

Jacky Xia

(415) 309 8111 | jacky.w.xia@gmail.com | [linkedin.com/in/jacky-xia](https://www.linkedin.com/in/jacky-xia) | github.com/jx114

Technical Skills

Languages | JavaScript(ES5/6) \ TypeScript \ HTML \ CSS

Libraries | React \ jQuery \ Axios \ Ajax \ Webpack \ Babel \ Compression \ Express.js \ Node.js \ MobX State Tree

Databases | MongoDB \ MySQL \ SQLite \ PostgreSQL

Deployment | AWS(EC2, S3) \ Heroku \ NGINX \ Docker

Testing | Jest \ Enzyme \ Mocha \ Jasmine \ Chai \ CircleCI

Experience

MedifriendRx - *Software Engineer*

2020-Present

- Created integration test suite to ensure persisting state management between models and API routes along with websocket interactions.
- Tested, iterated, and recorded at each step while balancing the load times of additional servers, utilizing NGINX proxy server.
- Integrated custom automated tester which builds out all the combinations of the test cases with pertaining test conditions then initiates testing those cases to minimize redundant code.
- Modeled relationships between patients, physicians, and pharmacists with prescriptions, drugs, and containers while maintaining state management with MobX State Tree.
- Utilized Jest/Enzyme unit tests and supertest for API calls; implemented CircleCI for Continuous Integration.

Software Engineering Applications

HouseHub: *I React, Express/Node, MongoDB, NGINX, AWS, Docker, Jest/Enzyme*

Developed and deployed the front end of a home/property rental application.

- Developed a photos gallery that dynamically renders photos of the listing with ratings/location/save functionality with two modals which adapts to the viewport size.
- Optimized RPS **90%** by horizontally scaling microservice to multiple AWS instances using Docker, which improved performance from 3105ms to 298ms fetches data such as images and bundles from other instances.
- Utilized Jest/Enzyme unit tests and supertest for API calls; implemented CircleCI for Continuous Integration.

britestack *I React, Express/Node, PostgreSQL, NGINX, AWS, Docker*

Designed and deployed the back-end of a real estate listing application

- Improved overall performance by 95% and reduced time from 392ms to 201ms by balancing server loads while maintaining a baseline of a client load consisting of 200-2000 clients per minute. Maintained a client load of 200-2000.
- Optimized response times by constructing meaningful relationships and indexing between mortgages, listings, and users to improve query times from server to database.
- Utilized Jest/Enzyme unit tests and supertest for API calls; implemented CircleCI for Continuous Integration

MelodyMatch *I React, Express/Node, MongoDB, Spotify API*

Created and designed a full stack application that allows you to swipe between less popular artists.

- Implemented swiping functionality where the users can swipe either left or right on the presented song which allows them to build their personal preferences for specific artists and their genre.
- Created algorithm of building personal preferences and catering that preference is the data catalogs of liked or disliked artists with the genre, artists, and danceability of the track randomly provided.
- Designed backend to manage music preferences using Spotify API with user's music preferences persisted through front-end interface interaction of swiping through less popular artists.

Education

Hack Reactor - Advanced Software Engineering Immersive Program

2020

UC Berkeley - Bachelor of Arts, Political Science

2017