

CAREER PROFILE

Senior Technical Architect with over 18 years of hands-on experience in designing, developing, and architecting large-scale, cloud-native, and microservice-based enterprise applications. Expert in reactive systems, event-driven architecture, domain-driven design, and performance engineering. Proven leadership in technical governance, scalability optimization, and DevSecOps enablement across diverse industries including e-commerce, telecom, utilities, and sustainability. Skilled in mentoring teams, conducting architecture reviews, and delivering business-aligned technical solutions.



Virendra Kumar Mishra

Principal Software Architect

 virend70@rediffmail.com

 +91 9582176478

 Indian

 IST (UTC+05:30)

 virendra

 Resume PDF

EDUCATION

B.Tech (Computer Science & Engineering)

BBD (Uttar Pradesh Technical University)

2004 - 2008

LANGUAGES

English (Professional)

Hindi (Native)

INTERESTS

football

Badminton

Computers

EXPERIENCES (18 YEARS)

SAP, Bangalore

Architect

July,2016 - Present

Roles and responsibilities :-

- New projects evaluation
- Requirement gathering & Vision scope
- Planning and Estimation of requirements
- Client discussion for major functionality design or issue discussion
- Design and Development of application functionalities
- Individual contributor
- Hybris SME
- Project Code reviewer
- Internal/external interviewer

Sapient, Noida

Senior Associates

Roles and responsibilities :-

- Module lead
- Hybris SME
- Individual contributor
- Code reviewer
- Working on stories with unit and integration test

Dec,2012 - June,2016

Centurylink

Software Engineer

Dec,2010 - Nov,2012

Roles and responsibilities :-

- Individual contributor
- Story implementation
- Unit test & integration test
- Peer review

Evolko System Pvt Ltd

Software Engineer

Oct,2008 - Sep,2010

Roles and responsibilities :-

- Individual contributor
- Story implementation
- Unit test & integration test
- Peer review

SKILLS & PROFICIENCY

Java



Spring

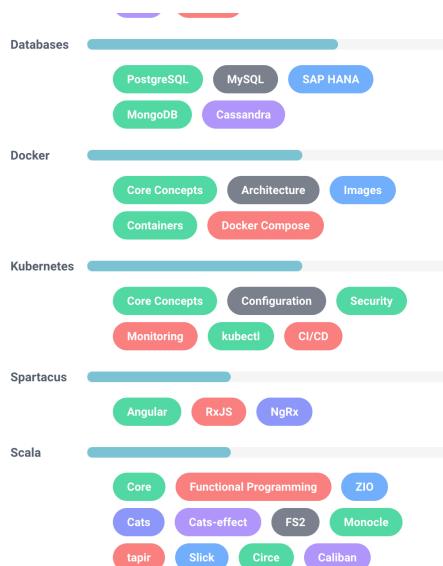


Hybris



Node





PROJECTS

Below are the list of projects in opposite chronological (from latest to oldest) -

Life Science Product

- Automate measuring and reporting on the carbon emissions associated with your company's corporate travel. Then, you can address these emissions by supporting verified, high-impact carbon reduction (through Sustainable Aviation Fuel), removal, or offset solutions. The Climate App is built on SAP BTP, ensuring you manage your emissions in the same SAP solutions and workflows in which you already manage your other business processes.

Architecture Highlights :-

- Reactive Microservices: Built with Spring Boot (WebFlux) and Node.js, leveraging Project Reactor for non-blocking, reactive flows.
- Event-Driven Architecture: Utilized Kafka and RxJS for asynchronous inter-service communication.
- Multitenancy Support: Tenant-based data isolation using Redis caching and contextual request scoping.
- Resiliency Patterns: Implemented retry, circuit breaker, and fallback mechanisms using Resilience4j.
- Chaos Engineering: Performed failure injection tests with Chaos Monkey to validate resilience.
- Cloud-Native Deployment: Hosted on SAP Cloud Foundry, integrated CI/CD pipelines using Jenkins + Piper.

