

# JEREMY HSU

jeremyhsu.me | jeremyhsu@college.harvard.edu | github.com/hsujeremy

## EDUCATION

**Harvard University** | B.A. in Computer Science

May '24

**GPA** 3.84/4.0 · **Computer Science GPA** 4.0/4.0

**Relevant Coursework** Abstraction and Design in Computation · Design of Useful and Usable Interactive Systems · Linear Algebra and Differential Equations · Introduction to Computer Science · Multivariable Calculus

## EXPERIENCE

**Lime** | Software Engineer Intern – [Partnerships Platform](#)

San Francisco, CA

Ruby on Rails, JavaScript, Java, MySQL

May '21 – Present

- Developing a server-side API to enable seamless MaaS transit integration between Lime vehicles and third-party apps and services set to launch early this fall.
- Architecting and implementing the backend logic and corresponding API endpoints for the user lifecycle, trip refund flow, and real-time vehicle data.
- Creating UI components for a company-wide internal administrative dashboard that allow for permission access-control assignments and viewing mobility data feed instructions for regions and external partners.

**Cisco** | Software Engineer Intern – [Webex Media Engine](#)

San Jose, CA

C++, Python, Elasticsearch

May '20 – May '21

- Overhauled and extended the team's suite of internal debugging and triage tools by writing new and updating existing automation, analysis, and visualization scripts, deployed using custom-built Jenkins jobs and CLI commands, saving engineers multiple hours per day.
- Collaborated with overseas teams to develop and deploy a data aggregation module leveraging the Elastic API that resulted in Kibana dashboards loading up to 5x faster.
- Debugged production code and wrote unit tests for the Webex Media Engine metrics and audio echo canceler components.
- Created and monitored interactive dashboards visualizing critical app quality and network performance metrics, enabling engineers to identify problematic trends and issues more rapidly.

## ACTIVITIES

**Datamatch** | Algorithm Lead

Cambridge, MA

C++, Python, Sentence-BERT

Sep '19 – Present

- Wrote the mutual and pseudo-random matching algorithms with two other members for Match21, Datamatch's May launch targeting graduating Harvard seniors.
- Improved Datamatch's user-to-user compatibility score function by collaborating with a small team to implement semantically-nuanced sentence embeddings with Sentence-BERT and apply an inverse-proportional weighting to questions to account for distribution polarity in survey responses.
- Datamatch's February 2021 launch matched 42,000+ students across 34+ universities, including Harvard, MIT, and Berkeley.

## PROJECTS

**YouTube Party**

[ Node.js, React, Socket.io, YouTube iFrame API ]

A full-stack web app that allows any number of users to stream YouTube videos synchronously using web sockets. The site synchronizes pause and play operations, keeps track of the number of online users, and includes an in-app chatroom.

**ML-Enabled Spotify Curator**

[ Flask, React, Celery, Redis, Scikit-Learn, Spotify Web API ]

A full-stack web app that predicts whether a user will like a particular song based on musical characteristics of songs in their Liked Songs library. Song characteristics are fetched from the Spotify Web API while training and prediction tasks are handled asynchronously on the backend using Celery as a distributed task queue and Redis as both a transport store and message broker.

**Roommate Hub**

[ Swift, UIKit, MessageUI, Firebase ]

An iOS app organizing the shared living experience for roommates. Features include secure user authentication, synchronized task lists, interactive roommate profiles, anonymous message forums, and embedded iMessaging.

**OCaml Interpreter**

[ OCaml ]

A set of interpreters for a Turing-complete subset of OCaml with support for simple atomic types; unary and binary operations; recursion; and higher-order functions. Each line of OCaml is evaluated following three distinct semantic models: the substitution model, the dynamically-scoped environment model, and the lexically-scoped environment model.

## TECHNICAL SKILLS

**Programming Languages** C++ · Java · Python · JavaScript · Swift

**Frameworks & Technologies** React · Ruby on Rails · Node.js · SwiftUI · UIKit · Firebase · Elasticsearch · Figma