**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 19 June 2025 |
| Team ID | LTVIP2025TMID53033 |
| Project Name | ShopSmart: Your Digital Grocery Store Experience |

**Solution Architecture:**

The **solution architecture** of ShopSmart is structured to address the business need for a seamless, scalable, and efficient online grocery shopping experience. The architecture effectively connects user-facing interfaces with backend services and data storage systems to ensure responsive and secure operations for customers, sellers, and administrators.

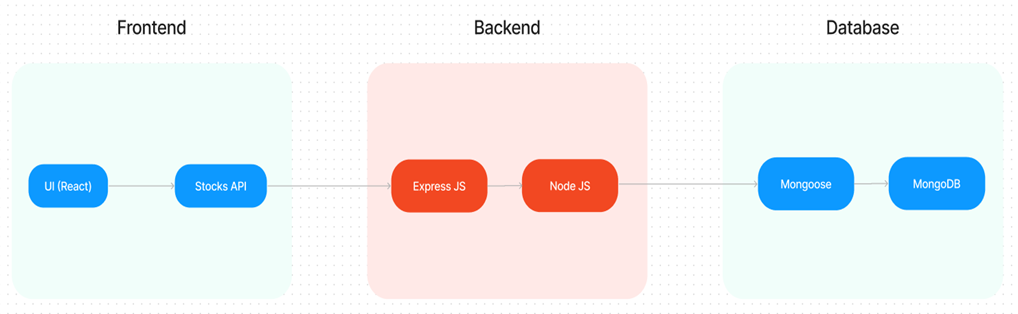
**Key Goals of the Architecture:**

* Identify the best-fit technology stack (MERN) to address the core challenges of traditional grocery shopping.
* Design a robust system capable of handling dynamic user roles and real-time operations.
* Define component interaction, data flow, and modular responsibilities across the system.
* Ensure maintainability, scalability, and performance optimization in a live e-commerce environment.

**Architecture Layers and Components:**

1. **Frontend Layer (React.js):**
   * Provides dynamic and interactive UI for different user roles (Customer, Seller, Admin).
   * Implements responsive design using modern libraries to support both mobile and desktop.
   * Integrates features like product listings, search filters, shopping cart, login/register, and order history.
2. **Backend Layer (Node.js + Express.js):**
   * Acts as the central API layer handling all HTTP requests from the frontend.
   * Uses Express routes and controllers to manage business logic like authentication, cart updates, product CRUD operations, and order processing.
   * Implements JWT-based authentication and middleware for secure and role-specific access control.
3. **Database Layer (MongoDB):**
   * Stores structured collections for Users, Products, Orders, Admins, and Carts.
   * Supports flexible document-based storage, enabling quick updates and retrievals for dynamic shopping behavior.
   * Integrated using Mongoose ODM to simplify queries and enforce data validation rules.
4. **Authentication & Authorization:**
   * JWT tokens are issued on login and stored in frontend localStorage.
   * Tokens are validated on every API call to ensure secure access.
   * Access is controlled through roles: Customers access user features, Sellers manage inventory, and Admins oversee the entire system.
5. **Real-Time Operations:**
   * Cart state, product availability, and admin dashboards reflect live updates.
   * Orders are tracked post-placement, with confirmation sent via email/SMS.
6. **Deployment and Scalability:**
   * Designed for local development (localhost:3000/frontend, 5000/backend) and scalable cloud deployment.
   * Modular folder structure and reusable components allow for easy feature expansion and scaling to additional regions or product categories.
7. **Error Handling and Security:**
   * API endpoints are protected with authentication middleware.
   * Server errors are caught and handled gracefully, with appropriate status codes and user feedback.
   * Passwords are hashed using bcrypt for secure storage.

**Technical Architecture:**

****

**Solution Architecture:**

