

Roki Roy

Siliguri | rokiroydev@gmail.com | +91 9800 881 300 | rokiroy.dev | linkedin.com/in/rokiroy |
github.com/royroki

Summary

- Full-stack developer with hands-on experience shipping production Flutter, React, and Golang systems. Strong in clean architecture, backend performance optimization, and **real-world app delivery** (web, mobile, Apple Watch).

Experience

SDE 1, Fordel Studios · (Onsite)

- Led end-to-end Flutter app development using clean architecture, supporting 5,000+ active users with 98% crash-free sessions; currently developing v2 with advanced features.
- Coordinated cross-functional teams (design, backend, QA) across multiple initiatives to deliver features on schedule using Agile workflows.
- Designed and implemented Golang backend services, reducing API response time by 25% and improving system reliability.
- Optimized a React-Node.js e-commerce platform by integrating complex third-party APIs and reducing page load times by 30%.
- Developed and shipped an Apple Watch app in Swift, extending the product ecosystem to wearable platforms.

Full Stack Developer Intern, Fordel Studios · (Onsite)

- Optimized backend APIs and database queries, improving data retrieval performance and supporting new frontend features.
- Participated in Agile sprints, code reviews, and bug triage, contributing production-ready fixes.

Projects

ZenliAI – AI-Powered Content Generation Platform

- Built full-stack platform enabling users to generate eBooks, SVG, clipart, and SEO content with integrated Pinterest automation for pin upload.
- Architected complex backend in Golang with GenAI API integration and S3 for asset storage.
- **Stack:** Golang, React, GenAI APIs, AWS S3, Pinterest API

BattleManager – Tournament Management PWA

- Built a tournament management PWA with real-time updates and messaging using React, Firebase, and Capacitor.
- Implemented real-time messaging system with Firebase for instant tournament updates and player communication.

LetsGo – Tag-Based Chatting App

- Built real-time chat app using Golang microservices.
- Used gRPC + Protocol buffers for sub-200ms message latency.
- **Stack:** Golang, Microservices, gRPC, Protobuf, Docker

Technologies

Primary: Golang, React, Flutter, PostgreSQL, Docker

Secondary: Java (Spring Boot), JavaScript (ES6+), TypeScript, Swift, Firebase, AWS, gRPC

Education

University of North Bengal, BCA (pursuing)