

Problem Statement — BBSR Smart Grocery Delivery Platform

Background

With Bhubaneswar rapidly developing as a smart city, there is a growing demand for **fast, reliable, and efficient grocery delivery services**.

Existing delivery solutions face several challenges:

- **Delayed deliveries** due to poor route optimization.
- **Stock mismatch** because of unsynchronized inventory updates.
- **Limited customer engagement** with no personalized recommendations.
- **Manual order processing** leading to human errors.
- **Low scalability** during peak festival seasons or emergencies.

These issues reduce customer satisfaction and profitability while increasing operational costs.

Need

To address these challenges, we propose a **Cloud-based Grocery Delivery Management System** for Bhubaneswar that will:

- **Enable real-time ordering** through mobile & web platforms.
 - **Integrate with local stores** for live inventory updates.
 - **Optimize delivery routes** using GPS & AI-based traffic prediction.
 - **Provide real-time order tracking** for customers.
 - **Implement automated billing and digital payments**.
 - **Scale instantly** during high-demand periods using cloud infrastructure.
-

Objectives

1. **Customer Convenience** – Mobile app for quick ordering, personalized suggestions, and live tracking.
2. **Efficient Inventory Management** – Live updates from multiple partner stores.
3. **Faster Delivery** – AI-based route optimization and GPS tracking.
4. **Operational Transparency** – Dashboards for store owners, delivery partners, and admins.

5. **Scalable Architecture** – Cloud-native platform that can expand across Odisha.
 6. **Continuous Feedback & Improvement** – Agile sprints with customer feedback loops.
-

Agile Delivery Plan

Epics & Features

1. **Customer Ordering System**
 - User registration/login via phone or social login
 - Product search, filter, and recommendation engine
 - Multi-store cart & checkout
 2. **Inventory & Store Management**
 - Store onboarding module
 - Real-time stock updates via APIs
 - Low-stock alerts
 3. **Delivery Management**
 - GPS-based delivery partner tracking
 - AI-based route optimization
 - Proof of delivery via digital signature/OTP
 4. **Payments & Billing**
 - Multiple payment gateways
 - Auto-generated invoices
 - Refund and cancellation system
 5. **Analytics & Reports**
 - Daily sales and delivery statistics
 - Customer buying pattern analysis
 - Delivery performance metrics
-

Sample Sprint Plan (2-week sprints)

Sprint Deliverables

- 1 Cloud infrastructure setup, authentication module
- 2 Product catalog & search API
- 3 Inventory sync with stores
- 4 Delivery partner tracking & route optimization
- 5 Payment gateway integration
- 6 Analytics dashboard
- 7 Gamification for customer loyalty
- 8 UAT & launch in BBSR

Latest DevOps Toolchain

Stage	Tools
Project Management	Jira, Trello, Confluence
Source Control	GitHub
CI/CD	GitHub Actions, ArgoCD
Testing	Selenium (UI), Postman (API), PyTest
Containerization	Docker
Orchestration	Kubernetes (EKS/GKE)
Monitoring	Prometheus, Grafana
Logging	ELK Stack
Security	SonarQube, OWASP ZAP
Infrastructure as Code	Terraform

Cloud Architecture (AWS Example)

- **Frontend:** React.js / Flutter → AWS Amplify

- **Backend:** Node.js / Django → AWS Lambda / EKS
 - **Database:** Amazon RDS (PostgreSQL) + DynamoDB (cache & lookups)
 - **Storage:** Amazon S3 (images, documents)
 - **AI/ML:** AWS SageMaker (route optimization & recommendations)
 - **Auth:** AWS Cognito
 - **Payments:** Razorpay / Paytm integration
 - **Notifications:** AWS SNS + Twilio
 - **CI/CD:** AWS CodePipeline + GitHub Actions
 - **Monitoring:** Prometheus + AWS CloudWatch
-

Workflow

1. **Plan** – Define user stories in Jira.
2. **Develop** – Code in GitHub with feature branches.
3. **Build & Test** – Automated pipelines for build, unit & integration tests.
4. **Containerize** – Package services in Docker images.
5. **Deploy** – Kubernetes via ArgoCD.
6. **Monitor & Improve** – Track KPIs, customer feedback, and improve via next sprint.