# Project Experience

## ClaimConnect – Insurance Claim Processing System

Full Stack Developer | [Company Name], [Location] | [Month, Year] – Present

* Developed an end-to-end insurance claim processing platform using Spring Boot Microservices and React, enabling role-based workflows for hospitals, patients, and insurance companies.
* Implemented four modular microservices:  
  - Hospital Service – allows hospitals to log in and initiate claim requests on behalf of patients.  
  - Patient Service – enables patients to view claim requests and update their status (Accept / Reject / Revert).  
  - Insurance Company Service – processes claims only after patient approval, allowing insurance officers to approve valid claims.  
  - Claim Request Service – acts as the central orchestrator handling end-to-end claim lifecycle.
* Used Eureka Server for service discovery and Spring Cloud Gateway for secure, unified API routing with JWT-based authentication.
* Managed centralized configuration for all services using Spring Cloud Config Server backed by a GitHub repository.
* Enabled inter-service communication through Open Feign, ensuring efficient RESTful interactions.
* Built an interactive frontend using React, Redux, and Axios, offering role-specific views and operations.
* Enforced role-based access control (RBAC) to segregate functionalities based on the logged-in user type.
* Secured all endpoints using Spring Security and JWT, supporting token-based authentication and authorization.
* Used MySQL for persistent storage across services with proper entity mapping and indexing.
* Containerized microservices using Docker, with deployment support for Kubernetes environments.

Tech Stack:

React, Redux, Bootstrap, Spring Boot, Spring Cloud (Eureka, Gateway, Config Server), Feign Client, JWT, MySQL, Docker, Kubernetes, GitHub