Jay Chakravarty
jaychakravarty@alumni.harvard.edu
+1 (425) 306-8865
Newcastle, WA 98056
www.linkedin.com/in/jay-chakravarty

Experience:

Technology Analyst — New York, NY
Goldman Sachs — July 2015 – Apr. 2017

- Design, build, and maintain high-performance yet nimble and adaptive platforms for clients.
- Understand US Cash and Options market structure, regulations, and exchange service offerings.
- Communication with traders, sales, and clients about explanation of new feature requests.
- Proficiency in Java and C++ and experience with Linux operating system.
- Strong knowledge of data structures, algorithms, and design pattern.
- Strong communication skills and the ability to work in a team.
- Experience with Event Sourcing Sequenced core stream-based architecture.
- Knowledge about JVM (Java Virtual Machine) internals and tuning for low latency.
- Experience with building C++/Java systems with strict requirements for performance.
- Experience with developing order management and exchange connectivity systems.
- Implemented multithreading, concurrency control, and low-latency optimization techniques

Education:

Full-Stack Development with JavaScript — Seattle, WA
University of Washington — Oct. 2024 – June 2025

- How to combine JavaScript with CSS and HTML for application development
- DOM (Document Object Model) manipulation, data structures and state containers
- How to create single-page applications using React.js
- Various JavaScript libraries and how to bundle them using Vite
- How to use Node.js for back-end web application development
- Built a back-end web application using HTML, JavaScript, and Node.js

Programming for the Browser with JavaScript:

- How to run JavaScript on the browser
- DOM (Document Object Model) manipulation in response to user and system events
- Techniques for working with third-party APIs to retrieve and integrate data
- How to utilize JSON and JavaScript libraries
- JavaScript design patterns and basic object-oriented programming

Front-End Application Development with JavaScript:

- How to write code in ECMAScript/ES6 and take advantage of design patterns
- How to set up and use the Node.js framework environment
- Techniques for using state containers such as Redux to store and update data
- Front-end workflow tools and processes (Git, CI pipeline, code reviews, linting)
- How to interact with a document database (Firebase)

Back-End Application Development with JavaScript:

- How to create a web application server with Express
- Interactions with a document database
- Isomorphic JavaScript development
- How to design RESTful APIs
- Cloud hosting using AWS, Azure, and others

Front-End Development with HTML, CSS, & JavaScript — Seattle, WA
University of Washington — Feb. 2024 – Oct. 2024

- How to write HTML code and intrinsic APIs, and do simple styling with CSS
- The latest features of HTML and CSS, such as the ability to produce interactive animations
- Responsive design concepts and methods
- JavaScript programming basics
- Design principles tailored to how sites interact with devices such as tablets, smartphones, cameras, and GPS units
- Created a responsive, client-side web application

HTML & CSS Foundations:

- Proper use of elements, including new HTML semantic elements
- Custom styling, including web fonts
- Basic elements of client-server architecture
- Principles of effective user experience
- Media playback with new HTML media elements
- Bootstrap fundamentals

JavaScript & Responsive Design:

- How to make dynamic sites interactive with JavaScript
- Best practices to design a mobile-first responsive website
- How to use Git to work with projects started by others
- How to ensure that designs are cross-browser compatible

JavaScript & Responsive Design:

- How to write functions in JavaScript using ES6
- Building websites with dynamic data from an API
- Organizing code using Vue, a front-end JavaScript framework
- Applying transitions to create animations that move when the user hovers over them
- How to create a responsive site using Bootstrap 5 or the CSS grid system

Bachelor of Arts in Computer Science — Cambridge, MA
Harvard University — August 2011 – May 2015

- *Introduction to Computer Science* (C, Arrays, Algorithms, Memory, Data Structures, Python, SQL, HTML, CSS, JavaScript, Flask):
  This course teaches you how to solve problems, both with and without code, with an emphasis on correctness, design, and style. Topics include computational thinking, abstraction, algorithms, data structures, and computer science more generally. Problem sets inspired by the arts, humanities, social sciences, and sciences. More than teach you how to program in one language, this course teaches you how to program fundamentally and how to teach yourself new languages ultimately. The course starts with a traditional but omnipresent language called C that underlies today's newer languages, via which you'll learn not only about functions, variables, conditionals, loops, and more, but also about how computers themselves work underneath the hood, memory and all. The course then transitions to Python, a higher-level language that you understand all the more because of C. Toward term's end, the course introduces SQL, via which you can store data in databases, along with HTML, CSS, and JavaScript, via which you can create web and mobile apps alike. Course culminates in a final project.

- *Introduction to Programming with Python* (Functions, Variables, Conditionals, Loops, Exceptions, Libraries, Unit Tests, File I/O, Regular Expressions, Object-Oriented Programming):
  An introduction to programming using a language called Python. Learn how to read and write code as well as how to test and debug. Designed for people with or without prior programming

experience who'd like to learn Python specifically. Learn about functions, arguments, and return values; variables and types; conditionals and Boolean expressions; and loops. Learn how to handle exceptions, find and fix bugs, and write unit tests; use third-party libraries; validate and extract data with regular expressions; model real-world entities with classes, objects, methods, and properties; and read and write files. Hands-on opportunities for lots of practice. Exercises inspired by real-world programming problems.

- *Web Programming with Python and JavaScript* (HTML, CSS, Git, Python, Django, SQL Models, and Migrations; JavaScript, User Interfaces, Testing, CI/CD, Scalability and Security):
  This course dives more deeply into the design and implementation of web apps with Python, JavaScript, and SQL using frameworks like Django, React, and Bootstrap. Topics include database design, scalability, security, and user experience. Through hands-on projects, you learn to write and use APIs, create interactive UIs, and leverage cloud services like GitHub and Heroku. By semester's end, you emerge with knowledge and experience in principles, languages, and tools that empower them to design and deploy applications on the Internet.

- *Introduction to Artificial Intelligence with Python* (Search, Knowledge, Uncertainty, Optimization, Learning, Neural Networks, Language):
  This course explores the concepts and algorithms at the foundation of modern artificial intelligence, diving into ideas that give rise to technologies like game-playing engines, handwriting recognition, and machine translation. Through hands-on projects, you gain exposure to the theory behind graph search algorithms, classification, optimization, machine learning, large language models, and other topics in artificial intelligence by incorporating them into your own Python programs. By course's end, you emerge with experience in libraries for machine learning and knowledge of artificial intelligence principles that enable you to design your own intelligent systems.