International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Stationary Connect

Dr P Deepthi[1],Boddupally Deepthi[2],Shrungaram Divija[3],Mohmmad Sumayya[4]
[1]Associate Professor;Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India
[2,3,4]B.Tech Students;Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India
Mail Id; deepthi.putnala@slv-edu.in[1],boddupallydeepthi23@gmail.com[2],divija.gpet@gmail.com[3], sumayyamd4@gmail.com[4]

Abstract
Stationary Connect is a campus-oriented web application developed to modernize the exchange of stationery products among students through a secure digital marketplace. The platform enables users to post, discover, and purchase new or pre-owned stationery items within their college environment, minimizing reliance on traditional physical stores and informal peer networks. Built using the MERN technology stack—MongoDB, Express.js, React.js, and Node.js—the system delivers a responsive interface, scalable architecture, and efficient data handling. To improve user interaction, real-time messaging is integrated through Socket.io, allowing direct communication between buyers and sellers. Security is strengthened through JSON Web Token (JWT) authentication and bcrypt-based password hashing for protected access and data privacy. In addition to convenience and affordability, the platform encourages the reuse of notebooks, calculators, drawing tools, and other academic materials, thereby reducing waste and supporting sustainable consumption practices. Stationary Connect demonstrates how digital solutions can streamline campus commerce while fostering transparency, accessibility, and environmental responsibility in student communities.

Keywords: *Campus marketplace, Stationery exchange, MERN stack, Real-time chat, JWT authentication, Sustainable commerce, Student e-commerce platform.*

Introduction
The rapid adoption of digital technologies has transformed the way goods and services are exchanged, creating new opportunities for localized online marketplaces. Within academic institutions, students frequently require stationery materials such as notebooks, pens, drawing tools, calculators, and printing accessories for their daily educational activities. However, obtaining these items often depends on nearby physical stores or unorganized peer networks, which may not always be convenient, economical, or reliable. To address this challenge, Stationary Connect is proposed as a dedicated web-based platform that digitizes the purchase and sale of stationery products within college campuses and student communities.The platform offers a centralized environment where users can securely register, upload product listings, search available items, and complete transactions through an intuitive interface. By connecting buyers and sellers directly, the system reduces unnecessary delays, improves product visibility, and creates a more transparent pricing structure. In addition, students can access required materials without leaving campus premises, saving both time and effort. The platform also encourages the reuse of lightly used stationery items, contributing to cost savings and environmentally responsible consumption practices.Developed using the MERN stack, consisting of MongoDB, Express.js, React.js, and Node.js, the application ensures responsive performance, scalable architecture, and efficient management of user and product data. Real-time communication between users is enabled through Socket.io, allowing buyers and sellers to negotiate or clarify product details instantly. Security mechanisms such as JSON Web Token (JWT) authentication and bcrypt password hashing are incorporated to protect user accounts and sensitive information. Through these features, Stationary Connect provides a modern, secure, and efficient solution for campus-based stationery exchange.

Existing System
In many educational institutions, students traditionally obtain stationery products through local shops, campus stores, or informal exchanges with classmates and seniors. Although functional, these methods often present several limitations. Physical stores may have restricted operating hours, limited stock availability, or higher prices. Students may also need to travel outside campus to purchase urgent materials, which can be inconvenient during examinations or project deadlines.Informal peer-to-peer transactions are equally challenging because they lack a structured communication channel, transparent pricing model, and reliable product information. Buyers may struggle to identify available items, while sellers often have no organized platform to advertise unused materials. Miscommunication regarding prices, product condition, or meeting locations is common. Furthermore, manual transactions provide no systematic record management or verification process, reducing trust and efficiency. These drawbacks highlight the need for a dedicated digital

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

platform specifically designed for student stationery exchange.

Proposed System

Stationary Connect introduces an integrated online marketplace tailored to the stationery needs of students and campus communities. The proposed system enables users to create accounts as buyers or sellers, allowing smooth participation in product transactions. Sellers can upload stationery listings with images, descriptions, prices, and stock status, while buyers can browse products using search and filtering features to locate required items quickly.

