Aliraza Shinde

(864)-722-7967 | alishinde21@gmail.com | https://github.com/AliShinde |

Education

Clemson University, M.S. in Computer Science   **May 2026 (expected)**
Coursework: Cloud Computing Architecture, Machine Learning, Deep learning, DBMS, AI in Software dev, Information Retreival
University of Mumbai, B.E. in Computer Science   **Aug 2018 - Aug 2022**

Technical Skills

Languages:	Java (Java 17, J2EE), Python, JavaScript (ES6+), TypeScript
Backend:	Node.js, Nest.js, Spring Boot, FastAPI, Flask, REST APIs
Frontend :	React.js, Next.js, Angular, HTML, CSS, Tailwind, Bootstrap, Bulma, Vue.js
DB/Cloud:	MySQL, PostgreSQL, MongoDB, Prisma ORM, AWS (S3, EC2, Lambda, DynamoDB), Redis, NoSQL
ML/Visualization:	PyTorch, Scikit-learn, NetworkX, R,Power BI, Tableau
DevOps:	Terraform, Docker, Amazon CloudWatch, ELK, Jenkins, Kubernetes, ECS, EKS, Kafka, RabbitMQ

Work Experience

Graduate Student Assistant, *Data Visualization lab, Clemson University*   **Aug 2025 - Present**

- Created a web scraping user guide, cutting new analyst onboarding time by 20%, by using **BeautifulSoup and Selenium.**
- Analyzed **150+ library event records**, improving visibility into department and individual-level contributions, by applying Python-based EDA.
- Identified patterns in audience engagement,by applying **NLP techniques**, sentiment analysis, and content categorization on posts.
- Built visual dashboards and trend reports, improving research dissemination and stakeholder understanding, by leveraging Python (pandas, seaborn) and data storytelling methods.

System Engineer, *Infosys - Airbus*   **Oct 2022 - Dec 2024**

- Designed an automated CI/CD pipeline to reduce manual deployment using **AWS** CodePipeline, CodeBuild, and CodeDeploy.
- Integrated Apache **Kafka messaging** into microservices for reliable event-driven communication reducing data latency by **25%**
- Reduced query turnaround time by **30%** by consolidating security authentication services into a single call
- Boosted API reliability for 5 cross-functional teams by co-leading **ODATA service development** for ERP integration.
- Improved data fetch performance by **20%** by optimizing SQL queries with **CDS views** for efficient data modeling.
- Optimized query response time by 30% by using paging and sorting with **Spring Data JPA**.
- Mitigated parameter tampering attacks by route masking with **NGINX reverse-proxy**, reducing exposed endpoints by 40%
- Delivered 10+ features on schedule by working in **Agile** Scrum teams, actively contributing to **sprint planning**, backlog refinement, and retrospective meetings.
- Reduced frontend maintenance efforts by **18%** by designing and implementing **reusable React components** and applying **Redux**.

Projects

GNN text summarizer *A GNN summarizer using NetworkX and pytorch geometric*

- Built a scalable **Graph-to-Sequence summarization pipeline** on 10k+ legal/medical documents using **PyTorch Geometric GNNs + BART**, improving **ROUGE-1 accuracy by 14%**.
- Generated domain-specific embeddings, improving semantic representation of complex legal corpora by 30%, through implementing mean-pooled Legal-BERT embeddings using Hugging Face's **AutoTokenizer and AutoModel**.
- Optimized preprocessing for large-scale legal datasets, supporting reproducible training and evaluation pipelines.

YelpCamp *A full stack application using MERN stack*

- Built a full-stack campground directory app with **Express, Node.js, and MongoDB**, and authentication via **PassportJS**.
- Enhanced security by integrating **Helmet** and input sanitization, reducing session hijacking and XSS vulnerabilities by **40%**.
- Increased user engagement by **25%** by integrating geospatial mapping and real-time location services into the platform.

File Upload Module *A plug-in module using Nestjs and Cloudinary*

- Built a **NestJS project** with **mimetype validation**, preventing invalid file submissions and ensuring data integrity.
- Integrated **Cloudinary API** with configurable environment variables, enabling seamless cloud storage and scalable management.
- Implemented **Prisma ORM with Postgres** to persist upload metadata, improving searchability and auditing.

Rate Limiter with Redis *A module to add rate limiting for any api*

- Engineered a distributed **API rate limiter** with **NestJS + Redis**, supporting per-user, per-IP, and per-endpoint quotas.
- Reduced API overload incidents by **70%** by providing dynamic quota configuration and customizable request thresholds.
- Improved system reliability by adding a **/health endpoint** and containerizing the stack with **Docker Compose**.

Publications

AI in Entertainment - Using Cosine Similarity   **Jan 2023**
IEEE Xplore (Link)