Contact

Phone
+91 9483126598

Email
rameshgond945@gmail,com

LinkedIn
www.linkedin.com/in/ramesh-gond

GitHub
https://github.com/RameshGond

Address
Bengalore Karnataka

Education

2019
Bachelor of Engineering
VISVAISARAYA TECHNOLOGICAL UNIVERSITY

Expertise

- Java: Core Java, OOPS, Exceptions
- Hibernate, JDBC
- Spring Boot
- MySQL
- Rest Services
- Micro Service
- JSON, XML
- Tomcat, Postman
- Git Hub
- Docker
- HTML ,CSS
- React Js

Language

Kannada

English

Hindi

Ramesh Gond
BACK-END DEVELOPER

Having 1 Years of Experience in Developing Software Programming using Java Technologies like Core java, Hibernate, Spring Boot, MySQL DB with great potential of implementing java development application software programs. Also Having Experience in Agile Methodology.

PROJECTS

Feb 2023 - current
Zhun Solution Pvt Ltd

Open Element   Jan/2023 -Aug/2023

- As part of the project, Zhun Solution develops a scalable low-code platform as a service based on an event-driven architecture, the solution-architecture is based on asynchronous event-driven micro-profile services based on eclipse-micro profile & spring-boot based rest micro services backed by a decentralized in-memory data grid based on Apache Ignite.
- custom-elements integrating to the micro services backend over the standard web-socket protocol.

CAAS Application   sep/2023 -Present
Java Developer

CAAS is a product catalog management system that allows users to create and manage their own product catalogs and definitions a runtime. With CAAS, users can easily add, edit, or delete attributes and attribute definitions for their products, as well as configure various settings and options for their catalogs.

My Roles and responsibilities:

- Developed spring-boot based rest-endpoint to provide crud and business logic. We used MySQL as a database.
- During this we developed around 2-3 various APIs in catalog service. We used the MySQL DB . We developed a criteria builder which creates criteria based on input received and gives it to the aggregation framework. Also, we used reactive programming and Reactive Streams
- Developing Java-Functional Interface based Predicates to support checks on the entity-objects.