A real-time chat facility powered by Socket.io allows direct communication between buyers and sellers, making negotiation and clarification faster and more convenient. Secure user authentication is implemented using JWT, while bcrypt encryption safeguards stored passwords. MongoDB is used for efficient storage of user profiles, product data, and transaction records. Administrative controls are also included to monitor users, verify listings, manage inappropriate content, and maintain overall platform reliability.By combining accessibility, affordability, transparency, and sustainability, the proposed system modernizes stationery exchange within educational environments and creates a practical digital ecosystem for students.

REQUIREMENT ANALYSIS

Functional Requirements

The functional requirements of Stationary Connect define the core operations expected from each user role within the platform. The system is designed for three primary actors: buyer, seller, and administrator.A buyer must be able to create an account, securely log in, browse available stationery products, place orders, track order progress, communicate with sellers through an integrated chat system, and log out after completing activities. These features allow students to conveniently purchase required items through a digital interface.A seller, who may be a student or vendor, must be able to register, log in, create product listings, edit product details, update stock availability and pricing, review incoming orders, process customer requests, and log out securely. These functions enable efficient management of stationery inventory and customer interactions.The administrator is responsible for platform supervision and control. Administrative functions include secure login, user management, verification of product listings, monitoring of transactions, handling misuse reports, and logout. This role ensures that the system remains reliable, organized, and secure for all participants.

Computational Resource Requirements

The successful deployment of Stationary Connect requires suitable hardware and software resources. A standard development machine with an Intel Core i5 processor or equivalent, at least 8 GB RAM, and 256 GB SSD storage is adequate for development and testing activities.The recommended software environment includes Windows 11 as the operating system and Visual Studio Code as the primary development environment. React.js is used for frontend development, while Node.js with Express.js supports backend services. MongoDB is employed as the database system for storing user records, products, chats, and order data. These resources provide a stable environment for building and maintaining the application.

DESIGN

Architecture

System architecture describes how software components interact to process user requests and manage data efficiently. Stationary Connect adopts a layered and modular architecture to support scalability, maintainability, and secure communication. The architecture can be viewed from two perspectives: software architecture and technical architecture.

Software Architecture

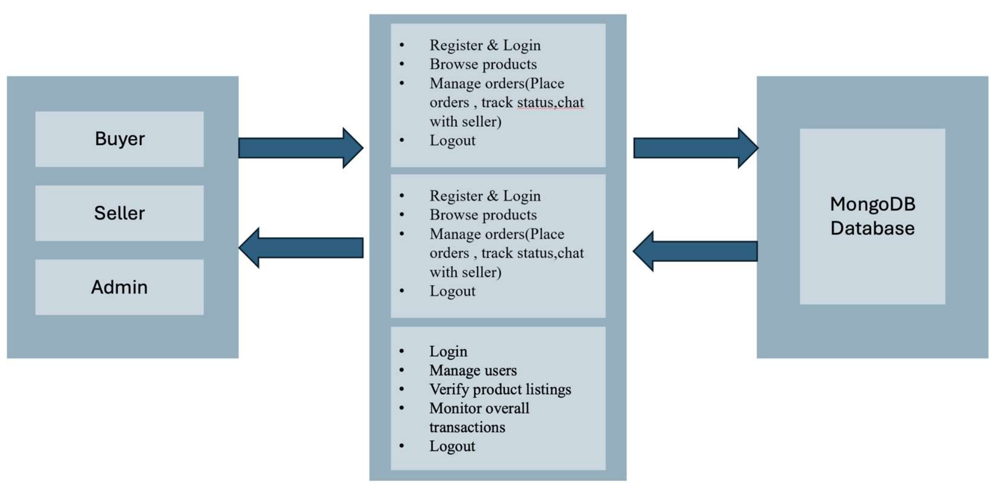

Fig 1 Software architecture

The software architecture is organized into independent functional modules such as authentication, product management, order management, messaging, and administration. This modular approach simplifies development and allows each subsystem to evolve independently.
The frontend is developed using React.js together with HTML, CSS, and JavaScript to deliver responsive pages for dashboards, product browsing, chat, and account management. The backend is implemented using Node.js and Express.js, where business logic, API handling, authentication, and transaction processing are managed.
MongoDB stores product details, user profiles, orders, and communication records. Data exchange between frontend and backend occurs through RESTful APIs. Real-time interaction between buyers and sellers is achieved using Socket.io, enabling instant communication and timely order coordination.

