CHUXI WANG

(215) 356-6516 | wangchuxi94@gmail.com | Fremont, CA, 94536

EDUCATION

University of San Francisco *Aug. 2020 – Dec. 2022*

- Master of Science in Computer Science (**GPA: 4.0/4.0**)

University of Pennsylvania *Aug. 2016 – May. 2018*

- Master in Biotechnology

Nanjing University (Nanjing, China) *Sep. 2012 – Jun. 2016*

- Bachelor of Science in Biological Sciences

WORKING EXPERIENCE

Meta Platforms, Inc. *[Flask, Socket.IO, React, AWS S3, ML/NLP]* **Menlo Park, CA**

Software Engineering Intern (FAIR Labs) *Jun. 2022 – Aug. 2022*

- Built a dashboard for the Human-in-the-loop learning (HiTL) system of the *Droidlet* project, where *Droidlet* is a modular, heterogenous, multimodal agent architecture and platform. In this project, users can easily monitor job progress, view experiment results, and manage assets (e.g. datasets, models) interactively.
- Implemented the dashboard application with three layers: a cloud system data storage layer that stores data (including experiment metadata, logs, datasets, etc.) on AWS S3, a Python backend layer, and a React.js frontend layer.
- Developed the backend that retrieves data from S3 via boto3 and serves data to the frontend layer via Socket; designed and implemented APIs with Flask to support reading experiment runs, interaction logs, datasets, and models, reading and updating of the model version, and the allow & blocklist of the crowdsourcing workers.
- Created extensible and reusable dashboard components that support viewing and managing Natural Language Understanding (NLU), Turk-as-Oncall (TAO), and Vision pipelines with Ant Design components and React-router-dom; enhanced user experience by developing interactive graphs with ReChart to improve model visualization.
- Improved TAO pipeline of the *Droidlet*, where crowdsourcing workers interact with our agent, report agent errors, and route the errors to the engineers.

Salesforce.com, Inc. *[Java, Lightning Framework, Spring, MySQL, MVC, RESTful API]* **San Francisco, CA**

Software Engineering Intern (Record Access Control) *Jun. 2021 – Aug. 2021*

- Developed the Admin UI for Restriction Rules based on RESTful API for admins to manage rules for accessing different salesforce records. Implemented a highly interactive and accessible list view and detail view using the Lightning Aura framework, which is an MVC framework with a markup view, a JavaScript controller, and a Java model layer. This Admin Restriction Rule UI is a highly requested feature by users and resolves a critical functionality gap as Restriction Rule is an API-only feature in the previous release.
- Worked with cross-functional teams to refactor a copious amount of lightning aura components, added more interfaces to the components, and made them more accessible and forward compatible.
- Maintained and improved the stability of the backend RESTful APIs to create, delete and update Restriction Rule based on Java *Spring*.
- Designed and developed JavaScript unit testing using a host-agnostic JS test framework xUnit.js for the JavaScript controllers and helpers.

University of San Francisco Information Technology Services *[Apex, Python, Unit Testing]* **San Francisco, CA**

Part-time Student Developer (Salesforce Team) *Apr. 2022 – May. 2022*

- Maintained the USF Student Hub web application which is a Salesforce Application and wrote Apex tests to improve the application's test coverage. Developed easy-to-use utility scripts to visualize duplications in charity donation records of USF using Python and Salesforce APIs.

SKILLS

- Java, JavaScript, Python, C, SQL, Go, Swift, C++
- Databases & Cloud: MySQL, AWS EC2, AWS S3, Mephisto DB
- Web: Java Servlet, Node.js, HTML, CSS, React, Spring, Hibernate

PROJECTS

Agami: Live-updating & interactive visualization of streaming data. *[Spark, Streaming, Distributed System]*

- Implemented a Jupyter Notebook client for real-time querying of the steaming data and animated visualization of the queried streaming data with *PySpark* & *Altair*. Wrote scripts to compare the querying and plotting performance of running Spark with various worker count, thread count, and master machine combinations in a cluster.