JIMMY NGUYEN

React Developer Engineer

Milwaukee, WI
(262) 957-6232
Nguyen89.MKE@gmail.com
LinkedIn | GitHub | Portfolio

TECHNICAL SKILLS

Next.js, React, React Hook Forms, MongoDB, Node, JavaScript, RESTful APIs, HTML/CSS, GitHub, Mocha, Jest

SUMMARY

Experienced React Frontend and Full Stack Developer skilled in Next.js, React, and Prisma. Strong track record leading end-to-end frontend projects, mentoring junior developers, and consistently exceeding client expectations. Proficient in responsive UI design and database management, with a commitment to delivering high-quality solutions. Ready to contribute expertise and create exceptional user experiences.

PROFESSIONAL EXPERIENCE

React Frontend Developer | GCOM Partners | Milwaukee, WI — **March 2023 – Current**

Technologies: Next.js, React, React Hook Forms, Prisma

- Spearheaded the development of a feature-rich frontend application using cutting-edge technologies, including Next.js 13, Chakra UI, React Hook Forms, React, JavaScript, and Storybook.
- Single-handedly owned the end-to-end frontend development, conceiving and crafting all visual components, driving the project's user interface from concept to reality.
- Demonstrated leadership and commitment to team growth by coaching and mentoring junior developers in the intricacies of React and functional components, fostering a culture of continuous improvement.
- Faced a formidable time constraint and emerged triumphant by delivering a complete frontend web application on schedule. The innovative use of reusable functional components ensured both swift development and a remarkable client experience.
- Exceeded client expectations to such an extent that our contract was extended, reflecting our exceptional commitment to delivering high-quality solutions and outstanding client satisfaction.

Full Stack Developer | Wcket LLC | Remote — **October 2021 – March 2023**

Technologies: MongoDb, Express, React, Node, ChakraUI, MochaJS

- Developed responsive web interfaces with HTML, CSS, and JavaScript, leveraging React for enhanced interactivity and user experience.
- Managed server-side programming, utilizing Node.js and Express to handle requests, data processing, and database communication.
- Designed and maintained SQL database schemas, built RESTful APIs for seamless front-end to back-end communication, and ensured efficient data storage and retrieval.
- Implemented a comprehensive testing strategy, including unit, integration, and end-to-end testing, to maintain application reliability and code quality.

EDUCATION

TECH I.S. | Web Development — Completed 2022

University of Wisconsin - Milwaukee | Human Resources — 2016-2019

Full Stack Developer Certificate
Programming Foundations: Algorithms Certificate
Programming Foundations: Object-Oriented Design Certificate

Personal Projects

Full Stack Web Application | MERN - MongoDB, Express, React, Node | MMORPG Resource Website

Technologies: MongoDb, Express.js, React.js, Node.js

- Collaborated with other developers in the gaming community to put together a web application that provides information in a convenient area to speed up player progression.
- Fetching data and hosting it in a convenient and accessible location has increased progression speed by up to 45%. (Numbers compared during speed runs)
- Created as a free resource to support the community, it has received positive feedback and support. With plans to incorporate ideas from other members of the community, there will be more features implemented.

Player Profile MOBA API

Technologies: MongoDb, Express.js, React.js, Node.js

- Using Riot Games' public API, this website searches for a given name and returns their player profile and details.
- When the page is rendered, the useEffect hook on mount sends a GET Request for the JSON data. Upon success, stores the JSON data to the state, then compares it to the given search string.
- The client sends CRUD requests to the server which then communicates to the MongoDb database.

2D Fighting Game - Vanilla JS

- Created with Object Oriented Design and Programming fundamentals.
- Characters were created under a class with objects inheriting shared characteristics from it.
- Refactored the code using polymorphism and was able to remove Switch/Case or long If/Else statements.