



Date	10 Agust 2025
Team ID	PNT20225TMID10192
Project Name	Project:- GROCERY WEBAPP
Maxumum marks	8 Marks

Sr No	Name	Colloge Name	Contact
1	Krushna Waingade	D.Y. Patil Agriculture & Techinal University	9579241280
2	Siban Hipparajge	D.Y. Patil Agriculture & Techinal University	9665456161
3	Sidharth Chava	D.Y. Patil Agriculture & Techinal University	96659 77617
4	Rushikesh Mokashi	D.Y. Patil Agriculture & Techinal University	9665456161

1.INTRODUCTION

1.1 Purpose

The purpose of this document is to specify the functional and non-functional requirements for the Grocery Web Application. This system is designed to allow customers to conveniently browse, select, and purchase grocery items online, while enabling the store administrators to manage products, categories, and orders efficiently.

It aims to:

- Provide customers with a user-friendly platform to shop for groceries from anywhere.
 - Facilitate order management, inventory tracking, and customer communication.
 - Support secure payment processing and order delivery scheduling.
-

1.2 Scope

The Grocery Web Application will provide the following core functionalities:

- **For Customers:**
 - User registration, login, and profile management.
 - Browse grocery categories and products with search and filter options.
 - Add items to cart, update quantities, and place orders.
 - Secure checkout with multiple payment methods.
 - Track order status and delivery details.
- **For Administrators:**
 - Add, update, and remove products and categories.
 - Manage inventory stock levels.
 - View and manage customer orders.
 - Generate sales reports and analytics.
- **Platform Scope:**
 - Accessible via modern web browsers on desktop and mobile devices.
 - Secure authentication and data protection measures.
 - Scalable to handle growing product lists and user base.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS** – Software Requirements Specification
- **UI** – User Interface
- **UX** – User Experience
- **API** – Application Programming Interface
- **DBMS** – Database Management System
- **HTTPS** – HyperText Transfer Protocol Secure

1.4 References

- IEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications
- Payment Gateway API Documentation (e.g., Razorpay/Stripe/PayPal)
- Web Security Standards (OWASP Top 10)

1.5 Overview

The Grocery Web Application will consist of a **frontend** for customer interaction, a **backend** for business logic and database management, and an **admin panel** for store management.

The system will follow a **three-tier architecture**:

1. **Presentation Layer:** User interface built with responsive web design.
2. **Application Layer:** Backend server handling requests, authentication, and business logic.
3. **Data Layer:** Centralized database storing user, product, and order data.

This application aims to **modernize grocery shopping** by offering convenience, accuracy in stock management, and improved customer satisfaction through a seamless digital experience.