Technology Highlights :-

- Converted blocking APIs to non-blocking reactive endpoints with WebFlux and R2DBC, improving concurrent user handling.
- Redis-based caching for static metadata and frequently accessed tenant information to cut DB roundtrips.
- Tuned Kafka consumer parallelism and batch sizes for high-throughput event processing.
- Implemented autoscaling policies in Cloud Foundry to scale pods dynamically based on CPU/memory thresholds.
- Configured G1GC tuning and thread pool optimizations to minimize GC latency under high load.
- Load testing & Chaos Testing
 - JMeter and K6, validating SLAs under 5K concurrent users.
 - Simulated high-volume data ingestion using K6
 - Validated fault tolerance under simulated Kafka node failures using chaos monkey
 - Autoscaling policies in Cloud Foundry based on CPU/memory thresholds.
 - Event-driven asynchronous flows allow horizontal service scaling.
 - Multitenant architecture isolates tenants for independent scaling.
 - Redis caching reduces DB contention during spikes.

Roles & Responsibilities :-

- Owned end-to-end architecture, from domain modeling and service decomposition to deployment topology.
- Defined microservice interaction contracts (API-first approach).
- Designed observability strategy - logs (ELK), metrics (Dynatrace), and traces (OpenTelemetry)
- Conducted architecture reviews and code audits to ensure performance, scalability, and best practices.
- Mentored development teams on reactive programming and event-driven design principles.
- Drove DevOps enablement, aligning CI/CD and monitoring with architectural vision.

Technology :- SAP BTP cloud (Business Technical Platform) , SAP CAP, Node js, Event Messaging framework (AMQP), SAP HANA, ODATA V4

Choose Climate App

- Automate measuring and reporting on the carbon emissions associated with your company's corporate travel. Then, you can address these emissions by supporting verified, high-impact carbon reduction (through Sustainable Aviation Fuel), removal, or offset solutions. The Climate App is built on SAP BTP, ensuring you manage your emissions in the same SAP solutions and workflows in which you already manage your other business processes.

Architecture Highlights :-

- Service-Oriented Workflow: Designed modular microservices for Request Management, Approval Workflow, and Payroll Integration.
- Spring Boot Microservices: Implemented secure and stateless REST APIs using Spring Boot and Spring Security.
- SAP CAP Integration: Leveraged SAP Cloud Application Programming Model (CAP) to connect SuccessFactors with backend approval logic.
- API Gateway & Security Layer: Introduced OAuth2 authentication and token-based authorization for SuccessFactors API access.
- Event-Driven Integration: Used Kafka to asynchronously publish status events from approval workflows to the payroll microservice.
- Cloud-Native Architecture: Deployed on SAP Cloud Foundry with configuration-driven scalability and service isolation.

Technology Highlights :-

- API Performance Optimization: Cached user and allowance metadata in Redis to avoid repetitive API calls to SuccessFactors. Introduced response compression and pagination for large data sets.
- Workflow Engine Tuning: Preloaded workflow configurations to eliminate startup lag. Asynchronous event handlers for non-critical tasks (e.g., audit trail logging).
- Database Performance: Connection pooling (HikariCP) and optimized query structure using JPA fetch tuning.
- Performance Testing: Used JMeter to measure end-to-end workflow latency; reduced response times from 1.2s to under 600ms.
- Auto-Scaling via Cloud Foundry: Configured horizontal scaling for API services based on CPU and latency metrics.
- Stateless Service Design: Ensured all REST services were stateless to allow seamless scaling across instances.
- Asynchronous Communication: Kafka message queues used to decouple workflow and payroll services, improving elasticity under load.
- Centralized Configuration Management: Leveraged Spring Cloud Config for environment-based scaling and resource tuning.

Roles & Responsibilities :-

- Defined end-to-end architecture blueprint, aligning SAP CAP services with Spring Boot microservices.
- Led the integration design between SuccessFactors and external systems using OData APIs.
- Supervised API security and authentication strategy using OAuth2 and token lifecycles.
- Conducted code reviews and architecture validations ensuring adherence to best practices.
- Collaborated with business analysts and payroll teams to model approval workflows using event-driven orchestration.
- Guided development team on CI/CD pipelines, automated deployments, and rollback strategies in Cloud Foundry.

Technology :- SAP BTP cloud (Business Technical Platform) , SAP CAP, Node js, Event Messaging framework (AMQP), SAP HANA, ODATA V4

CISN (Construction Site Logistics Network) App

- It provides a single platform for supplier (Material supplier) and consumer (Constructor) where they can directly connect to each and create order for materials for construction. Deliveries can be chased and in case of issue with any material, it can be replaced.

Architecture Highlights :-

- SaaS Multitenant Microservice Architecture with isolated tenant context.
- Event-Driven Communication: Kafka-based messaging among supplier, order, and logistics services.
- Domain-Driven Design (DDD): Clear bounded contexts separating supplier management, order management, and delivery tracking domains.
- CI/CD Automation: Used SAP MTA (Multi-Target Application) deployment model with Jenkins pipelines.

