

# PRASHASTI KARLEKAR

+1 (765)-346-7967 | [prkarl@iu.edu](mailto:prkarl@iu.edu) | [LinkedIn](#) | [GitHub](#) | Dallas, TX

## EDUCATION

Indiana University Bloomington, M.S. Data Science (Computational Track)	August 2021 – May 2023
Rajiv Gandhi Technical University, B.E. Computer Science	August 2014 – May 2018

## WORK EXPERIENCE

Indiana University	Bloomington, IN
Associate Instructor (Engineering Distributed Systems)	January 2023 – May 2023

- Mentored 60+ students in designing fault-tolerant, scalable distributed systems projects, emphasizing coding standards and architectural best practices while utilizing industry-standard tools such as **Redis** and **Kafka**.
- Executed comprehensive grading process for exams and assignments, conducted code reviews, and promptly resolved student queries, resulting in 98% satisfaction rate as evidenced by online student feedback.

Indiana University	Bloomington, IN
Graduate Research Assistant	March 2022 – December 2022

- Led UI/UX design teams to migrate the multi-page user experience into a single-page web application using **React.js**, **JavaScript**, and **Bootstrap** for AEI Lab with optimized loading and caching, enhancing user engagement by 8%.
- Created 5 **D3.js** data visualizations on artists' data that facilitated data exploration and effectively identified trends.

Bloom Insurance Agency	Bloomington, IN
Software Developer Intern	May 2022 – August 2022

- Enhanced Ascend application for agents by integrating Formik for client-side validation and profiling critical rendering paths using Chrome DevTools/Lighthouse, yielding an optimized UX with a 20% faster page load.
- Leveraged Agile and Scrum-aligned code reviews to enforce adherence to industry-standard coding practices, contributing significantly to identifying and resolving critical issues, leading to a 30% decrease in post-release defects.

Tata Consultancy Services	Ahmedabad, India
Software Developer	November 2018 – August 2021

- Collaborated with CTOs and analysts to develop a user-friendly front-end of an in-house fitness web application by utilizing expertise in **React Native**, **RESTful web services**, and Material UI, currently boasting over 10,000 users.
- Utilized application state management libraries such as **React Redux**, authorization mechanisms, and tools such as **Expo**, **React Navigation**, and **Visual Studio Code** to create a secure and reliable user experience.
- Developed a web-based dashboard using **React.js** for administrative tasks and health data monitoring by implementing backend systems using **Node.js**, integrating RESTful and **GraphQL** services, and retrieving fitness data using complex **SQL** queries to enable data-driven decision-making within the software development life cycle.
- Implemented robust **CI/CD** pipelines using **Jenkins**, integrated **Jest** for testing, and front-end build tools like **Babel** and **Webpack**, while utilizing version control systems such as **GitHub**, reducing deployment time by 60%.
- Enabled the transition to paperless practice in the healthcare domain by automating document approvals using **Microsoft Power Automate**, achieving an 80% reduction in manual intervention and client overhead costs.

## SKILLS

**Programming languages and Web technologies:** Python, JavaScript, TypeScript, ReactJS, React Redux, NodeJS, C, C++, Java, HTML, CSS, JSON, Bootstrap, RESTful Webservices, Webpack, Parcel, GraphQL

**Databases:** MySQL, Postgres, Oracle, MongoDB, Microsoft SQL Server

**Software and Tools:** Visual Studio, Git, AWS, GCP, Microsoft Azure, RabbitMQ, Containers and Docker

## PROJECTS

**Distributed Map-Reduce :** ( Python, GCP, Shell / Bash Scripting, Network Sockets )

- Designed a distributed MapReduce system implementing Map Reduce as a library for data processing, including Word Count and Inverted Index applications, with integrated design patterns for scalability and reliability, utilizing Google Cloud Platform (GCP) services such as cloud functions, virtual machines (VMs), and cloud storage.

**Distributed Multi-Consistency Key-Value Store :** ( Python, GCP, Shell, Memcached server, TCP/IP )

- Implemented different consistency models like Sequential, Linear, and Eventual Consistency that use the underlying key-value store replicas by enforcing consensus algorithms and meeting the requirements of the CAP theorem.

**C++ Bro Server :** ( C++, Socket Programming, Multithreading, Text-processing )

- Engineered a high-performance HTTP Web server utilizing socket programming and object-oriented techniques, featuring advanced text-processing capabilities to facilitate seamless webpage routing and delivery of static resources.