2. Ideation Phase

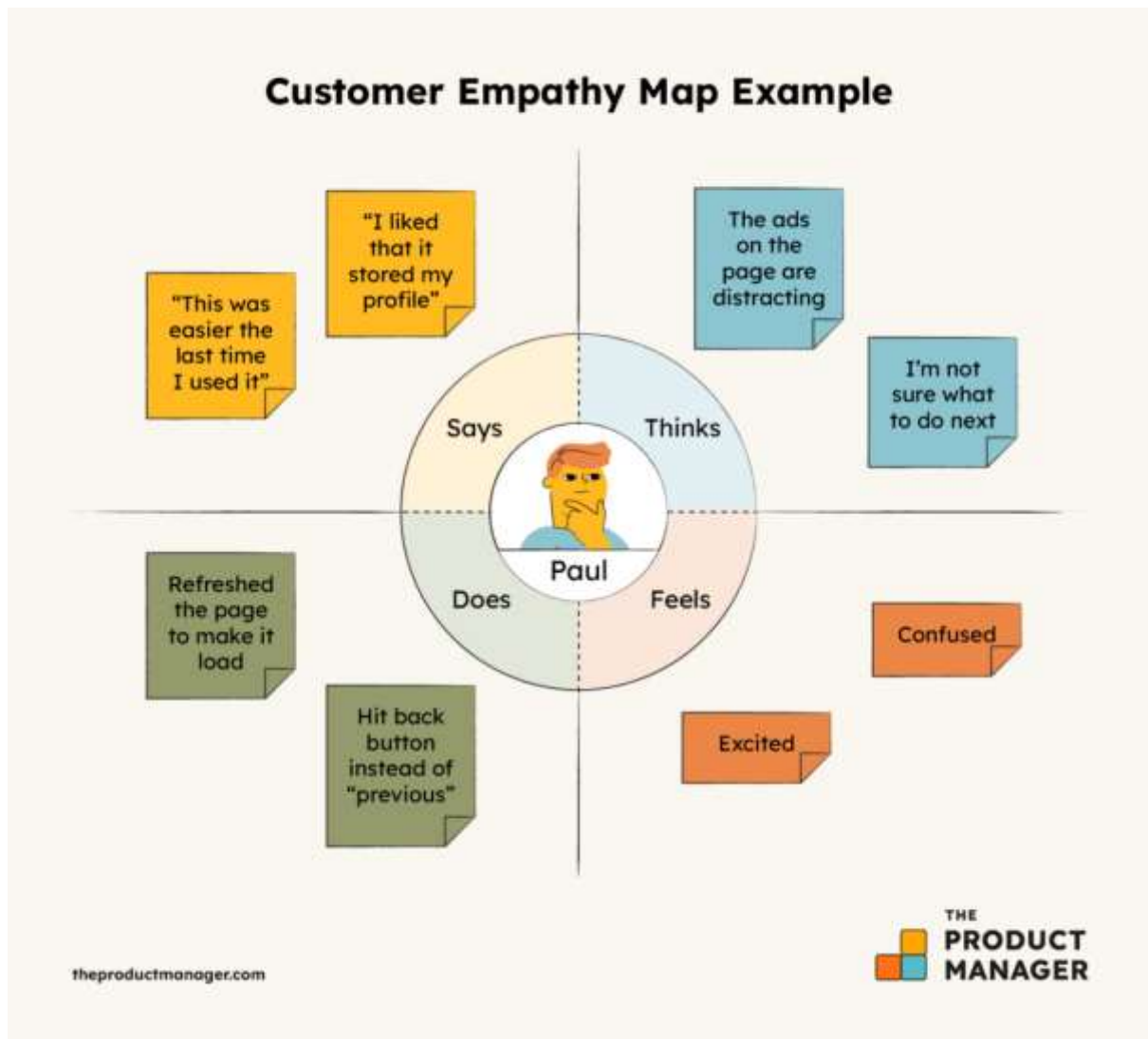
The ideation phase of the Grocery Web Application project focused on identifying the need for a convenient, accessible, and efficient online platform for purchasing groceries. The primary goal was to provide customers with an easy-to-use website where they could browse, select, and order grocery items from the comfort of their homes, while enabling store administrators to manage products, inventory, and orders seamlessly.

During brainstorming sessions, the following core objectives and features were identified:

- **User Convenience:** Allow customers to shop online without visiting a physical store.
- **Efficient Store Management:** Enable administrators to maintain product lists, update stock, and track orders in real time.
- **Responsive Design:** Ensure accessibility from desktop and mobile devices using HTML, CSS, and JavaScript for a smooth user experience.
- **Dynamic Functionality:** Implement PHP for backend logic to handle user authentication, cart management, and order processing.
- **Data Management:** Use an RDBMS (Relational Database Management System) to store and manage product information, customer data, and transaction history securely.
- **Scalable Structure:** Design a system that can be expanded to include more categories, advanced search filters, and integration with payment gateways in future phases.

The ideation stage concluded with the decision to adopt a **three-tier architecture**—frontend (HTML, CSS, JavaScript), backend (PHP), and database layer (RDBMS)—to ensure maintainability, scalability, and secure operations.

Step2 Empathy Map:



3. Requirement Analysis

3.1 Functional Requirements

The system must provide the following functionalities:

For Customers

1. User Registration & Login

- Create an account with basic details (name, email, password, phone).
- Secure authentication and password encryption.

2. Product Browsing

- View grocery items categorized (e.g., Fruits, Vegetables, Dairy, Beverages).
- Search products by name or keyword.
- Filter by category, price, or brand.

3. Shopping Cart

- Add, update, or remove products in the cart.
- Display total price and quantity.

4. Order Placement

- Checkout with payment options (Cash on Delivery / Online Payment Gateway).
- Select delivery address and preferred time.

5. Order Tracking

- View current order status (Processing, Shipped, Delivered).

6. Profile Management

- Update personal information, password, and delivery addresses.

For Administrators

1. Product Management

- Add, edit, or delete products.
- Upload product images and set prices.

2. Inventory Management

- Monitor stock levels and update quantities.

3. Order Management

- View, approve, and update customer orders.
- Change order status and assign delivery staff.

4. Reports & Analytics

- View sales reports, top-selling products, and customer activity.
-

3.2 Non-Functional Requirements

1. Performance

- The system should handle at least 100 concurrent users without performance degradation.

2. Usability

- The UI must be responsive and intuitive for both desktop and mobile users.

3. Security

- Implement HTTPS for secure data transfer.
- Use password hashing for user credentials.
- Validate all inputs to prevent SQL injection and XSS attacks.

4. Reliability

- The system should have 99% uptime during operational hours.

5. Scalability

- Should support adding more categories, products, and features without major code refactoring.

