# SmartShop Full Stack Project Documentation

### Team ID: LTVIP2025TMID49801

### App Name: Glosary

### Team Member: Kagitha Jayavamsee

## Ὅ8 Table of Contents

1. Project Overview
2. Planning Phase
3. Requirement Analysis
4. Design Phase
5. Technology Stack
6. Implementation Details
7. Testing Phase
8. Deployment
9. Conclusion
10. Future Enhancements

## 🔹 Project Overview

**Glosary** is a full-stack SmartShop application designed to provide users with an intuitive online shopping experience. The platform integrates modern front-end interfaces, secure backend operations, and efficient database management to simulate a complete e-commerce solution.

**Objective:**  
To build a scalable, user-friendly online shopping application for managing products, cart operations, orders, payments, and user accounts.

## 🔹 Planning Phase

**Duration:** 10 Days  
**Goals:** - Define scope of the application - Set timeline and deliverables - Assign roles and responsibilities - Design UI wireframes and system architecture

**Tools Used:** - Trello for task management - Figma for wireframes - Draw.io for architecture diagrams

## 🔹 Requirement Analysis

**Functional Requirements:** - User authentication (Signup/Login) - Product listing and search - Add to cart and checkout - Order history - Admin panel for product management

**Non-Functional Requirements:** - Scalability - Responsiveness - Data security - Performance optimization

## 🔹 Design Phase

**Frontend Design:** - Responsive layout - Modern UI with accessible navigation

**Backend Design:** - RESTful API structure - MVC architecture

**Database Design:** - ER Diagram with tables: Users, Products, Orders, CartItems

**Wireframe Links:**  
(Add your Figma or design tool link here if available)

## 🔹 Technology Stack

| Layer | Technology |
| --- | --- |
| Frontend | HTML, CSS, JavaScript, React.js |
| Backend | Node.js, Express.js |
| Database | MongoDB (Mongoose ODM) |
| Authentication | JWT |
| Hosting | Render / Vercel / Netlify |
| Version Control | Git & GitHub |

## 🔹 Implementation Details

**Core Modules Implemented:** - User Module (Signup/Login/Logout) - Product Module (CRUD) - Cart Module (Add/Remove items) - Order Module (Place and view orders) - Admin Module (Dashboard, Product management)

**APIs Created:** - POST /register - POST /login - GET /products - POST /cart - POST /order - GET /orders

## 🔹 Testing Phase

**Testing Types:** - Unit Testing (Jest for Node.js) - Integration Testing (Postman, Swagger) - Manual UI Testing (Cross-device and browser) - Bug tracking with GitHub Issues

**Test Cases Examples:**

| Module | Test Case | Expected Result |
| --- | --- | --- |
| Login | Invalid credentials | Show error message |
| Cart | Add product to cart | Cart updates |
| Order | Checkout without login | Redirect to login |
| Admin | Delete product | Product removed |

## 🔹 Deployment

**Frontend:** Vercel / Netlify  
**Backend:** Render  
**Database:** MongoDB Atlas  
**Version Control:** GitHub  
**CI/CD:** GitHub Actions (Optional if used)

## 🔹 Conclusion

The **Glosary SmartShop** application successfully demonstrates a full-stack e-commerce solution with robust functionalities, user authentication, and administrative controls. It is scalable and adaptable for real-world deployment.

## 🔹 Future Enhancements

* Integrate Payment Gateway (e.g., Razorpay, Stripe)
* Add Wishlist Functionality
* Implement Real-time Notifications
* Admin Analytics Dashboard
* Multilingual support

### 