

# TECHNOLOGY STACK

## To Supply Leftover Food to Poor - Technical Architecture & Technology Stack

### Project Design Phase-II: Technology Stack & Architecture

**Date:** November 1, 2025

**Team ID:** NM2025TMID08527

**Project Name:** To Supply Leftover Food to poor

**Maximum Marks:** 4 Marks

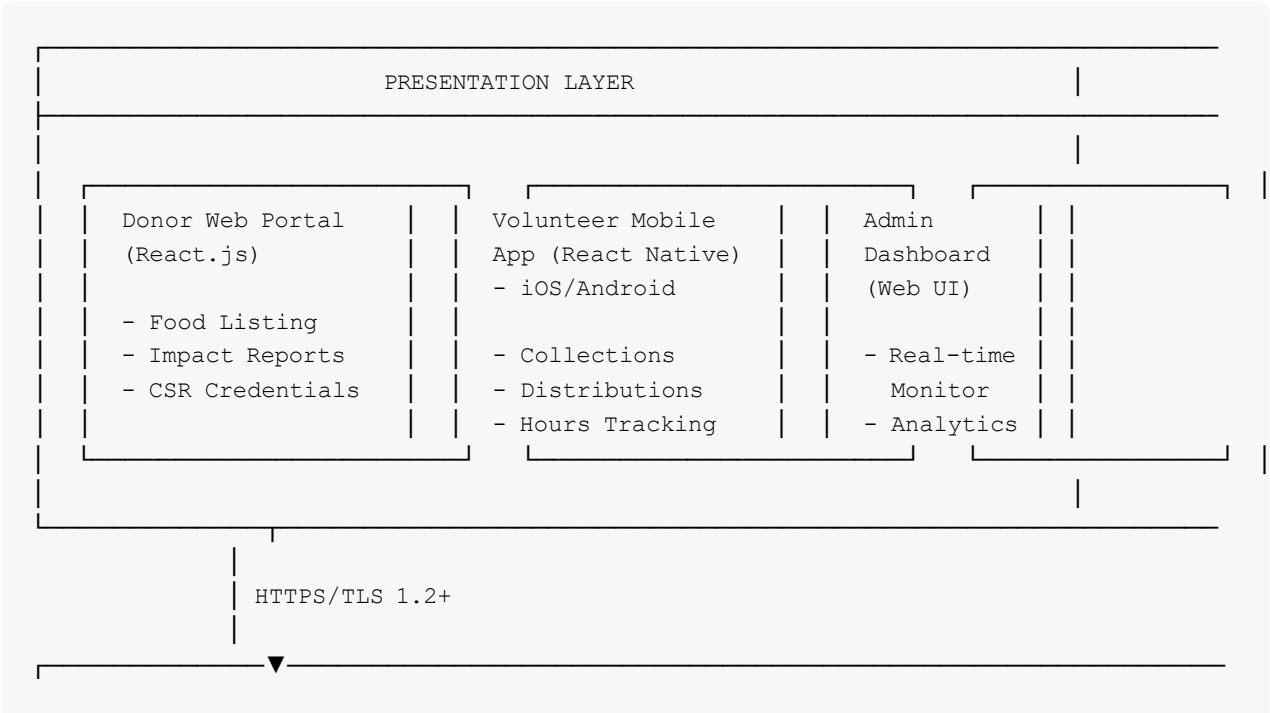
### TECHNICAL ARCHITECTURE

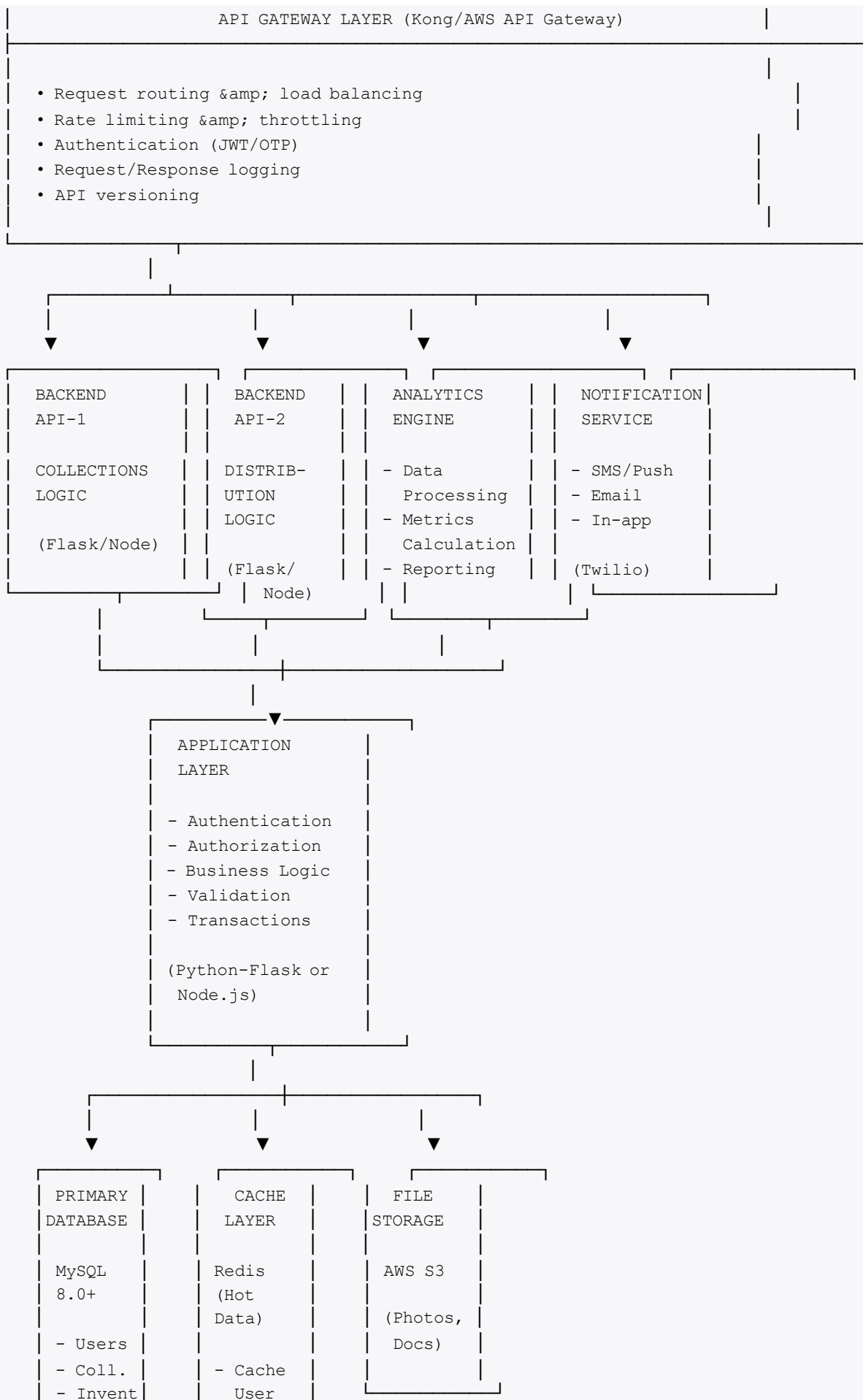
#### Architecture Overview

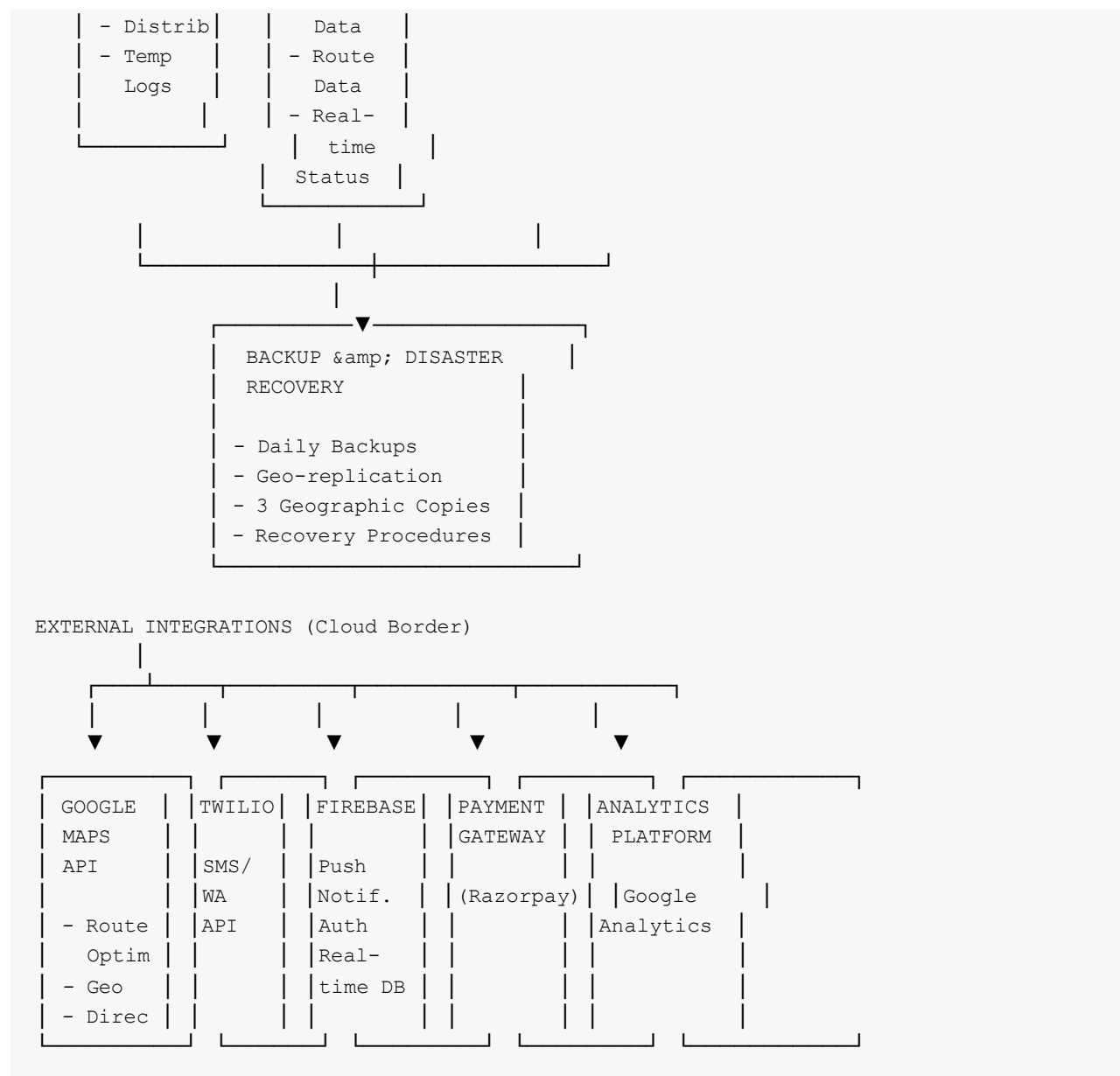
The food supply redistribution system is built on a **cloud-native, microservices architecture** designed for scalability, reliability, and ease of maintenance. The system is demarcated into:

- **Local Components:** Volunteer mobile devices, hub facility systems
- **Cloud Components:** Backend servers, databases, APIs, analytics
- **External Integrations:** Google Maps, SMS/WhatsApp APIs, payment gateways
- **Data Storage:** Cloud databases with backup, distributed caching
- **Machine Learning:** Optional future expansion for route optimization and demand forecasting

#### Architecture Diagram







**TABLE-1: COMPONENTS & TECHNOLOGIES**

S.No	Component	Description	Technology
1	<b>Donor Web Portal</b>	Frontend interface for restaurants, caterers, and donors to list food, track donations, view impact	React.js 17+, TypeScript, Bootstrap 5, Redux State Management
2	<b>Volunteer Mobile App</b>	Native mobile application for iOS and Android for collections and distributions	React Native, Expo (for rapid development), Native modules for camera/GPS
3	<b>Operations Dashboard</b>	Real-time dashboard for hub managers and coordinators showing collections, distributions, metrics	React.js, D3.js/Chart.js for data visualization, <a href="#">Socket.io</a> for real-time updates
4	<b>API Gateway</b>	Central entry point for all API requests with routing, authentication, rate limiting	Kong or AWS API Gateway with load balancing, SSL termination

