteui)

- Fixed bugs with several components where custom Svelte actions were being rejected.

VSCode Material Icon Theme (https://github.com/PKief/vscode-material-icon-theme)

- Fixed bug where file association options were not being merged correctly, resulting in incorrect file icons for a few patches.

Selected Honors and Awards

- National Merit Scholarship
- Bright Flight Scholarship
- Curators Scholar Scholarship
- 2MM MORTON DEUTCH ENDOWED Scholarship
- Missouri S&T Scholarship

Clubs and Activities

- Missouri S&T Symphonic Orchestra
- Webmaster, Missouri S&T Underwater Robotics Design Team

- Dev Team, Pickhacks
- Regional Champion, ACF Fall
- Co-President, Ladue Computer Science Club
- Secretary, Academic Competition Organization
- Participated in TigerHacks, WaffleHacks, etc.
- Nationals Champion, National Ocean Science Bowl