

“SRS Document on SociaMart”

Prepared by:

Antor Hawlader

(ID: 222071024)

Tamim Chad Likhon

(ID: 222071045)

Dept: CSE

8th Semester

Submit On: 15-04-2025

Submitted to:
Ekhfa Hossain,
Lecturer,
Department of
CSE & CSIT
SMUCT

SRS Document On: SociaMart (A Social Marketplace)

1. Introduction

1.1 Purpose

This document outlines the software requirements for SociaMart, a web-based platform that integrates e-commerce functionalities with social media features. The system aims to provide a unified solution for sellers and buyers, eliminating the need for third-party platforms like Facebook or standalone e-commerce websites.

1.2 Document Conventions

- **Shall** indicates a mandatory requirement.
- **Should** denotes a recommended requirement.
- **May** signifies an optional feature.

1.3 Purpose


This document is intended for:

- **Developers:** To understand the system's functionalities and constraints.
- **Designers:** To align UI/UX designs with system requirements.
- **Stakeholders:** To comprehend the project's scope and objectives.
- **Testers:** To develop test cases based on specified requirements.

Readers are advised to familiarize themselves with the overall description before delving into specific requirements.

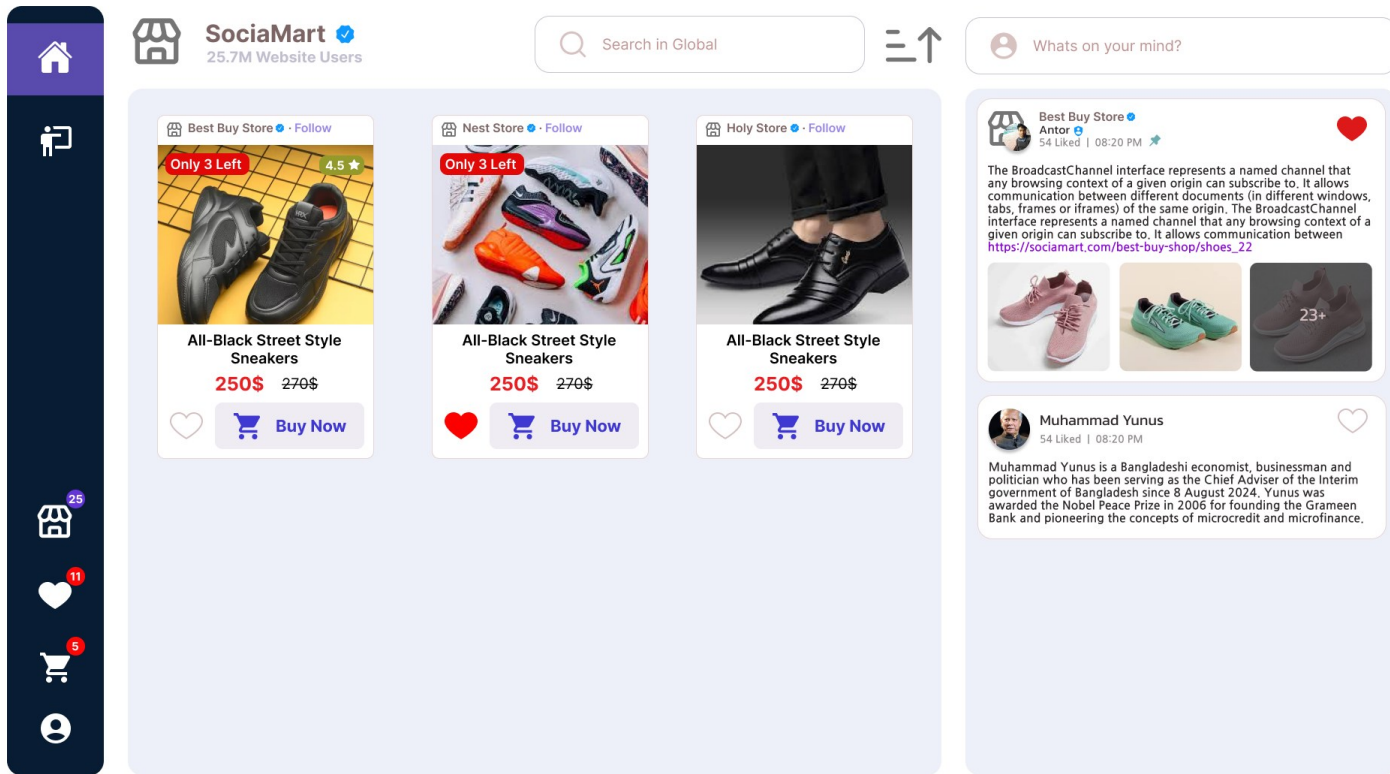
1.4 Project Scope

SociaMart is designed to:

- Enable users to register as buyers or sellers.
 - Allow sellers to create and manage their shops.
 - Facilitate product listings, purchases, and order management.
 - Incorporate social features like messaging and reviews.
 - Provide a responsive interface accessible via web browsers.
- 

1.5 References

Figma Design:



2. Overall Description

2.1 Product Perspective

SociaMart is a standalone web application developed using React.js for the frontend, Node.js with Express for the backend, and MongoDB for data storage. Firebase Authentication is utilized for user management, providing secure and scalable authentication services.