<u>S.No</u>	Component	Description	Technology
5	<b>Collection Microservice</b>	Business logic for handling collection assignments, verification, and completion	Python-Flask or Node.js with Express, validates all collection data
6	<b>Distribution Microservice</b>	Business logic for route optimization, distribution assignments, beneficiary tracking	Python-Flask or Node.js with route optimization algorithms
7	<b>Analytics Engine</b>	Real-time processing of operational metrics and impact calculations	Apache Kafka for event streaming, Apache Flink or Spark for real-time processing
8	<b>Notification Service</b>	Sends SMS, WhatsApp, email, and push notifications to users	Twilio API for SMS/WhatsApp, Firebase Cloud Messaging for push, SendGrid for email
9	<b>Authentication Service</b>	Handles user authentication (OTP for volunteers, Email/Password for donors/admins)	JWT tokens, Auth0 or Firebase Authentication, OTP via Twilio
10	<b>Food Safety Module</b>	Implements HACCP procedures, critical control point monitoring, compliance tracking	Custom module with business rules, alert system for violations
11	<b>Primary Database</b>	Central data store for all operational data (users, collections, distributions, inventory)	MySQL 8.0+ or PostgreSQL 13+ with replication for high availability
12	<b>Cache Layer</b>	High-speed in-memory cache for frequently accessed data to improve response times	Redis 6.0+ with cluster mode for scalability, pub/sub for real-time features
13	<b>File Storage</b>	Cloud storage for photos, documents, compliance records, backups	AWS S3 with lifecycle policies, CloudFront CDN for fast delivery
14	<b>Temperature Monitoring System</b>	IoT integration for hub refrigeration and vehicle temperature sensors	Raspberry Pi/Arduino with temperature sensors, MQTT protocol for communication
15	<b>GPS Tracking</b>	Real-time tracking of volunteer vehicles during collections and distributions	Google Maps API with real-time location services, vehicle tracking software
16	<b>Payment Gateway</b>	Integration for accepting donations and processing corporate sponsorships	Razorpay or Stripe with PCI-DSS compliance, webhook integration
17	<b>Analytics &amp; Reporting</b>	Aggregates data for impact reporting, financial analysis, and operational metrics	Google Analytics for web tracking, custom reporting engine, data warehouse
18	<b>Backup &amp; Recovery</b>	Automated backup system with disaster recovery capabilities	AWS S3 with cross-region replication, daily snapshots, RTO <4hrs, RPO <1hr
19	<b>Message Queue</b>	Asynchronous task processing and event-driven architecture	Apache RabbitMQ or AWS SQS for decoupling services, batch processing
20	<b>Logging &amp; Monitoring</b>	Centralized logging and system health monitoring	ELK Stack (Elasticsearch, Logstash, Kibana) or AWS CloudWatch

<u>S.No</u>	Component	Description	Technology
21	CI/CD Pipeline	Automated testing and deployment pipeline	Jenkins or GitLab CI/CD with Docker containerization
22	Infrastructure (Cloud)	Managed cloud hosting with auto-scaling, load balancing	AWS (EC2, RDS, ElastiCache, S3) or Google Cloud Platform with Kubernetes

**TABLE-2: APPLICATION CHARACTERISTICS**

<u>S.No</u>	Characteristics	Description	Technology
1	Open-Source Frameworks	Foundation of backend services for rapid development and community support	Python: Flask (micro-framework), SQLAlchemy ORM; Node.js: Express, Mongoose/Sequelize
		Frontend frameworks for responsive UI/UX	React.js (UI library), React Native (cross-platform mobile)
			Bootstrap 5 (CSS framework), Material-UI for design consistency
2	Containerization & Orchestration	Docker containers for consistent deployment across environments	Docker for containerization, Kubernetes (K8s) for orchestration, Helm for package management
			Docker Compose for local development
3	Security Implementations	Role-Based Access Control for multi-tier user access	JWT tokens, OAuth 2.0 for authentication, Spring Security or Passport.js
		Data encryption in transit and at rest	TLS 1.2+, SSL certificates, AES-256 encryption, bcrypt for password hashing
		API security with rate limiting, WAF	CORS protection, CSRF tokens, API key management, DDoS protection via WAF
		Secure secrets management	HashiCorp Vault or AWS Secrets Manager for storing sensitive credentials
		Code security scanning	SAST tools (SonarQube), dependency scanning (Snyk), DAST testing
4	Scalable Architecture	Horizontal scaling with load balancers	AWS ELB/ALB or NGINX load balancers distributing traffic across multiple instances
		Auto-scaling groups based on CPU/memory metrics	AWS Auto Scaling with target tracking policies
		Database replication and sharding	MySQL replication (Master-Slave), potential sharding for extreme scale
		Microservices architecture for independent scaling	Each service scales independently based on demand

