

James Kolsby

jrkolsby@mac.com | (404) 386 6207 | New York, NY

jameskolsby.com
github.com/jrkolsby
[linkedin.com/in/jameskolsby](https://www.linkedin.com/in/jameskolsby)

Goldman Sachs

August 2020 - Present

Full Stack Developer – Marquee Risk Analytics

Cut page load times 15% by unifying and memoizing API responses. Built autosave settings and optimized report scheduling using diff algorithm in Redux / Reselect. Built Express service to render React pages and distribute PDFs using Node and Puppeteer, improved request throughput with worker pools. Built a visual compiler for internal charting language. Backfilled price data, debugged failed risk calculations, and delegated severe outages on client support.

1stdibs

August 2019 - April 2020

Frontend Developer – Platform & Dealer Experience

Doubled content team productivity by rebuilding internal CMS platform and optimizing API logic with React Hooks and GraphQL. Cut sales vetting times by 40% by replacing multi-page workflows with a PWA and minimizing API calls by implementing diff algorithm. Shortened dev / deploy times by 2 minutes by reconfiguring Lerna to build frontend packages in a monorepo. Organized meetings with product teams to organize bugs and delegate feature requests.

1stdibs

May - August 2018

Frontend Intern

Minimized handoffs to design team by migrating React component library to Storybook. Configured WebPack to load all JSX components into bundle. Added Jenkins CI pipeline to deploy compiled app to AWS S3 using Node and Bash.

Coursework

Columbia University

2019

BS Computer Science

Embedded Systems

Implemented a networked chat room on an FPGA as a USB peripheral with VGA output. Designed a falling sand game by performing 60fps frame buffer transformations on DDR3 memory using Intel® Quartus IP cores.

Columbia Computer Music Center

Worked with a research group restore a prototype 1950s Buchla Synthesizer. Thoroughly documented malfunctioning components and modern replacements. Returned fully operational to CCMC.

Programming Languages and Translators

Designed programming language for graph algorithms with syntactic sugar for succinct traversals and literals. Implemented in OCaml using Yacc and Lex for parser, and C / C++ for standard library algorithms. Targeted LLVM intermediate Representation. Wrote automated testing suite in Bash and set up development environment in Docker with a team of five.

Projects

Song Garden

2020

Built Digital Audio Workstation in Rust with realtime multi-threaded audio engine. Implemented audio recording using buffer pools, fast scrubbing using Whittaker–Shannon interpolation, and a scriptable plugin API using the Faust DSP language. Deployed cross-platform binary using ASLA and PortAudio APIs.

QuickSub

2018

Minimized keystrokes in closed captioning application by pre-splitting audio regions in a video stream with Voice Activity Detection on Python and Node.

WeView

2019

Built a bracket voting app in React for realtime WebSocket decision trees. Used Flask and Python with an SQLite database for backend service. Built user token system for moderator and admin permissions.

Periodic iOS App

2015

Developed an intuitive and gesture-based chemistry app in Swift and Objective-C using UIKit. Sold over 800 apps over two years and supported three versions of iOS on iPhone and iPad.