

Darshan Sharma

+91-7009746321

thedarshansharma@gmail.com

<https://linkedin.com/in/darshansharmain>

Education

Panjab University

Bachelors in Engineering (Computer Science) 2014 - 2018

Skills

Web:	HTML5, CSS3
Frameworks:	React.Js, Node.Js , Express, Nest.Js, Flutter
Tech Stack:	MERN, LAMP, PERN
DevOps:	CI/CD, AWS- Lambda , ECS, EKS, S3, Docker, Kubernetes
Web API:	REST API, GraphQL
Database:	PostgreSQL, MongoDB, Redis
Load Balancer:	Rabbit MQ
Prog. Languages:	Java, Python, Dart, JavaScript , TypeScript

Work Experience (5+ Years)

MonkAI Technologies

Position: Senior Software Engineer

May 2020 - Dec 2023

- Improved React video player load time in the app feed by pre-loading just 25% of the content, fetching the rest only after 90% viewership of the initial segment, guided by user data analysis.
- Reduced DB expenses by 10% by integrating Redis as a caching layer, resulting in a single definitive database call rather than 6-7 heavy mongodb calls.
- Developed 9 Flutter mobile and web apps featuring in-app purchases, animations, push alerts, maps, auth. Released them on Play/App Store

Paxcom

Position: Software Engineer

May 2019 - Oct, 2019

- Decreased the load on the node server backend by 70% as it was being used for heavy CPU computations like image processing, and CSV to HTML conversion using a RabbitMQ as a load balancer.
- Achieved a 30% decrease in server expenses by transitioning from a monolithic to microservices, used RabbitMQ for load distribution, coupled with the integration of automated scaling and efficient traffic management.

Block8

Position: Software Engineer

May 2018 - May, 2019

- Improved React-Node app's speed by tweaking database queries, majorly enhancing its operational efficiency. Reduced LCP from 6s to 3s using CDN to serve static assets.

Projects

- MyStake:** Reduced loading time of web page by 20% by optimizing database queries, specifically by selecting before joining operations.
- Paxcom:** Reduced the memory usage by 70% of a CPU-intensive task of Excel to HTML conversion in Node.js by architecting a dedicated microservice separately for it.