Requirements Specification Document

QuickCart Functional Specification Document

ADIDEV M S - AM.AI.U4AID23001

C R ADHARSH - AM.AI.U4AID23006

RUGMINIDEVI S - AM.AI.U4AID23018

ADITHYA ASHOK - AM.AI.U4AID23025

1. Introduction

1.1 Overview

This document outlines the complete functional and non-functional specifications of **QuickCart**, a responsive full-stack e-commerce platform for customers and sellers. The backend is developed in **Python**, interfacing with a **PostgreSQL** database, while the frontend uses **React.js** with routing and state management. It aims to simulate a real-world e-commerce experience, with future scope for full authentication and payment integration.

1.1 Purpose

QuickCart is a modern, responsive e-commerce web application built with React, offering a seamless shopping experience for customers and a dedicated dashboard for sellers. It features product browsing, category filtering, cart management, order placement, and a simulated login system, all with a mobile-friendly design. The app uses React Context for global state management and Tailwind CSS for styling, but operates without a real backend or payment system—data is managed locally using dummy datasets. QuickCart showcases smooth navigation, toast notifications, and real-time cart updates, providing a solid frontend framework for an e-commerce platform.

2. Overall Description

2.1 Product Perspective

QuickCart is a web-based system consisting of:

- **Frontend**: React.js with React Router, Tailwind CSS
- **Backend**: Python (FastAPI or Flask), connected to PostgreSQL
- **Middleware**: Optional Node.js/Express interface (if used for routing or token middleware)
- **Note**: No real-time payment or email service yet (simulated)

3. Functional Requirements

3.1 User Functionalities

3.1.1 Home Page (/)

- Welcome message and featured categories/products
- Responsive banners
- Navigation links to products/categories

3.1.2 Product Browsing

- **All Products Page (`/products`)**
    - Grid view of all available products
    - Pagination (if implemented)
- **Category Browsing (`/products/:category`)**

    - Filter products by category
- **Product Detail Page (`/products/:category/:id`)**
    - Title, description, images, pricing, seller
    - Add to cart button
    - Quantity selector

3.1.3 Search Functionality

- Search bar in Navbar
- Typing auto-navigates to `/products` with filtered results
- Case-insensitive match by product name/category

3.1.4 Cart (`/cart`)

- Add products from product detail
- Increase/decrease quantity or remove items
- Price breakdown:
    - Subtotal
    - Tax (if simulated)
    - Total
- Button to continue to checkout

3.1.5 Checkout Flow

- **Select Address**
    - Option to add a new address
    - Dropdown to select saved address
- **Payment Method**
    - Simulated selection (COD, UPI, Card)
    - No actual payment gateway
- **Place Order**
    - Sends cart + address + payment to backend
    - Clears local cart after placing
    - Redirects to order confirmation

3.1.6 Order Confirmation (`/order-confirmation`)

- Message confirming order placement

- Summary of items ordered and total amount

3.1.7 Order History (`/my-orders`)

- List of past orders
- Each order shows:
  - Items
  - Address used
  - Payment method
  - Status (Pending, Delivered, etc.)
- Backend stores orders in `orders` + `order_items` tables

3.1.8 Address Management (`/add-address`)

- Form to add a new address:
  - Name, address line, city, state, pincode, phone number
- Backend stores in `addresses` table

3.1.9 Login Modal (Shown from anywhere)

- Overlay login modal controlled via context
- No real authentication (currently simulated)
- **Future**: convert to real JWT-based login

3.1.10 Contact Page (`/contact`)

- Form with name, email, subject, and message
- Stores in `contact_messages` table or sends email (future scope)

3.1.11 Toast Notifications

- On add to cart, order success, errors, login prompt
- Uses `react-hot-toast`

3.1.12 Responsive Design

- Mobile-friendly:
  - Hamburger navbar

    - Collapsible menus
- Desktop-optimized grid/product layouts

3.2 Seller Functionalities

3.2.1 Seller Login (`/seller`)

- Simulated login (real backend ready)
- Login decides whether to show `SellerLayout` or `SellerLogin`