6. Maintainability

- Code should be modular for easy updates and debugging.
-

3.3 Software & Hardware Requirements

Software Requirements

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** PHP

- **Database:** MySQL or any RDBMS
- **Server:** Apache (XAMPP/WAMP)
- **Browser Support:** Chrome, Firefox, Edge, Safari

Hardware Requirements

- **Minimum Client System:** Dual-core processor, 4GB RAM, Internet connection
- **Minimum Server System:** Quad-core processor, 8GB RAM, 500GB storage

4. Project Design

4.1 Architectural Design

The Grocery Web Application follows a Three-Tier Architecture to ensure scalability, maintainability, and security:

1. Presentation Layer (Frontend)

- Built using HTML, CSS, and JavaScript.
- Responsible for displaying product listings, cart, order details, and admin dashboard.
- Responsive design for mobile and desktop compatibility.

2. Application Layer (Backend)

- Implemented using PHP.
- Handles user authentication, product management, order processing, and communication with the database.
- Contains business logic for calculating totals, managing inventory, and validating user inputs.

3. Data Layer (Database)

- **MySQL (RDBMS) used for storing:**
 - Product details
 - Customer information
 - Orders and transaction history
 - Inventory stock levels
- **Ensures data integrity and security with relational constraints.**

4.2 Module Design

The system is divided into the following modules:

Customer Module

- Registration & Login
- Product browsing & search
- Shopping cart management
- Checkout & payment
- Order tracking

Admin Module

- Product management (Add/Edit/Delete)
- Category management
- Inventory management
- Order management (Process/Update)
- Report generation

Database Module

- Tables for users, products, categories, orders, order_items, inventory
- Relationships maintained using primary and foreign keys

4.3 Interface Design

Customer Interface

- Home Page: Featured products, search bar, category list.
- Product Page: Product image, name, price, “Add to Cart” option.
- Cart Page: Item list, total amount, checkout button.
- Order Tracking Page: Displays order status and estimated delivery date.

Admin Interface

- Dashboard: Overview of orders, products, and stock status.
- Product Management Page: Form to add/edit/delete products.

- Order Management Page: Table to update order statuses.
 - Reports Page: Graphs and tables showing sales analytics.
-

4.4 Data Flow Diagram (DFD)

Level 0 (Context Diagram)

- User → Grocery Web App → Database → Admin

Level 1

- Customers can register, log in, browse products, add to cart, place orders, and track them.
- Admins can manage products, categories, orders, and inventory.

5. Functional Techniques

These describe what the system **should do** and the methods to ensure those functions are reliable and usable.

5.1 Functional Requirements Coverage

- **User Management:**
 - Registration, login/logout (with session handling).
 - Profile update and password change features.
- **Product Management:**
 - Add, edit, delete, and categorize products (for admin).
 - Product search, filtering, and sorting (for customers).
- **Cart & Checkout:**
 - Add/remove products to/from cart.
 - Update quantity in real time.
 - Checkout process with payment integration.
- **Order Management:**
 - Order placement, tracking, and cancellation.
 - Order history viewing for users.
- **Inventory Management:**

- Stock update in real time after purchase.
- **Notification System:**
 - Order status updates via email/SMS.
- **Security Measures:**
 - Password hashing, SQL injection prevention, CSRF protection.

5.2 Functional Techniques Implementation

- **Modular Code Structure:** Separate files for UI (HTML/CSS/JS), business logic (PHP), and database (RDBMS).
 - **Form Validation:**
 - Client-side validation (JavaScript) for faster feedback.
 - Server-side validation (PHP) for security.
 - **Role-based Access Control (RBAC):**
 - Separate permissions for admin, seller, and customer.
 - **REST API Integration:**
 - For mobile app or third-party service compatibility.
 - **Responsive Design:**
 - Using CSS media queries and frameworks like Bootstrap.
-

5. Performance Techniques

These ensure the system **runs fast, uses resources efficiently, and scales well.**

5.1 Backend Performance Techniques

- **Database Optimization:**
 - Use **indexes** on frequently queried columns (e.g., product name, category).
 - Avoid redundant queries by using **JOINS** effectively.
 - Use **prepared statements** in PHP to improve query execution time and security.
- **Caching:**

- Store frequently accessed data (like product lists) in memory (e.g., Redis or PHP file-based cache).
- **Pagination:**
 - Load product lists in pages instead of all at once.
- **Optimized Queries:**
 - Limit SELECT fields instead of using SELECT *.
 - Batch updates for multiple records at once.

