Aditya Dhage

aditya.s.dhage@gmail.com • Los Angeles, CA - 90018 • LinkedIn • GitHub • Website

Professional Summary

Full-Stack Software Developer with **4 years** of experience designing and building **scalable**, **distributed applications**. Proficient in crafting responsive, user-friendly **front-end** interfaces with **React.js**, developing robust **back-end** systems using **Spring Boot**, **Node.js**, and various **SQL/NoSQL** databases. Comprehensive understanding of **application development** and **problem-solving** using **Java**, **JavaScript**, and **TypeScript**. Proven ability to **collaborate** with teams to execute **high-stakes projects**, ensuring **client satisfaction** in time-sensitive environments. Focused on writing clean, efficient, modular code, proactive learning and keeping up with emerging technologies.

Technical Skills

Programming	-	Java, JavaScript, TypeScript, Python(Basic Proficiency), SQL, NoSQL, Shell Scripts
Frontend	-	React.js, Next.js, HTML, CSS, TailwindCSS, Redux, Zustand, Radix-UI, Framer Motion
Backend	-	Spring Boot, Node.js, Express.js, Auth.js, JWT, OAuth, REST APIs, GraphQL APIs
Databases	-	MongoDB, PostgreSQL, Oracle DB, Prisma ORM
Cloud and DevOps	-	AWS (EKS, S3, EC2), GCP, Vercel, Webpack, Docker, Kubernetes, Kafka, Grafana, CI/CD
Tools and Misc.	-	Git, Jenkins, Figma, Maven, NPM, Generative AI, ESLint, JIRA, Microservices, Agile/SCRUM, Waterfall

Experience

Full-Stack Software Engineer **Aug 2018 - Nov 2022**

Hansen Technologies, Pune, MH, India (Duration - 4 Years and 4 Months)

- Designed, implemented, and integrated B2B telecommunications software solutions using Java, Node.js, and React.js while diagnosing and resolving critical production issues to ensure 24/7 service availability
- Overhauled a key workflow UI microservice with React, significantly improving user experience and reducing load time to 3.5 seconds
- Implemented advanced error handling and auto-correction workflows that improved overall system reliability by 5%, reducing downtime and customer support incidents
- Led a two-person team to build prototypes and proof-of-concept features, directly helping to secure a 1-year product and services contract with a major client
- Developed a microservice to automate QA processes, cutting manual effort by 20%, shortening the delivery cycle from 5 to 4 days
- Engineered a library using well-documented design patterns to streamline future system upgrades and reduce maintenance overhead
- Built and maintained CI/CD pipelines for Java and Node.js projects, streamlining multi-platform deployments on AWS EKS and improving deployment speed and consistency by 85%

Education

Master of Science in Computer Science **Jan 2023 - Dec 2024**

University of Southern California (USC), Los Angeles, CA

Bachelor of Engineering in Computer Engineering **Aug 2014 - Jun 2018**

Savitribai Phule Pune University, Pune, MH, India (First Class With Distinction)

Projects

Full-Stack Web Applications (Personal Projects) **Jan 2023 - Present**

React.js, Next.js, Node.js, Auth.js, REST/GraphQL API, PostgreSQL/MongoDB, TailwindCSS, Vercel Deployment, Figma, Zustand

- **One-Clip:** Developed a low-latency note-taking app featuring rich text format, one-click copy, and MongoDB Atlas storage with Google OAuth integration — GitHub
- **Spotify Clone:** Designed and implemented a self-upload MP3 web player inspired by Spotify's Web UI, incorporating a two-tier paid subscription model with Stripe for monetization — GitHub
- **Next-Auth Toolbox:** Engineered a robust authentication and authorization toolkit to streamline user management across multiple projects — GitHub
- **Task-It:** Built a minimalist, cross-platform task planner featuring intuitive drag-and-drop scheduling and dynamic list management, powered by a secure cloud-hosted PostgreSQL database — GitHub

Video Library Search with Video Clip Query (Academic Projects) **Dec 2023**

Python, Numpy, CV2, PyQt

- Developed a video library preprocessing algorithm (shot boundary detection, frame histogram calculation and hashing) in Python to streamline video data indexing and retrieval with an interactive desktop video player interface (PyQt5)
- Enabled querying the library using a short video clip as input, achieving precise frame matches with an average lookup time of 200–300 ms for a 100+ video database — GitHub