Technology Highlights :-

- Implemented rate limiting and response caching in API gateway layer using Redis.
- Optimized Kafka batch message consumption and parallel partition handling for large event volumes.
- Adopted connection pooling and DB sharding per tenant for better resource isolation.
- Tuned GC and memory footprint for both Java and Node.js containers, cutting startup latency by 35%.
- Integrated Dynatrace APM for latency tracing, enabling early detection of service-level bottlenecks.
- Cached Maven artifacts and optimized Docker layers in build pipeline, improving deployment time by ~40%.
- Horizontal replication of microservices per tenant based on load.
- Database sharding per tenant + connection pooling for resource isolation.
- Event-driven architecture supports adding more consumers to scale throughput.
- CI/CD pipeline and containerization enable rapid scaling of service instances.

Roles & Responsibilities :-

- Defined multi-tenant SaaS architecture blueprint and service ownership boundaries.
- Led cross-functional team across multiple geographies for supplier module delivery.
- Established Kafka schema registry standards and message versioning guidelines.
- Supervised performance engineering cycles and load/stress test benchmarking.
- Drove architecture governance, reviewing code quality and adherence to DDD principles.
- Collaborated with Product Owners to translate domain needs into scalable microservice designs.

Technology :- SAP BTP cloud (Business Technical Platform) , SAP CAP, Node js, Event Messaging framework (AMQP), SAP HANA, ODATA V4

Samsung

- Samsung B2C website which deals with selling samsung products to directly customers. It is across the world from UK , UAE, Asia, Russia. There is single codebase to deal all location. Any location specific changes are maintained in same codebase. It contains many other system interactions and many payment providers.

Architecture Highlights :-

- Reactive Microservices: Combined Spring Boot, Hybris, and Angular (Spartacus) PWA architecture

- CQRS Implementation: Separated command and query pipelines to improve scalability and consistency.
- Micro-Frontend Design: Modular Angular applications with state management via NgRx and RxJS.
- Containerized Deployment: Docker + Kubernetes clusters across multiple data centers.
- Dynamic Rule Engine: Pricing and promotions managed through Drools microservices.

Technology Highlights :-

- Reactive WebFlux migration reduced synchronous call latency and improved concurrent request handling by ~40%.
- Tuned Solr indexing and Redis caching for popular search queries and product details.
- Used Content Delivery Network (CDN) for static assets and geo-distributed K8s clusters for low-latency delivery.
- Pre-compiled Drools rules into memory to eliminate runtime parsing overhead.
- Implemented lazy loading, bundle optimization, and NgRx memoization on frontend for better page load times.
- Introduced Prometheus + Grafana dashboards to visualize service latencies and autoscaling metrics.

Roles & Responsibilities :-

- Designed multi-region architecture ensuring data sovereignty and minimal latency for global traffic.
- Architected CQRS + Event Sourcing model for order and inventory management.
- Defined API orchestration layer for integrating multiple regional payment providers.
- Directed performance and scalability tuning workshops with Hybris and DevOps teams.
- Collaborated with product managers and business analysts to design country-specific customization strategy within a single codebase.
- Enforced DevSecOps practices, integrating vulnerability scans and automated quality gates in pipelines.

Technology :- SAP Hybris, Spartacus, Java 8, Spring

Mercury NZ

- Mercury is New Zealand electricity and gas provider. Mercury B2C Hybris implementation platform where customer can join for electricity and/or gas. It gets details from customer and provides offers. These offers are very configurable based on different parameters like location / customer index etc. These offers are created by Mercury business team to attract more business.

Architecture Highlights :-

- Service-Oriented B2C Platform: Decoupled microservices for offers, customers, and billing.
- Authentication Microservice: Implemented OAuth2 and SAML flows for secure multi-tenant access.
- Containerization: Dockerized deployments orchestrated using Kubernetes.
- Security Gateway: Used Spring Security and JWT-based service-to-service authentication.