S.No	Characteristics	Description	Technology
5	High Availability	Multi-zone deployment across geographic regions	AWS Multi-AZ for automatic failover
		Load balancing and failover	Automatic failover, health checks every 30 seconds
		Database replication with automatic failover	RDS Multi-AZ with synchronous replication, automated backups
		Redundant cache layer	Redis Cluster with sentinel for automatic failover
6	Performance Optimization	Database query optimization and indexing	Query analysis, index creation on frequently searched columns, query caching
		Asynchronous processing for long-running tasks	Task queues (Celery with Redis), background jobs for batch processing
		API response caching	HTTP caching headers, Redis caching for frequently accessed data
		CDN for static content delivery	CloudFront or Cloudflare for global content distribution
		Code minification and bundling	Webpack/Parcel for bundling, gzip compression for transfers
7	Disaster Recovery	Automated daily backups with point-in-time recovery	AWS Backup, incremental backups to minimize storage
		Cross-region replication for critical data	Multi-region S3 replication, database replication across regions
		Documented runbooks for recovery procedures	Disaster recovery playbooks, tested quarterly
		RTO <4 hours, RPO <1 hour targets	Monitored and validated through regular drills
8	Monitoring & Observability	Real-time system health monitoring	Prometheus for metrics collection, Grafana for visualization
		Centralized logging with search capabilities	ELK Stack or CloudWatch Logs with full-text search
		Application Performance Monitoring (APM)	New Relic or DataDog for transaction tracing
		Alert system for critical events	PagerDuty for alert routing, Slack integration for notifications
9	DevOps & Automation	Infrastructure as Code (IaC) for reproducible environments	Terraform for AWS resource provisioning, Ansible for configuration management
		Continuous Integration/Deployment (CI/CD)	Jenkins or GitLab CI with automated testing, staging deployment, production release

S.No	Characteristics	Description	Technology
		Automated testing at multiple levels	Unit tests, integration tests, end-to-end tests with >80% code coverage
10	<b>Data Privacy &amp; Compliance</b>	GDPR compliance for EU data subjects	Data processing agreements, right to be forgotten implementation
		Data encryption and access controls	Encrypted storage, role-based access, audit logs
		Regular security audits and penetration testing	Quarterly assessments, third-party security reviews
		Compliance documentation and certifications	SOC 2, ISO 27001 readiness (future certification target)
11	<b>Offline Capabilities</b>	Mobile app functions without internet (limited)	Local SQLite database on mobile device for offline data
		Automatic data sync when connectivity restored	Service workers for offline support, data reconciliation on sync
12	<b>Integration Capabilities</b>	Webhook support for third-party integrations	Webhook delivery system with retry logic and signature verification
		REST API for partner integrations	OpenAPI/Swagger documentation, API versioning strategy
		Event-driven architecture for loose coupling	Event bus pattern using message queues (RabbitMQ/SQS)

## TECHNOLOGY STACK SUMMARY BY LAYER

### Frontend Stack

Layer	Component	Technology
<b>Web UI</b>	Framework	React.js 17+, TypeScript
	State Management	Redux / Context API
	UI Components	Material-UI / Bootstrap 5
	Charts/Visualization	D3.js, Chart.js, Recharts
	HTTP Client	Axios
	Real-time Updates	<a href="#">Socket.io</a> client
<b>Mobile</b>	Framework	React Native (Expo)
	Platform	iOS 12+, Android 6.0+
	Native Modules	React Native Camera, Geolocation
	Local Storage	AsyncStorage, SQLite
	Navigation	React Navigation

## Backend Stack

Layer	Component	Technology
API Gateway	Service	Kong or AWS API Gateway
	Authentication	JWT, OAuth 2.0
	Rate Limiting	Kong plugins or Lambda functions
Application Server	Framework	Python-Flask or Node.js Express
	ORM	SQLAlchemy or Sequelize
	Validation	Marshmallow or Joi
	Task Queue	Celery (Python) or Bull (Node)
Microservices	Pattern	Microservices Architecture
	Communication	REST APIs, gRPC (optional)
	Containerization	Docker, Kubernetes
	Service Mesh	Istio (optional for advanced deployments)

