Solution Architecture

Date:	1 July 2025
Team ID:	LTVIP2025TMID54172
Project Name:	ShopSmart: Your Digital Grocery Store Experience
Maximum Marks:	4 Marks

Solution architecture is a structured approach that bridges the gap between business needs and technology implementation. In the context of our project, ShopSmart, the solution architecture provides a clear, technical roadmap for delivering a robust, secure, and scalable grocery web application.

Purpose and Goals

- Identify the best technical framework to solve the problems faced by customers and sellers in the grocery domain.
- Describe system structure, behaviors, and interactions to ensure stakeholder clarity.
- Define solution features, development stages, and infrastructure needs.
- Offer a blueprint to guide developers, testers, and project managers throughout the lifecycle.

Key Features of the Architecture

- **Modular Design**: Separation of concerns between customer UI, seller dashboard, and admin panel.
- **Cloud-Based Backend**: Hosting on scalable cloud infrastructure (e.g., AWS/GCP/Azure) for high availability.
- **Data Security**: Use of secure protocols for payment and data handling.
- **API-Driven**: RESTful APIs to manage communication between frontend and backend.
- **Authentication System**: Secure login for customers, sellers, and admins with role-based access control.
- **Database Layer**: Relational/NoSQL database for storing user data, products, orders, etc.

Development Phases

1. Requirement Gathering & UI/UX Design
2. Backend & API Development
3. Frontend Development (Web Interface)
4. Database Integration
5. Security & Testing
6. Deployment and Monitoring

Tools and Technologies

- **Frontend**: HTML, CSS, JavaScript (React.js or Angular)
- **Backend**: Node.js / Django / Spring Boot
- **Database**: MongoDB / MySQL / PostgreSQL
- **Authentication**: JWT / OAuth 2.0
- **Hosting**: AWS EC2 / Firebase / Heroku
- **Version Control**: Git & GitHub

Solution Architecture Diagram

The architecture diagram illustrates the interactions between users (customers, sellers, admins), frontend interfaces, backend services, and the database. It shows the flow from user actions (e.g., placing an order) through the API to the database and back with real-time updates.

[Insert Architecture Diagram Here – if required, I can create one for you.]