Technology Highlights :-

- Token caching to prevent repeated authentication calls between microservices.
- Redis cache for rule configuration and user profile data, minimizing DB access.
- Implemented read-replica strategy for PostgreSQL to handle high read workloads.
- Fine-tuned Kubernetes Horizontal Pod Autoscaler (HPA) to scale with seasonal demand spikes.
- Introduced GZIP compression, ETag headers, and response caching for REST APIs.
- Conducted load testing via JMeter and Neoload; optimized JVM parameters for consistent GC performance.
- Geo-distributed Kubernetes clusters for low-latency global delivery.
- Horizontal scaling of microservices per region based on traffic patterns.
- CDN for static content reduces load on origin servers.
- CQRS + Event Sourcing allows independent scaling of read/write pipelines.

Roles & Responsibilities :-

- Architected identity management integration with Mercury's IDP and Microsoft ADFS.
- Owned end-to-end design for authentication and authorization flows (OAuth2, SAML).
- Guided DevOps team in Kubernetes-based container orchestration and rollout strategies.
- Directed cross-team performance reviews, defining KPIs and SLA compliance matrices.
- Mentored 8–12 developers, conducted code and design reviews to maintain architectural integrity.
- Coordinated directly with client stakeholders to define roadmap and technical risk mitigation.

Technology :- SAP Hybris, Java 8, Spring, OAuth Authentication

Asahi (Beverage Company)

- Its B2b platform where it sells beverages (alcohol and non-alcohol) to different companies.

Architecture Highlights :-

- Domain-Driven Design (DDD): Independent microservices for Product Catalog, Order Processing, Payment, and Promotion Management.
- Event-Driven Architecture: Used Kafka to publish order status and inventory events across modules.
- Payment Integration Microservice: Hosted Order Page (HOP) implemented as a standalone payment microservice.
- Rule Engine Service: Implemented business rules using Drools for cart promotions and discount logic.
- Assisted Service Module: Separate microservice layer for customer support staff to intervene in customer sessions.
- Containerized Microservices: Dockerized Spring Boot applications orchestrated using Kubernetes.
- Observability: Centralized monitoring through Prometheus, Grafana, and ELK stack.

Technology Highlights :-

- Optimized Drools rule execution by precompiling rules and caching session factories.
- Implemented Redis caching for product catalog and promotions to reduce database I/O latency.
- Batched order updates in Kafka to prevent queue saturation during sales events.
- Migrated blocking I/O to asynchronous calls using Spring WebFlux for cart updates and pricing calculations.
- Applied connection pooling and query indexing in MySQL to enhance transaction throughput.
- Load testing performed with Neoload, achieving 35% latency reduction under peak load.
- Kubernetes Horizontal Pod Autoscaling (HPA): Configured auto-scaling for order and payment microservices based on CPU/memory usage.
- Database Read Replicas: Introduced MySQL read replicas for parallel read operations.
- Event Partitioning: Kafka topic partitioning based on region and distributor type.
- Stateless Deployments: All services designed stateless, ensuring linear horizontal scalability.

Roles & Responsibilities :-

- Led architecture and technical design for B2B e-commerce workflows.
- Oversaw the integration of Drools-based rule engine and custom payment gateway APIs.
- Mentored developers on Spring Boot best practices and microservice design patterns.
- Managed code reviews, security audits, and CI/CD integration via Jenkins.
- Coordinated with product teams to ensure regulatory compliance for alcohol sales.
- Collaborated with infrastructure teams to tune Kubernetes and Redis clusters for high availability.

Technology :- SAP Hybris, Java 8, Spring.

Heritage Cananda

- B2b platform where it sells OEM parts to companies.

Architecture Highlights :-

- SAP Hybris Commerce Backend decomposed into independent Spring Boot microservices for Catalog, Pricing, and Punchout Integration.
- Punchout Integration Service: Implemented integration with Birchstreet Procurement System, allowing seamless supplier catalog access.
- API Gateway with OAuth2 Security: Ensured secure communication between client and internal microservices.
- Containerized Deployments: Dockerized Hybris components with Kubernetes orchestration for CI/CD delivery.
- Data Caching Layer: Redis for catalog metadata and frequently accessed SKUs.

Technology Highlights :-

- Optimized API response caching for catalog and search services.
- Implemented lazy loading and query optimization for catalog pagination.
- Configured connection pooling and reduced DB query execution time by 25%.
- Introduced asynchronous Punchout calls to reduce synchronous API latency.
- Load tested system using JMeter, validating stable performance under concurrent B2B order traffic.
- Container Autoscaling: Kubernetes-based pod scaling per microservice workload.
- API Gateway Caching: Implemented rate-limiting and in-memory cache for repetitive catalog lookups.
- Replication for Search Services: Horizontal scaling for Redis nodes to handle concurrent Punchout requests.
- High Availability Setup: Multi-zone deployment across clusters for fault tolerance.

