Product Review System

❖ Student Information

- **Name**: Aditya Hans
- **Registration Number**: 12308188
- **Roll Number**: 16
- **GitHub**: https://github.com/AddyTheDeveloper/Product_Review_System

1. Abstract

The **Product Review System** is a comprehensive, enterprise-grade full-stack web application developed using the **MERN stack (MongoDB, Express.js, React.js, and Node.js)**. The primary objective of this system is to deliver a reliable, transparent, and scalable platform that enables users to evaluate products through authentic, community-driven reviews.

The application supports secure authentication, role-based authorization, dynamic review aggregation, and administrative analytics. Designed with modern UI/UX principles and robust backend architecture, the system ensures high performance, data integrity, and extensibility. This project reflects real-world industry practices and demonstrates the practical application of full-stack development concepts.

2. Introduction

In the contemporary digital ecosystem, consumers heavily rely on online reviews before making purchasing decisions. However, the credibility and organization of such reviews often remain questionable. The **Product Review System** addresses these challenges by offering a structured and secure review environment that prioritizes authenticity, transparency, and usability.

The platform enables registered users to explore products, submit ratings, and share detailed feedback, while administrators maintain platform quality through controlled moderation. The system integrates secure authentication mechanisms, automated rating computation, and responsive design, ensuring a seamless user experience across devices.

3. Problem Statement

Existing review platforms often suffer from:

- Fake or unverified reviews
- Lack of transparency in rating calculations
- Poor data organization
- Limited administrative control

This project aims to overcome these limitations by implementing a **secure, scalable, and role-based product review management system** using modern web technologies.

---

4. Objectives of the Project

- To design and develop a secure product review platform
- To implement role-based access control for users and administrators
- To ensure data integrity through authentication and authorization
- To provide a responsive and user-friendly interface
- To automate rating aggregation and review analytics
- To follow industry-standard software development practices

---

5. System Requirements

5.1 Hardware Requirements

- **Processor**: Intel Core i3 or higher
- **RAM**: Minimum 4 GB (8 GB Recommended)
- **Storage**: 500 MB free disk space
- **Internet Connectivity**: Required

5.2 Software Requirements

- **Operating System**: Windows 10/11, macOS, Linux
- **Frontend Framework**: React.js (Vite 6)
- **Backend Framework**: Node.js with Express.js
- **Database**: MongoDB (Local / Atlas)
- **Programming Languages**: JavaScript, HTML, CSS
- **Development Tool**: Visual Studio Code

---

6. System Architecture and Design

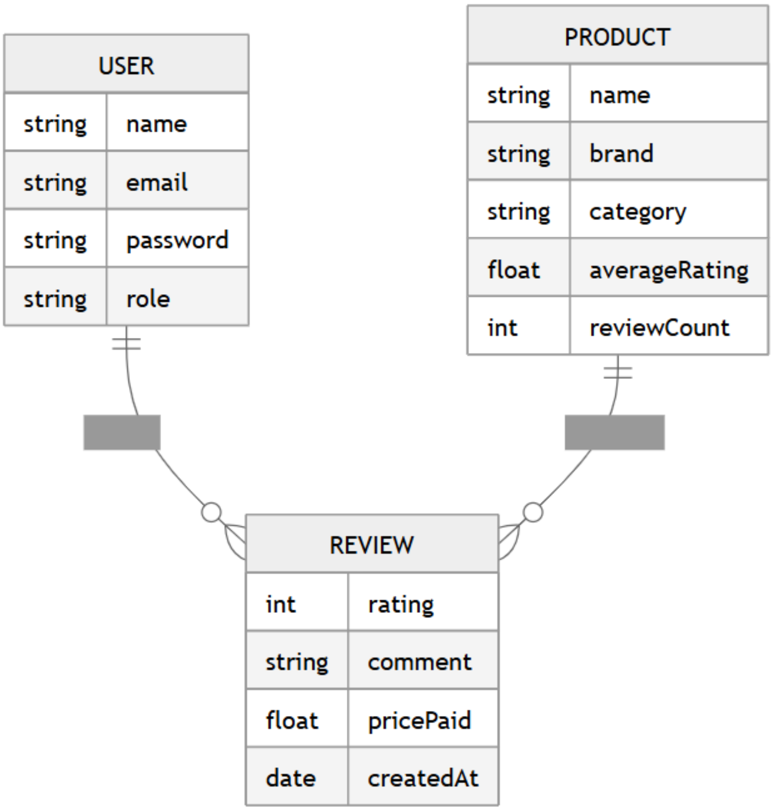
6.1 Architectural Overview

The system follows a **three-tier architecture**:

