

# Bharat Chundru

## Software Engineer

+1 (217) 953 5539

[chundrubharat1999@gmail.com](mailto:chundrubharat1999@gmail.com)

### SUMMARY

---

Software Engineer with 4+ years of experience in building scalable, high-performance applications using Java, Spring Boot, AWS, and React. Proficient in developing microservices, REST APIs, and event-driven architectures with Kafka. Adept at front-end development using React and Redux. Strong background in cloud deployment, agile practices, and CI/CD automation. Demonstrated ability to reduce system latency, enhance code quality, and deliver robust, production-ready solutions.

### Technical Skills

---

- **Languages:** Java 11/17, JavaScript/TypeScript, Python, SQL, Bash.
  - **Databases:** MongoDB, MySQL, Oracle DB, PostgreSQL.
  - **Frontend:** React, Redux Toolkit, Material UI, HTML5, CSS3.
  - **Backend:** Spring Boot, Node.js, Hibernate.
  - **Version Control & Integrations:** GitHub, Bitbucket, Docker, Jenkins, GitHub Actions, Nexus.
  - **Cloud:** AWS (EC2, Lambda, S3, DynamoDB, EKS).
  - **Testing:** JUnit, Mockito, Jest, Postman.
  - **Tools:** Kafka, REST, GraphQL, API Gateway, OAuth2.
  - **Development Practices:** Agile, Test-Driven Development, Behavior-Driven Development.
- 

### WORK EXPERIENCE

---

Department of Illinois Innovation and Technology (DOIT)  
Software Engineer

Springfield, IL (Remote)  
Jan 2024 – Present

- Built scalable **microservices** using **Spring Boot** and integrated asynchronous processing via **@Async** and **CompletableFuture**.
- Developed reusable **UI components** using **React**, **Redux Toolkit**, and **Material UI**, reducing development time by 25%.
- Reduced **API latency** by 60% through optimized multi-tenant query logic using **JPA Specifications** and **@EntityGraph**.
- Designed and implemented secure **REST APIs** with **OAuth2**, **JWT**, and **API versioning**, enabling secure third-party integrations.
- Utilized **Hibernate** framework for **object-relational mapping (ORM)**, including mapping entities to database tables, managing relationships between entities and optimizing database performance through **caching** and **lazy loading strategies**.
- Designed and maintained **CI/CD pipelines** using **Jenkins** and **GitHub Actions**, accelerating deployment frequency and increasing code quality across environments.
- Integrated **Docker** into **CI/CD pipelines** using tools like **Jenkins** and **GitHub Actions**, automating the build, test, and deployment processes to ensure faster and more reliable delivery.
- Configured **AWS Lambda** functions for serverless processing and integrated with **S3** and **DynamoDB** for scalable storage.
- Integrated **Kafka** for event-driven communication, improving system decoupling and resilience.
- Created unit and integration tests using **JUnit** and **Mockito**, increasing backend code coverage to 85%.
- Performed **UNIT** and **integration testing** using **Jasmine** and in writing Test Cases on every latest version before sending it for **User Acceptance Test (UAT)**

University of Illinois Springfield, Springfield, IL  
Master of Science, Computer Science, GPA : 3.8

Aug 2022 - Dec 2023

**In Technet Ltd**  
**Software Engineer**

**Aug 2021 - July 2022**

- Developed enterprise-grade **RESTful APIs** using **Spring Boot**, **JDBC**, and **Hibernate**, ensuring high performance, scalability, and reliability.
- Implemented secure authentication and authorization mechanisms using **OAuth2** and **JWT** to protect **API** endpoints and manage user access.
- Built dynamic dashboards using **React** and **React Query**, enabling real-time data synchronization from backend services for a responsive user experience.
- Integrated **Elasticsearch** to enhance search functionality, enabling faster and more accurate data indexing and retrieval.
- Accelerated deployment cycles by 50% through the creation of automated **CI/CD** pipelines using **Jenkins** and **GitHub Actions**.
- Containerized **microservices** with **Docker** and supported deployment across Linux environments, ensuring portability and consistent runtime behavior.
- Deployed and managed application builds on **AWS EC2 instances**, with continuous monitoring for uptime, performance, and reliability.
- Conducted thorough testing using **Postman** and **JMeter**, including test case development and load testing to validate system performance under stress.

**BSNL**  
**Software Engineer**

**March 2021 - June 2021**

- Developed dynamic, user-friendly dashboards using **React.js**, improving real-time monitoring of telecom systems.
- Used **Axios** and **React Hooks** to integrate **APIs** with frontend components, ensuring seamless user interaction.
- Implemented responsive design using **CSS3** and **Material UI** to support cross-device compatibility.
- Refactored backend tools using **Java multithreading** to improve concurrent telecom request processing.
- Used **Collections Framework (HashMap, ArrayList, HashSet)** to manage and manipulate in-memory data efficiently.
- Handled exceptions and ensured robust error logging using **custom exception handling**, **try-catch-finally blocks**, and **Java Logging APIs**.
- Built **RESTful** services for internal tool integration and managed secure backend communication.
- Conducted code reviews and maintained **CI workflows** using **GitHub Actions**, reducing deployment time and errors.

**APSSDC**  
**Software Engineer**

**Sep 2020 - Dec 2020**

- Designed and developed responsive front-end UIs using **React.js**, integrating **RESTful APIs** for a student training and management platform.
- Created backend services using **Node.js** and **Express.js**, enabling **CRUD** operations for training modules and user profiles.
- Wrote optimized **SQL queries** using **JDBC** to handle **Create, Read, Update, and Delete (CRUD)** operations in the database.
- Implemented business logic using **Java 8+ features** such as **Lambda expressions**, **Stream API**, and **Functional Interfaces** to write clean and efficient code.
- Implemented **Immutable Objects** and **Thread-safe Patterns (like synchronized, volatile, and ConcurrentHashMap)** to manage shared state across multiple threads.
- Used **Git SCM** for version control and collaborated via pull requests and issue tracking on **GitHub**.
- Automated build and test workflows with **GitHub Actions**, increasing deployment consistency and improving development velocity.