3.2.2 Seller Dashboard (`/seller`)

- Accessible after login
- Sidebar/menu options for navigation
- Hides main navbar/footer

3.2.3 Add Product (`/seller`)

- Default nested route
- Form for:
    - Title, price, category, description, image URL, stock
- Sends data to backend → saved in `products` table

3.2.4 Product List (`/seller/product-list`)

- Table/grid view of seller's added products
- Edit and delete buttons

3.2.5 Orders Management (`/seller/orders`)

- List of customer orders for seller's products
- Shows:
    - Items
    - Address
    - Status
- Update status (e.g., shipped, delivered)

4. User Interface Specification

The QuickCart Online Shopping Platform is designed with a clean, responsive UI that ensures a seamless shopping experience across devices.

4.1 Home Page

- Hero banner with grocery-themed promotional imagery
- Delivery info banner
- Navigation bar with links to Home, Products, Cart, Login/Logout, Profile
- Product sections include:
    - Horizontally scrollable sliders by category (Vegetables, Fruits, Frozen Foods)
    - Each product card includes:
        - Product image
        - Name
        - Price
        - “Add to Cart” button
    - Slider controls for navigation

4.2 Login / Signup Modal

- Modal popup triggered from navbar
- Toggle between login and signup views
- Input fields: username, password, confirm password (for signup)
- Inline validation messages for empty or mismatched fields
- Login state saved using `localStorage`, affecting UI visibility (e.g., Profile, Logout)

4.3 Profile & Order History Modal

- Available only to logged-in users
- Displays:
    - Username
    - List of past orders with:
        - Items
        - Order time
        - Payment method
        - “Cancel Order” button for valid orders

        - “Clear History” button

4.4 Cart Page (`/cart`)

- Dedicated cart view displaying:
    - Items added with:
        - Image, Name, Price, Quantity
        - “+”, “−” to adjust quantity
        - “Remove” button
    - Auto-updating total price
    - “Proceed to Payment” button leading to payment interface

4.5 Navigation Bar

- Positioned at the top of every page
- Contains:
    - Left: QuickCart Grocery logo
    - Right: Links to Home, Products, Cart, Login/Logout, Profile
- Fully responsive; mobile version will use a hamburger menu (to be implemented)

4.6 Payment Page (`/payment`)

- Displays:
    - Ordered items, subtotal, and total
    - Payment method options:
        - UPI
        - Cash on Delivery
    - UPI ID field appears dynamically if UPI is selected
    - Validation before placing order
- “Place Order” button triggers:
    - Order storage
    - Redirection to confirmation screen

4.7 Order Confirmation Page (`/order-confirmation`)

- Confirmation of successful order
- Displays:
    - Placed time

    - Total amount
    - Payment method
    - Estimated delivery (2 hours from order time)
- “Continue Shopping” link navigates back to home