Technical Architecture

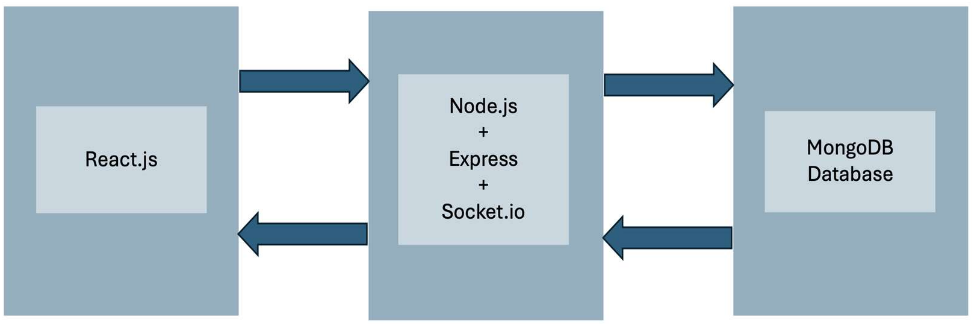

Fig 2 Technical architecture

The technical design follows a three-tier architecture composed of presentation, application, and database layers. The presentation layer includes all user-facing interfaces built with React.js. The application layer contains server-side services implemented with Node.js and Express.js. The database layer uses MongoDB for persistent storage.
When a user performs an action such as searching products or placing an order, the frontend sends a request to the backend. The backend validates the request, interacts with the database, and returns the processed response. This separation improves security, performance, and maintainability.

IMPLEMENTATION

Technologies Used

Stationary Connect is implemented using the MERN stack to create a full-stack web application. React.js is used for building dynamic and responsive user interfaces. Node.js and Express.js are used for backend processing, API creation, and server operations. MongoDB provides a flexible NoSQL database environment.Security is achieved through JWT-based authentication and bcrypt password hashing. Real-time messaging between buyers and sellers is enabled using Socket.io. Development activities are managed using Visual Studio Code, npm, Git, and GitHub.

Module Implementation

The registration module validates user details, stores encrypted credentials, and generates secure login tokens. The product management module allows sellers to create, edit, and remove listings. The order module handles purchase requests and status updates. The chat module supports real-time conversation between users. The admin module provides tools for user control, listing moderation, and transaction oversight.Each module communicates through REST APIs and follows reusable coding practices for easier maintenance.

TESTING

Software testing was conducted to ensure that Stationary Connect functions correctly, securely, and efficiently. Since the application handles user accounts, product listings, transactions, and communication, systematic validation was necessary.Modules tested include registration, login, product management, order handling, chat functionality, and administrator controls. Database validation ensured correct storage and retrieval of records. Security testing confirmed authentication integrity, while performance testing assessed system responsiveness under multiple-user conditions.

Testing Dimensions

Testing activities were performed across multiple layers including user interface, APIs, and database operations. Different scales of testing such as unit testing, integration testing, and system testing were considered. Functional, security, and performance testing methods were applied through both manual and automated approaches.

Testing Life Cycle

The testing process followed six stages: requirement analysis, test planning, test case preparation, environment setup, execution, and closure. This sequence ensured comprehensive validation of all project modules before deployment.

Types of Testing

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

Black box testing was used to verify system behavior without examining source code. It was applied to registration, login, ordering, and search features. White box testing focused on internal logic such as API routing, authentication conditions, database queries, and transaction workflows. Statement, branch, and path coverage methods were considered to improve code reliability.

Test Results

Major test cases including account creation, valid and invalid login attempts, product addition, search operations, order placement, status updates, chat messaging, navigation, editing, deletion, and dashboard display were executed successfully. Expected outputs matched actual results, indicating stable system performance and readiness for deployment.

