**Organic Produce Delivery App - Business Requirements Document**

**1. Project Overview**

**1.1 Project Description**

A mobile e-commerce application for ordering fresh vegetables, fruits, organic produce, and rice with scheduled delivery slots. The app connects customers with fresh produce suppliers through a user-friendly mobile interface.

**1.2 Technology Stack**

* **Frontend**: React Native (iOS & Android)
* **Backend**: Node.js with Express.js
* **Database**: MongoDB
* **Authentication**: JWT-based authentication
* **Payment Integration**: Stripe/Razorpay/PayPal

**1.3 Target Users**

* End customers seeking fresh organic produce
* Admin users managing inventory and orders
* Delivery personnel (future scope)

**2. Functional Requirements**

**2.1 User Authentication & Management**

* User registration with email/phone verification
* Login/logout functionality
* Password reset capability
* Profile management (address, preferences, payment methods)
* Guest checkout option

**2.2 Product Catalog Management**

* Browse products by categories:
  + Fresh Vegetables
  + Fresh Fruits
  + Organic Vegetables
  + Organic Fruits
  + Rice & Grains
* Product search and filtering
* Product details with images, descriptions, nutritional info
* Price display with units (per kg, per piece, per bundle)
* Stock availability status
* Seasonal product recommendations

**2.3 Shopping Cart & Checkout**

* Add/remove items from cart
* Quantity adjustment
* Cart persistence across sessions
* Price calculation with taxes
* Coupon/discount code application
* Multiple payment methods support

**2.4 Delivery Slot Management (Core Feature)**

* **Slot Configuration**:
  + Morning slots (6 AM - 12 PM)
  + Evening slots (2 PM - 8 PM)
  + Express delivery (same day, premium)
  + Scheduled delivery (up to 7 days advance)
* **Slot Selection**:
  + Calendar view for date selection
  + Available time slots display
  + Slot capacity management
  + Real-time slot availability
  + Slot pricing (free/premium charges)
* **Slot Restrictions**:
  + Minimum order value for delivery
  + Delivery area coverage
  + Weather-based slot suspension
  + Holiday schedule management

**2.5 Order Management**

* Order placement and confirmation
* Order tracking with status updates
* Order history and reordering
* Order modification (before processing)
* Order cancellation with refund policy
* Receipt and invoice generation

**2.6 Notification System**

* Push notifications for:
  + Order confirmations
  + Delivery updates
  + Promotional offers
  + Slot availability
  + Payment confirmations

**3. Admin Panel Requirements**

**3.1 Dashboard**

* Sales analytics and reports
* Order management overview
* Inventory status summary
* Delivery slot utilization
* Customer metrics

**3.2 Product Management**

* Add/edit/delete products
* Bulk product upload via CSV
* Product image management
* Category and subcategory management
* Pricing and discount management
* Stock level monitoring and alerts

**3.3 Inventory Management**

* Real-time stock tracking
* Low stock alerts
* Batch/expiry date tracking
* Supplier management
* Purchase order generation

**3.4 Order Management**

* View all orders with filters
* Order status updates
* Assignment to delivery slots
* Customer communication tools
* Refund and return processing

**3.5 Delivery Slot Management**

* Create and modify delivery slots
* Set slot capacities and pricing
* View slot utilization reports
* Manage delivery areas and zones
* Holiday and exception scheduling

**3.6 Customer Management**

* Customer profiles and order history
* Customer support ticket system
* Loyalty program management
* Customer segmentation and targeting

**3.7 Content Management**

* App banners and promotional content
* Blog/recipe content management
* FAQ and help content
* Terms and privacy policy updates

**4. Technical Requirements**

**4.1 Backend API Specifications**

**4.1.1 Authentication Endpoints**

POST /api/auth/register

POST /api/auth/login

POST /api/auth/logout

POST /api/auth/refresh-token

POST /api/auth/forgot-password

POST /api/auth/reset-password

**4.1.2 Product Endpoints**

GET /api/products

GET /api/products/:id

GET /api/products/category/:category

GET /api/products/search?q=query

POST /api/admin/products (Admin only)

PUT /api/admin/products/:id (Admin only)

DELETE /api/admin/products/:id (Admin only)

**4.1.3 Delivery Slot Endpoints**

GET /api/delivery-slots/available

GET /api/delivery-slots/date/:date

POST /api/delivery-slots/book

PUT /api/delivery-slots/:id/modify

DELETE /api/delivery-slots/:id/cancel

GET /api/admin/delivery-slots (Admin only)

POST /api/admin/delivery-slots (Admin only)

**4.1.4 Order Endpoints**

POST /api/orders

GET /api/orders/user/:userId

GET /api/orders/:orderId

PUT /api/orders/:orderId/status

GET /api/admin/orders (Admin only)