4.8 Orders Page (`/my-orders`)

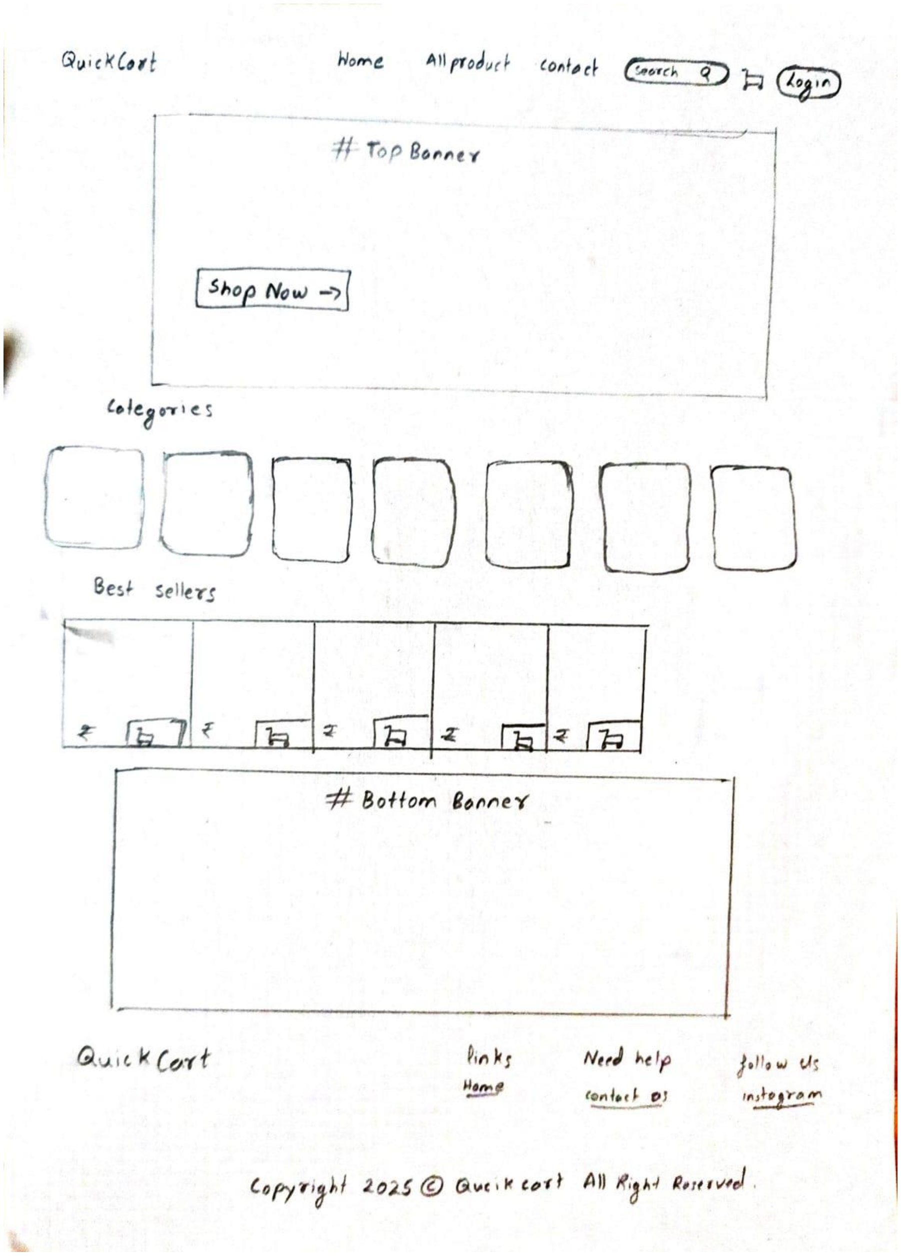
- Lists all user orders
- Each entry includes:
    - Items
    - Total price
    - Payment method
    - UPI ID (if used)
    - Order status (Confirmed/Cancelled)
    - “Cancel Order” button for eligible items

4.9 Footer

- Present on all main pages
- Contains:
    - Copyright
    - Basic footer note

4.10 Mobile Responsiveness

- Sliders support touch/swipe gestures
- Navbar collapses into hamburger menu on mobile (planned)
- All UI components are responsive and optimized for desktop, tablet, and mobile screens



5. Team Contributions

The development of QuickCart was a collaborative effort involving clearly defined roles and responsibilities among team members. Each member contributed to different facets of the system across frontend, backend, and documentation layers.

Team Members	Contribution

Adidev M S	Contact Page - Address Management Page - Payment Page with validation - Orders Page (Order status, cancellation, UPI display) - Requirements Specification Document preparation
C R Adharsh	Login/Signup Modal - User State Management (React Context) - Profile & Order History Modal - Checkout Flow: Address selection, payment selection, order placement - Order Confirmation Page
Rugminidevi S	Home Page, Navigation Bar, Cart Page (Frontend) - React Router setup - Search Functionality - Responsive Design (mobile & desktop) - Toast Notifications - Tailwind CSS styling - Mobile touch/swipe support
Adithya Ashok	Seller Login, Seller Dashboard, Add Product Page, Product List Page, Order Management Page (Seller Side) - Backend API development (FastAPI/Flask) - PostgreSQL database setup and integration - Cart persistence using localStorage