5.2 Frontend Performance Techniques

- **Minification & Compression:**
 - Minify CSS, JavaScript, and HTML.
 - Use Gzip compression on the server.
- **Image Optimization:**
 - Use WebP or compressed JPEG/PNG formats.
 - Lazy loading for product images.
- **Asynchronous Loading:**
 - Load non-critical scripts asynchronously.
 - Use AJAX for cart updates without reloading the page.

5.3 Scalability & Load Handling

- **Load Balancing:** Distribute requests across multiple servers if traffic increases.
- **CDN Usage:** Deliver images, CSS, and JS files from a content delivery network.
- **Session Management:** Use server-side storage or distributed session handling for multiple servers.

Advantages of Grocery Web App

1. **Convenience & Time Saving**
 - Customers can shop anytime without visiting a physical store.
 - Quick search, filter, and sorting make product selection faster.
2. **Wide Product Range**
 - Ability to showcase more products than a physical store.
3. **Real-Time Updates**

- Inventory, pricing, and discounts updated instantly.
 - 4. **Cost Efficiency for Business**
 - Reduced need for large physical outlets and staff.
 - 5. **Personalized Shopping Experience**
 - Recommendations based on previous purchases.
 - 6. **Better Order Management**
 - Automated billing, order tracking, and inventory handling.
 - 7. **Scalability**
 - Easy to expand product categories, locations, and users.
 - 8. **Payment Flexibility**
 - Multiple payment methods: cards, UPI, wallets, COD.
-

Disadvantages of Grocery Web App

1. **Dependency on Internet & Devices**
 - Customers without internet or smartphones can't access it.
2. **Delivery Time**
 - Unlike physical shopping, customers need to wait for delivery.
3. **Lack of Physical Inspection**
 - Customers cannot touch or check the freshness of products before purchase.
4. **Technical Issues**
 - Server downtime, bugs, or slow loading can affect sales.
5. **High Competition**
 - Competing with established e-commerce giants can be challenging.
6. **Logistics Challenges**
 - Maintaining timely deliveries and managing perishable goods can be difficult.
7. **Security Risks**
 - Data breaches or payment fraud if security measures are weak.

6. Conclusion

The Grocery Web App serves as a modern, efficient, and customer-centric solution for purchasing groceries online. By integrating essential features such as user-friendly navigation, real-time inventory management, secure payment processing, and a responsive interface, the system addresses both the convenience needs of customers and the operational requirements of the business.

This platform eliminates the need for customers to visit physical stores, allowing them to browse, select, and purchase products from the comfort of their homes at any time. The application's design focuses on delivering a smooth shopping experience through advanced search and filter options, personalized recommendations, and seamless checkout. On the business side, it enables better stock control, sales tracking, and order management, ultimately improving efficiency and reducing operational costs.

From a technological perspective, the system is built using HTML, CSS, JavaScript, PHP, and an RDBMS for reliable and scalable performance. Security measures such as encrypted transactions, authentication protocols, and secure data handling ensure customer trust and data protection. Performance optimization techniques like caching, database indexing, and image compression enhance the overall speed and responsiveness of the application.

While there are challenges such as delivery logistics, internet dependency, and competition from established e-commerce platforms, the benefits far outweigh these limitations. With the growing demand for online grocery shopping, this web app has significant potential to expand its features, integrate mobile applications, and scale to serve a larger customer base.

In conclusion, the Grocery Web App is not only a technological product but also a step towards redefining the way people purchase daily essentials. It bridges the gap between digital convenience and real-world needs, making grocery shopping faster, easier, and more accessible for everyone.

7. Future Scope

The Grocery Web App has the potential to evolve significantly in terms of features, technology, and market reach. As consumer behavior and technology trends continue to change, the following enhancements can be considered in the future:

1. Mobile Application Development

- Launching Android and iOS mobile apps for greater accessibility and improved user experience.

2. AI-Based Product Recommendations

- Using artificial intelligence and machine learning to analyze purchase history and suggest personalized products.

3. Voice Search Integration

- Implementing voice-enabled shopping using smart assistants like Alexa, Google Assistant, or Siri.

4. Augmented Reality (AR) Shopping Experience

- Allowing customers to virtually view and inspect products before purchase.

5. Subscription & Auto-Replenishment Services

- Enabling customers to subscribe to recurring grocery items and receive automatic deliveries.

6. Advanced Delivery Tracking

- Real-time GPS-based order tracking for better transparency and customer satisfaction.

7. Multi-Language & Multi-Currency Support

- Expanding accessibility for international customers.

8. Integration with Smart Kitchen Appliances

- Syncing with IoT-enabled devices to automatically order items when stock is low.

9. Blockchain for Supply Chain Transparency

- Using blockchain to track product origins, freshness, and authenticity.

10. Loyalty Programs & Gamification

- Reward points, digital coupons, and gamified shopping challenges to boost customer retention.