Screenshots

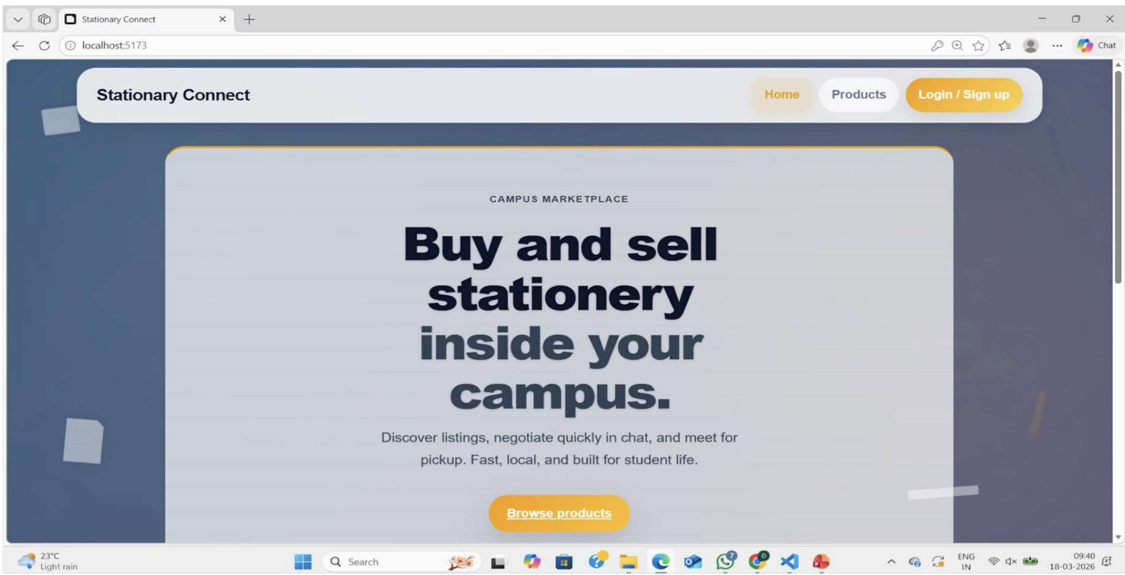

Fig 1 Home page

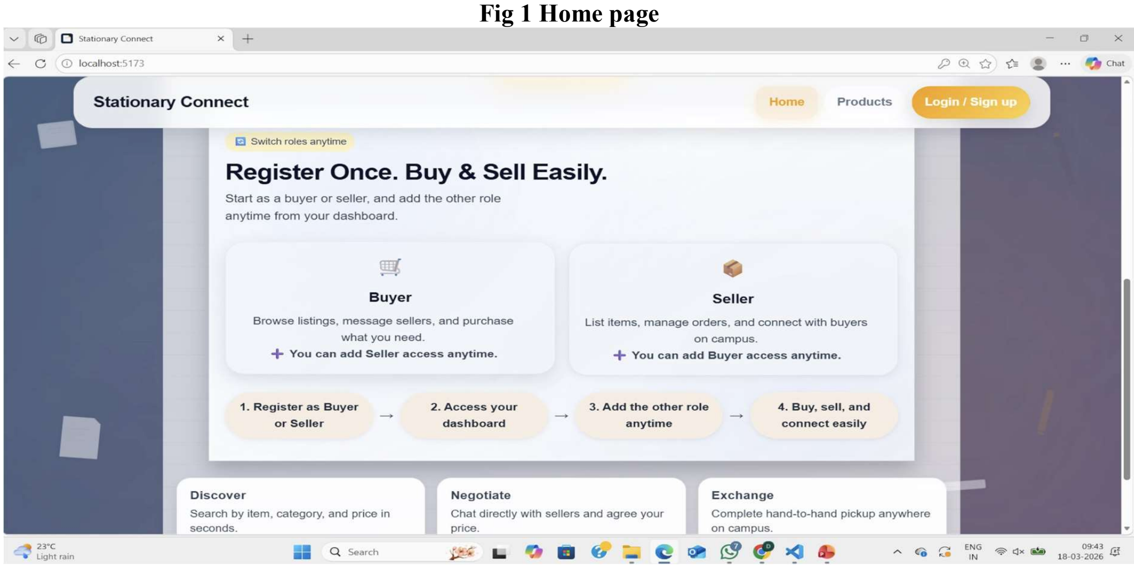

Fig 2 Home page

