Pleasanton, California
pctran98@gmail.com
(510) 406-9625

MINH TRAN

mctran-phi.github.io
github.com/mctran-phi
linkedin.com/in/minhctran

TECHNICAL SKILLS

Frontend: React.js, Next.js, HTML, CSS, JavaScript/Typescript
Backend: Express.js, Node.js, MongoDB, PostgreSQL, MySQL, Java, C++
Tools and Utilities: Webpack, Babel, Docker, AWS EC2, NGINX, Mocha, Chai, New Relic, jQuery, Socket.io

EDUCATION

Hack Reactor, *Advance Software Engineering Immersive Program* (2021)
California State University East Bay, *B.S. Computer Science* (2020)

- **Coursework:** Data Structures & Algorithms, Computer Network, Operating Systems, Analysis of Algorithms, Database Architecture, Mobile Programming, Web Development, Software Engineering

SOFTWARE APPLICATIONS

Tattoo Art | Fullstack | React.js, Express.js, Node.js, MongoDB, Typescript
Github | Demo
Tattoo Art is an application for users to create contests and compete as well as allowing users to show their work and artistic view.

- Constructed client using **React** and **MUI** for its quick and responsive UI.
- Structured server with **RESTful API** architectural style and **MVC** framework.
- Integrated **Sockets** for real-time private messaging to other users.
- Designed **UI/UX** for various components/pages.

Congo Prime | Fullstack | React.js, Express.js, Node.js, MongoDB, AWS EC2
Github |
An e-commerce platform composed of various microservices to improve user's experience with product purchase and surfing.

- Rendered using **React** with **styled-components** to isolate modules from other microservices.
- Utilized **MongoDB** for its schema-less database, deep queries abilities, and simple/quick setup for project's agile workflow
- Deployed server on **AWS EC2** to provide accessibility to all users, scalability, and reliability.
- Improved page load time by compressing the downloadable data served to users which reduces payload size and increased page load speed from 7 to 48.

Reactors | Back-end | PostgreSQL, AWS EC2, NGINX, New Relic
An e-commerce app that holds over 10M records of data in the database, which is scaled to handle thousands of user's requests.

- Populated database with COPY query and write stream to efficiently load a large amount of data.
- Created indexes to optimize queries, storing ids in a B-tree, and reduced query execution time from ~2000ms to < 50ms.
- Stressed test database, received a throughput of ~10k rpm with ~800ms per request on the local machine, and improved to ~40k rpm with ~1400ms per request on **EC2** instance.
- Horizontally scaled database using **NGINX** load-balancer, round-robin implementation on 4 microservices, received ~112k RPM with ~3.32ms per request.

Fanime | Front-end | Javascript, Next.js, React.js
Github | Demo
Anime viewing platform that provides users a fast and simple method of anime searching/browsing.

- Constructed a server-side rendering application for immediate availability to improve user's experience and to be detected by search engines.
- Implemented infinite scrolling over pagination to improve UX.
- Integrated CSS modules to locally scope components and for simple refactoring.