2.2 Product Features

- **User Registration and Authentication:** Secure sign-up and login using Firebase Authentication, supporting email/password and social logins.
- **Shop Creation:** Sellers can create personalized shops with unique URLs.
- **Product Management:** Sellers can add, edit, and delete product listings.
- **Shopping Cart and Checkout:** Buyers can add products to a cart and proceed to checkout.

- **Order Management:** Sellers can manage orders, and buyers can track their purchases.
- **Messaging System:** Real-time communication between buyers and sellers.
- **Reviews and Ratings:** Buyers can leave feedback on products and sellers.
- **Responsive Design:** Optimized for desktops, tablets, and mobile devices.

2.3 User Classes and Characteristics

Buyers: Individuals looking to purchase products. They can browse shops, add products to the cart, and communicate with sellers.

Sellers: Users who manage shops and product listings. They handle orders and interact with buyers.

Administrators: Oversee platform operations, manage user accounts, and ensure compliance with policies.

2.4 Operating Environment

Client Side: Modern web browsers (Chrome, Firefox, Safari, Edge).

Server Side: Node.js runtime environment.


Database: MongoDB Atlas.

Authentication: Firebase Authentication.

Hosting: Cloud-based hosting services.

2.5 Design and Implementation Constraints

User Registration and Authentication: Secure sign-up and login using Firebase Authentication, supporting email/password and social logins.

- The application must comply with data protection regulations.
 - Firebase Authentication will be the sole method for user authentication.
 - The system should be scalable to accommodate growing user bases.
 - All components must be responsive and accessible.
- 

2.6 Assumptions and Dependencies

- Users have access to stable internet connections.
- Firebase services are operational and accessible.
- Third-party services (e.g., payment gateways) are integrated successfully.

3. System Features

3.1 Functional Requirements

3.1.1 User Authentication

- The system shall allow users to register and log in using Firebase Authentication.
- Users shall be able to reset their passwords via email.

3.1.2 Shop Management

- Sellers shall be able to create, edit, and delete their shops.
- Each shop shall have a unique URL and customizable branding.

3.1.3 Product Listings

- Sellers shall be able to add, edit, and remove products.
- Product details shall include images, descriptions, prices, and stock levels.

3.1.4 Shopping Cart and Checkout

- Buyers shall be able to add products to a shopping cart.
- The system shall facilitate a secure checkout process.

3.1.5 Order Management

- Sellers shall be able to view and manage incoming orders.
- Buyers shall be able to track the status of their orders.

3.1.6 Messaging System

- The system shall provide real-time messaging between buyers and sellers.
- 

3.1.7 Reviews and Ratings

- Buyers shall be able to leave reviews and ratings for products and sellers.

4. External Interface Requirements

4.1 User Interfaces

The user interface shall be intuitive and responsive, adhering to the designs specified in the Figma prototype. Key interfaces include:

Home Page: Showcasing featured products and shops.

Shop Pages: Displaying seller-specific products.

Product Pages: Detailed views of individual products.

User Dashboards: Personalized areas for buyers and sellers.

4.2 Hardware Interfaces

No specific hardware interfaces are required. The application is accessible via standard computing devices.

4.3 Software Interfaces

Firestore Authentication: For user registration and login.

MongoDB: For data storage and retrieval.

Payment Gateway APIs: For processing transactions.


4.4 Communications Interfaces

HTTPS: Secure communication between client and server.

WebSockets: For real-time messaging features.

5. Nonfunctional Requirements

5.1 Performance Requirements

- The system shall handle up to 1,000 concurrent users without performance degradation.
 - Page load times shall not exceed 3 seconds under normal conditions.
- 

5.2 Safety Requirements

- The system shall implement data validation to prevent malicious inputs.
- Regular backups shall be performed to prevent data loss.

5.3 Security Requirements

- All user data shall be transmitted over secure channels (HTTPS).
- Firebase Authentication shall manage user credentials securely.
- Access controls shall be enforced to prevent unauthorized actions.

5.4 Software Quality Attributes

Usability: The application shall be user-friendly and intuitive.

Reliability: The system shall be available 99.9% of the time.

Maintainability: The codebase shall be modular to facilitate updates.

Scalability: The system shall accommodate increasing numbers of

