Project Design Phase
Solution Architecture

Date	15 February 2025
Team ID	LTVIP2026TMIDS79179
Project Name	ShopSmart – A Full-Stack Digital Grocery Store Web Application
Maximum Marks	4 Marks

Solution Architecture:

◆ 1. Introduction to Solution Architecture

The solution architecture of **ShopSmart** bridges the gap between traditional grocery shopping problems and a modern digital solution.

It defines:

- Overall system structure
- Technology components
- Data flow between layers
- Deployment architecture
- Security and scalability mechanisms

The system follows a **Three-Tier Architecture** to ensure modularity, scalability, and maintainability.

◆ 2. High-Level Architecture Overview

1 Presentation Layer (Frontend)

- Developed using **Angular 17**
- Single Page Application (SPA)
- Responsive UI using Bootstrap
- Role-based routing (Customer/Admin)
- HTTP Interceptor for JWT token handling
- Route Guards for secure navigation

Responsibility:

- User interaction
- Display products
- Cart management UI

- Order placement interface
- Admin dashboard interface

⚙ 2️⃣ Application Layer (Backend)

- Developed using **Node.js & Express.js**
- RESTful API architecture
- MVC design pattern
- Middleware for authentication & authorization
- File upload using Multer
- Error handling & validation

Core Modules:

- Authentication Service
- Product Service
- Cart Service
- Order Service
- Category Service
- Feedback Service

Responsibility:

- Business logic execution
- Data validation
- Token generation
- Secure API communication

🗄 3️⃣ Data Layer (Database)

- MongoDB Atlas (Cloud NoSQL Database)
- Mongoose ODM for schema management
- Indexed collections for fast querying
- Secure cloud-based data storage

Collections:

- Users
- Products
- Categories
- Carts
- Orders
- Feedbacks

◆ 3. Data Flow Architecture

Step 1: User Authentication

User → Angular UI → Login API →
Backend validates credentials →
Password verified using bcrypt →
JWT token generated →
Token returned to frontend →
Token stored in browser

Step 2: Product Browsing

User → Frontend Request → Product API →
Fetch data from MongoDB →
Return JSON response →
Display in UI

Step 3: Cart & Order Processing

User adds items → Cart API updates DB →
Checkout → Order API validates stock →
Save order → Reduce stock →
Return confirmation response

Step 4: Admin Operations

Admin → Admin Dashboard →
Product/Category/Order APIs →
Update MongoDB →
Changes reflected in frontend

◆ 4. Infrastructure Architecture

🖥 Development Environment

- Local Machine
- Node.js runtime

- npm package manager
- VS Code IDE

☁ Cloud Environment

- MongoDB Atlas (Cloud Database)
- Deployment-ready for AWS / Render / Railway

◆ 5. Security Architecture

- Password hashing using bcrypt
- JWT-based authentication
- Role-based access control (Admin/Customer)
- Protected REST APIs
- CORS configuration
- Input validation & sanitization

◆ 6. Scalability & Performance Design

- Three-tier modular architecture
- Cloud-hosted database for horizontal scaling
- Optimized API response (<500ms)
- Indexed database queries
- Efficient state management using RxJS
- RESTful API design for mobile app integration

◆ 7. Architecture Characteristics

Feature	Description
Architecture Type	Three-Tier Web Architecture
Communication	RESTful APIs (HTTP/JSON)
Security	JWT Authentication + bcrypt Encryption
Database	Cloud-based MongoDB Atlas
Deployment	Local / Cloud Ready

Feature	Description
Scalability	Cloud & Modular Design
Maintainability	MVC + Component-Based Design

◆ 8. Conclusion

The ShopSmart Solution Architecture effectively integrates frontend, backend, and cloud database components into a secure and scalable system.

It ensures:

- Efficient data flow
- Secure authentication
- Scalable cloud storage
- Modular development
- High performance

This architecture supports current requirements and allows future enhancements like mobile app integration, AI-based recommendations, and real payment gateway integration.