PUT /api/admin/orders/:orderId (Admin only)

**4.2 Database Schema (MongoDB)**

**4.2.1 Users Collection**

{

\_id: ObjectId,

name: String,

email: String,

phone: String,

password: String (hashed),

role: String, // 'customer', 'admin'

addresses: [{

type: String, // 'home', 'work'

street: String,

city: String,

state: String,

zipCode: String,

coordinates: { lat: Number, lng: Number }

}],

preferences: {

organic\_only: Boolean,

dietary\_restrictions: [String]

},

created\_at: Date,

updated\_at: Date

}

**4.2.2 Products Collection**

{

\_id: ObjectId,

name: String,

description: String,

category: String,

subcategory: String,

images: [String],

price: Number,

unit: String, // 'kg', 'piece', 'bundle'

stock\_quantity: Number,

organic: Boolean,

nutritional\_info: {

calories: Number,

vitamins: [String]

},

seasonal: Boolean,

available: Boolean,

created\_at: Date,

updated\_at: Date

}

**4.2.3 Delivery Slots Collection**

{

\_id: ObjectId,

date: Date,

time\_start: String, // "09:00"

time\_end: String, // "12:00"

capacity: Number,

booked: Number,

price: Number, // delivery charge

type: String, // 'standard', 'express'

areas: [String], // serviceable areas

active: Boolean,

created\_at: Date

}

**4.2.4 Orders Collection**

{

\_id: ObjectId,

user\_id: ObjectId,

order\_number: String,

items: [{

product\_id: ObjectId,

quantity: Number,

price: Number,

unit: String

}],

delivery\_slot\_id: ObjectId,

delivery\_address: Object,

subtotal: Number,

delivery\_charge: Number,

tax: Number,

discount: Number,

total: Number,

payment\_method: String,

payment\_status: String,

order\_status: String, // 'pending', 'confirmed', 'packed', 'out\_for\_delivery', 'delivered', 'cancelled'

notes: String,

created\_at: Date,

updated\_at: Date

}

**4.3 Performance Requirements**

* App load time: < 3 seconds
* API response time: < 500ms
* Support for 1000+ concurrent users
* 99.9% uptime availability
* Offline capability for browsing (cached data)

**4.4 Security Requirements**

* JWT token-based authentication
* Password encryption using bcrypt
* HTTPS for all API communications
* Input validation and sanitization
* Rate limiting for API endpoints
* PCI DSS compliance for payment processing

**5. User Experience Requirements**

**5.1 Mobile App Features**

* Intuitive navigation with bottom tab bar
* Quick reorder from previous orders
* Wishlist/favorites functionality
* Voice search capability
* Barcode scanning for products
* Social sharing of products
* Referral program integration

**5.2 Accessibility**

* Support for screen readers
* High contrast mode
* Font size adjustments
* Voice commands support
* Multi-language support (English, local languages)

**6. Integration Requirements**

**6.1 Payment Gateways**

* Credit/Debit card processing
* Digital wallets (Google Pay, Apple Pay)
* UPI payments (for Indian market)
* Cash on Delivery option
* Subscription payment handling

**6.2 Third-Party Services**

* SMS gateway for OTP verification
* Email service for notifications
* Maps integration for delivery tracking
* Analytics service (Firebase/Google Analytics)
* Cloud storage for images (AWS S3/Cloudinary)

**7. Deployment Requirements**

**7.1 Environment Setup**

* Development, Staging, and Production environments
* CI/CD pipeline setup
* Automated testing integration
* Database backup and recovery
* Load balancing configuration

**7.2 Monitoring and Logging**

* Application performance monitoring
* Error tracking and reporting
* User behavior analytics
* Server health monitoring
* Security audit logging

**8. Success Metrics**

**8.1 Business KPIs**

* Customer acquisition rate
* Order completion rate
* Average order value
* Customer retention rate
* Delivery slot utilization
* Customer satisfaction scores

**8.2 Technical KPIs**

* App crash rate < 1%
* API uptime > 99.9%
* Page load speed < 3 seconds
* Successful payment rate > 98%

**9. Future Enhancements**

**9.1 Phase 2 Features**

* Subscription-based recurring orders
* AI-powered product recommendations
* Augmented reality for product preview
* Community features (reviews, recipes)
* Delivery tracking with GPS
* Multi-vendor marketplace

**9.2 Advanced Analytics**

* Predictive inventory management
* Dynamic pricing based on demand
* Customer behavior analysis
* Supply chain optimization
* Seasonal demand forecasting

*This document serves as the foundation for developing the organic produce delivery application. All stakeholders should review and approve these requirements before development begins.*