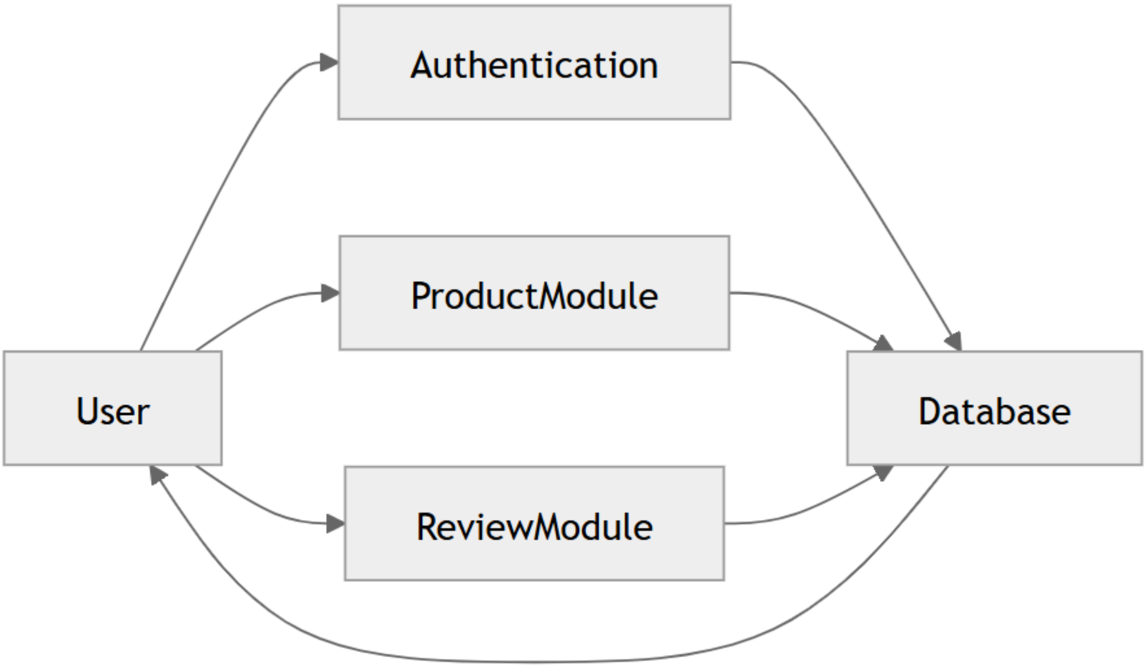
1. **Presentation Layer** – React-based frontend
2. **Application Layer** – RESTful backend API
3. **Data Layer** – MongoDB database

This modular architecture ensures scalability, maintainability, and separation of concerns.

6.2 Entity Relationship Diagram



6.3 Data Flow Diagram



7. Database Design

7.1 User Collection

- Stores authentication credentials
- Implements hashed password storage
- Maintains user roles (Admin/User)

7.2 Product Collection

- Stores product metadata
- Maintains average rating and review count
- Supports efficient filtering and searching

7.3 Review Collection

- Maintains user-generated reviews
- Links users and products
- Enforces rating validation rules

---

8. API Design and Endpoints

API Endpoint	Method	Access Type	Description
/api/auth/login	POST	Public	User authentication
/api/auth/register	POST	Public	User registration
/api/products	GET	Public	Fetch all products
/api/products/:id	GET	Public	Fetch product details
/api/reviews	POST	Authenticated	Submit a review
/api/admin/dashboard	GET	Admin	Analytics & statistics

---

9. Security Mechanisms

- **JWT Authentication** for secure session handling
- **Bcrypt Encryption** for password protection
- **Middleware-based Authorization**
- **CORS Configuration** to restrict unauthorized requests
- **Input Validation & Sanitization**

10. Frontend Implementation

- Global authentication state using **Context API**
- Protected routing for authorized access
- Responsive layout using **CSS Grid & Flexbox**
- Modern UI design principles for enhanced usability

11. Testing Strategy

- Manual unit testing of APIs
- Validation testing for user inputs
- Authentication and authorization testing
- Cross-browser compatibility testing

12. Future Enhancements

- Image-based product reviews
- Social media authentication
- AI-based sentiment analysis
- Recommendation system
- Advanced admin analytics

13. Conclusion

The **Product Review System** successfully fulfills its objectives by delivering a secure, scalable, and user-centric application. It demonstrates strong adherence to software engineering principles and reflects real-world industry development standards. The system is robust, extensible, and suitable for deployment in commercial environments.

14. References

1. MongoDB Official Documentation
2. React.js Developer Guide
3. Node.js Documentation
4. Express.js API Reference