## Data Stack

Layer	Component	Technology
Relational DB	DBMS	MySQL 8.0+ or PostgreSQL 13+
	Replication	Master-Slave replication
	Backups	Automated daily snapshots
Cache	In-Memory Store	Redis 6.0+ Cluster
	Pub/Sub	Redis Pub/Sub
	Session Store	Redis Sessions
Data Warehouse	Storage	AWS S3 Data Lake
	Analytics	AWS Athena or BigQuery
	ETL	Apache Spark or Airflow

## Infrastructure Stack

Layer	Component	Technology
Cloud Provider	Hosting	AWS or Google Cloud Platform
Compute	Virtual Machines	EC2 instances (t3.medium to t3.large)
	Container Orchestration	Kubernetes (EKS) or Docker Swarm
Database	Managed Service	AWS RDS (Multi-AZ)
Storage	Object Storage	AWS S3 with CloudFront CDN



Layer	Component	Technology
	File System	EBS volumes for persistent storage
<b>Networking</b>	Load Balancer	AWS ELB/ALB
	CDN	CloudFront or Cloudflare
	DNS	Route 53 with health checks

### Third-Party Integrations

Service	Provider	Purpose
<b>Mapping &amp; Navigation</b>	Google Maps API	Route optimization, directions, geocoding
<b>Communications</b>	Twilio	SMS, WhatsApp, voice notifications
<b>Push Notifications</b>	Firebase Cloud Messaging	Mobile app push notifications
<b>Email</b>	SendGrid or AWS SES	Transactional emails
<b>Payment Processing</b>	Razorpay or Stripe	Donation processing, CSR payments
<b>Analytics</b>	Google Analytics / Mixpanel	User behavior tracking
<b>Monitoring</b>	DataDog or New Relic	APM and infrastructure monitoring
<b>Logging</b>	ELK Stack or CloudWatch	Centralized logging

## DEPLOYMENT ARCHITECTURE

### Phase-Based Deployment

**Phase 1-2 (Pilot & Scaling):** Single Region Deployment

- Primary region: AWS ap-south-1 (Mumbai)
- Single database with manual backup
- Basic load balancing
- Cost-optimized instance types

**Phase 3-4 (Compliance & Technology):** Multi-Region Preparation

- Primary region: ap-south-1 (Mumbai)
- Secondary region: ap-southeast-1 (Singapore) for disaster recovery
- Database replication across regions
- Automated failover capability

**Phase 5 (Scaling & Multi-City):** Multi-Region, Multi-City

- Primary regions: ap-south-1 (Mumbai), ap-southeast-1 (Singapore)
- Regional deployments in each major city (Tier-1 cities)

- Global CDN for content delivery
- Cross-region failover with <1 minute RTO

DEVELOPMENT TOOLS & ENVIRONMENTS

Tool Category	Tools	Purpose
Version Control	Git (GitHub/GitLab)	Source code management, branching strategy
CI/CD	Jenkins, GitLab CI	Automated testing, building, deployment
Testing	Jest, Mocha, Pytest	Unit testing, integration testing
Code Quality	SonarQube, ESLint	Code analysis, quality gates
Documentation	Swagger/OpenAPI, MkDocs	API documentation, technical docs
Project Management	Jira, Confluence	Sprint planning, documentation
Communication	Slack, Teams	Team communication, CI/CD notifications
Debugging	Chrome DevTools, VS Code	Frontend and backend debugging

TECHNOLOGY DECISIONS RATIONALE

Decision	Rationale
React.js & React Native	Single codebase for web and mobile, large community, extensive ecosystem
Python-Flask or Node.js	Rapid development, lightweight, easy to learn, good for microservices
MySQL/PostgreSQL	ACID compliance, strong community, suitable for structured data
Redis	Extremely fast, perfect for caching and real-time data, pub/sub capabilities
AWS/GCP	Managed services reduce operations overhead, auto-scaling, global infrastructure
Docker & Kubernetes	Consistent deployment, scalability, industry standard
Microservices	Independent scaling, fault isolation, team autonomy

TECHNOLOGY ROADMAP

Phase	Quarter	Technology Focus
Phase 1-2	Q1-Q2 2026	Foundation: Backend APIs, Mobile app MVP, Basic dashboard
Phase 3	Q3 2026	Compliance: Advanced monitoring, audit systems, security hardening
Phase 4	Q4 2026-Q1 2027	Technology: ML for route optimization, advanced analytics, real-time features
Phase 5	Q2 2027+	Scaling: Multi-region deployment, global CDN, advanced DevOps