Roles & Responsibilities :-

- Designed and documented Punchout Integration architecture with Birchstreet.
- Conducted technical workshops for B2B customer integration and onboarding.
- Defined service interaction contracts (Swagger/OpenAPI).
- Oversaw deployment strategy and CI/CD pipeline automation.
- Mentored a distributed team of 6 developers, ensuring consistent coding standards.
- Led architecture review sessions with SAP and Heritage IT leadership.

Technology :- SAP Hybris, Java 8, Spring, B2B Punchout implementation with Birchstreet

Clas Ohlson

- This project is regarding selling the kitchenware, luggage and travelware, electrical tools, household electrical items etc. It is complete ecommerce implementation using one of top rated ecommerce software

Architecture Highlights :-

- Microservice Decomposition: Core services for Cart, Checkout, Promotions, Inventory, and Search.
- Reactive Cart Module: Used Spring WebFlux for non-blocking cart operations.
- Promotion Engine: Integrated with Drools for dynamic discounts and coupons.
- Search Optimization: Implemented Solr-based product search with Redis caching for top queries.
- Security Layer: OAuth2-based authentication and token management microservice.
- Deployment: Dockerized microservices orchestrated via Kubernetes clusters.

Technology Highlights :-

- Introduced Redis session caching to improve cart retrieval performance.
- Optimized Solr index updates using delta indexing strategy.
- Replaced synchronous REST calls between cart and checkout with Kafka event-driven updates.
- Applied connection pooling (HikariCP) and database read replicas for better query throughput.
- Conducted JMeter load testing with emphasis on cart and checkout paths, reducing latency by 30%.
- Horizontal Scaling: Kubernetes HPA for checkout and inventory services.
- Elastic Redis Cluster: Expanded Redis nodes dynamically during promotional traffic spikes.
- Microservice Caching: Caching of promotion rules and frequently queried products.
- React Frontend: Anarchaeid SPA consuming REST API synchronously

- Decoupled Frontend: Angular based on a consuming REST API architecture for high responsiveness.

Roles & Responsibilities :-

- Owned cart and checkout architecture, ensuring consistent transaction flows.
- Designed promotion rule framework for dynamic campaign management.
- Led team of 8 developers; performed code reviews, deployment oversight, and sprint planning.
- Coordinated with product owners to translate business rules into Drools decision tables.
- Defined SLIs/SLOs for cart response time and checkout throughput in collaboration with DevOps.

Technology :- SAP Hybris, Java 8, Spring, Endeca

TUI

- This project is regarding the selling of holiday tours to user. This site was for Nordic countries (Sweden, Norway, Finland and Denmark). Tours are shown as country wise but user could be able to search for any destination, resort or hotel. Different offers are also displayed time to time which is maintained by different module. TUI is a multinational travel and tourism company. It is the largest leisure, travel and tourism company in the world, and owns travel agencies, hotels, airlines, cruise ships and retail stores. The group owns six European airlines - the largest holiday fleet in Europe - and UK-based tour operator Thomson.

Architecture Highlights :-

- Microservices for Travel Search and Booking: Separate services for Flight Search, Booking, Offers, and Payment.
- Search Service: Integrated Apache Solr and Lucene with Spring Integration for fast indexing of flight data.
- Workflow Microservices: Decoupled product and content workflows using event-driven orchestration.
- Caching Layer: Redis for flight and offer lookups.
- Containerized Hybris Platform: Deployed on Docker + Kubernetes infrastructure.

Technology Highlights :-

- Tuned Solr schema and index merge factors to improve query response time by ~40%.
- Cached high-demand routes and offers in Redis for instant lookup.
- Parallelized flight data ingestion using Spring Integration pipelines.
- Applied database query batching and pagination to improve backend throughput.
- Conducted performance validation using Neoload for flight search latency SLAs.
- Distributed Indexing: Partitioned Solr indexes per country to improve query speed.
- Regional Kubernetes Clusters: Deployed services regionally to reduce latency.
- Dynamic Cache Expansion: Redis cluster auto-scaled based on memory thresholds.
- Asynchronous Job Scheduling: Background indexing handled through message queues for scalability.

Roles & Responsibilities :-

- Led search module design and integrated Solr/Lucene with flight ingestion systems.
- Defined multi-region deployment architecture for Nordic countries.
- Directed performance tuning efforts for search and booking APIs.
- Mentored content/product teams on workflow microservice usage.
- Managed stakeholder communication and aligned releases across regional teams.

Technology :- SAP Hybris, Java 8, Spring, Solr, Lucene, Endeca

LadBrokes

- Ladbrokes is a British-based betting and gambling company. It is based in Rayners Lane in Harrow, London. It is betting site where different ongoing games/events is shown with different odds available for betting. These odds gets changed an game progress. User could also able to see the events on the event page. It also has the odds for future events with odds define on pro-rated basis.

Architecture Highlights :-

- Event-Driven Real-Time System: Java/Spring-based microservices processing live sports data feeds.
- Caching Layer: Used Oracle Coherence for real-time odds caching and updates.
- Reactive UI & Data Streams: Real-time updates delivered via WebSockets and reactive streams.
- Search Service: Endeca integration for historical event searches and user betting history.
- Security & Compliance: OAuth2-based API security and audit trail logging microservices.

Technology Highlights :-

- Implemented event buffering and batching to handle concurrent odds updates efficiently.
- Tuned Oracle Coherence cache size and distributed data across cluster nodes for faster lookups.
- Introduced async event publishing to decouple UI updates from backend odds computation.
- Optimized Endeca indexing frequency for faster retrieval of archived data.
- Conducted load tests simulating 100K+ concurrent live events using JMeter.
- Clustered Coherence Nodes: Deployed cache nodes with elastic scaling during live sporting events.
- Event Queue Scaling: Implemented message queue partitions for each sport category (Football, Tennis, etc.).
- Microservice Isolation: Deployed independent microservices for odds, bets, and results.
- Geo-distributed Deployment: Multi-region failover setup to ensure high availability and redundancy.

Roles & Responsibilities :-

- Designed real-time event architecture integrating data feeds, odds engine, and user dashboards.
- Defined API contracts between odds, betting, and payment services.
- Led 10+ member development team for real-time odds and betting module delivery.
- Oversaw performance tuning and load testing for peak sports seasons.
- Implemented DevSecOps pipelines ensuring secure deployments and

- observability integration.
- Worked with business stakeholders to define SLAs for live betting response times and uptime.

Technology :- SAP Hybris, Java 8, Spring, Oracle coherence, Endeca

Centurylink E-commerce Site

- This project is regarding the selling of holiday tours to user. This site was for Nordic countries (Sweden, Norway, Finland and Denmark). Tours are shown as country wise but user could be able to search for any destination, resort or hotel. Different offers are also displayed time to time which is maintained by different module. TUI is a multinational travel and tourism company. It is the largest leisure, travel and tourism company in the world, and owns travel agencies, hotels, airlines, cruise ships and retail stores. The group owns six European airlines - the largest holiday fleet in Europe - and UK-based tour operator Thomson.

Technology Highlights :-

- Network Services: Provides high-speed internet through fiber-optic and DSL technologies, offering speeds up to 940 Mbps.
- Self-Healing Network Service (SHNS): A service designed to automatically detect and reroute traffic in case of network failures, ensuring minimal downtime.
- Metro Ethernet: Offers scalable Ethernet solutions for metropolitan areas, supporting various bandwidths and interfaces to meet diverse customer needs.
- Cloud and Security Solutions: Delivers cloud services encompassing public and private cloud, multi-cloud management, and security services, including DDoS protection and threat intelligence.

Technology :- Struts, Spring , Jquery , vm

Evolko Hospital Management System

- It contains a full fledged patient Care System. where doctor can handle all the patient detail using software . This is designed to assist doctor in their everyday task. A strong prescription writer with following features allows doctor to have a paperless office in no time. App connected to Labs network and send and get investigation reports.

Technology Highlights :-

- Remote Patient Monitoring: Utilizes IoT devices to monitor patients' health metrics in real-time, enabling timely interventions.
- AI-Driven Analytics: Employs machine learning algorithms to analyze patient data, providing actionable insights for personalized care plans.
- Teleconsultation Platform: Offers a secure communication channel for patients and healthcare providers, supporting video consultations and digital prescriptions.
- Clinical Knowledge Integration: Integrates a comprehensive clinical knowledge base to assist healthcare professionals in decision-making processes.

Technology :- Java, EJB 3, Adobe Flex 3 , Actionscript 3

STRENGTH

Determinant ,quick learner , always interested for learning new technologies, Confident , having Strong Willpower , Hard worker, Honest