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

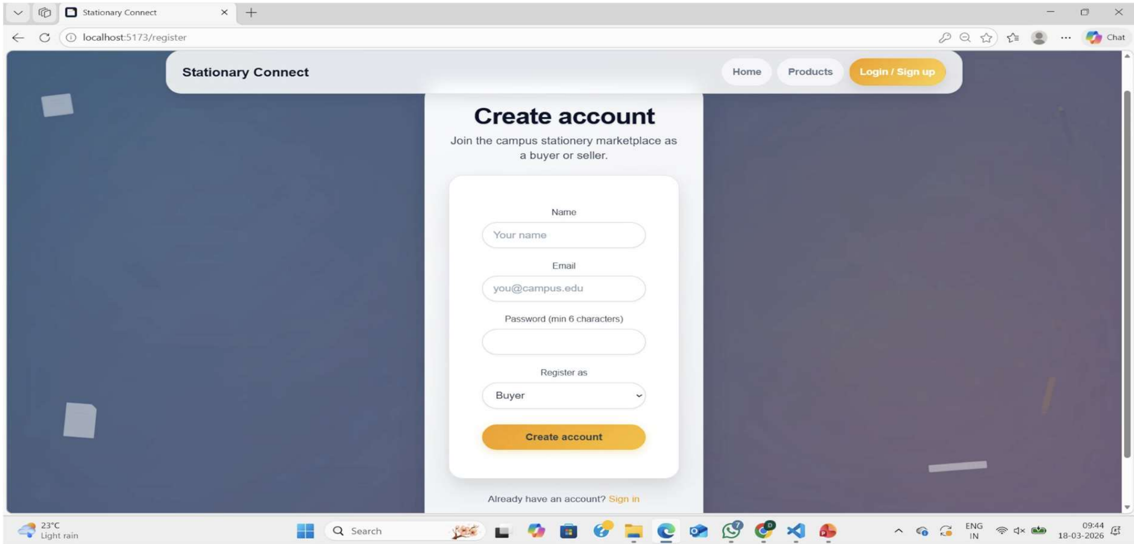

Fig 3 Register page

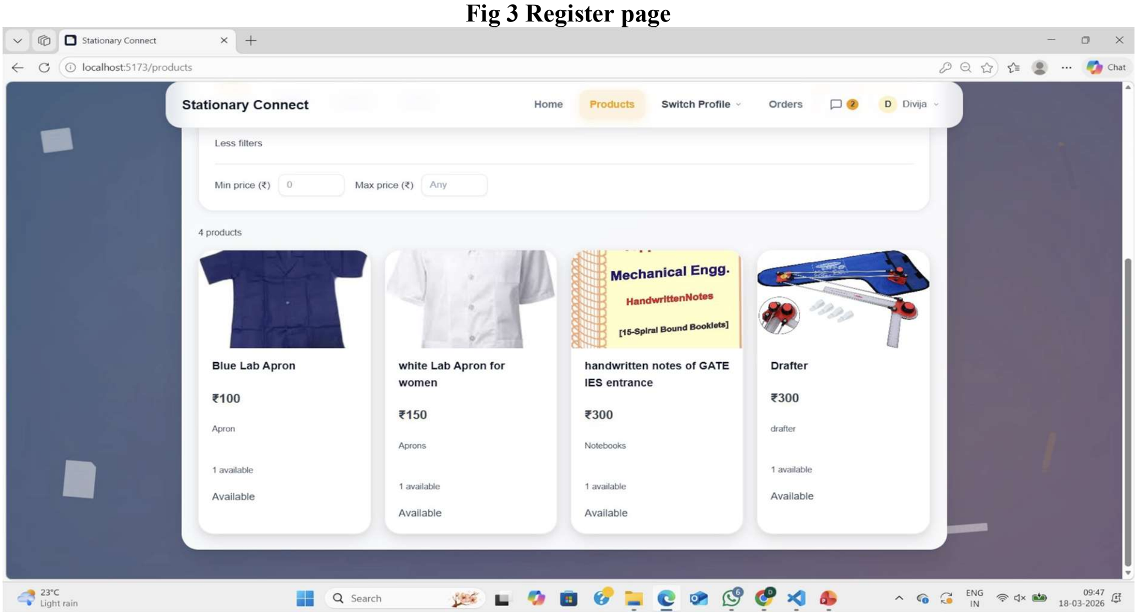

Fig 4 Products page

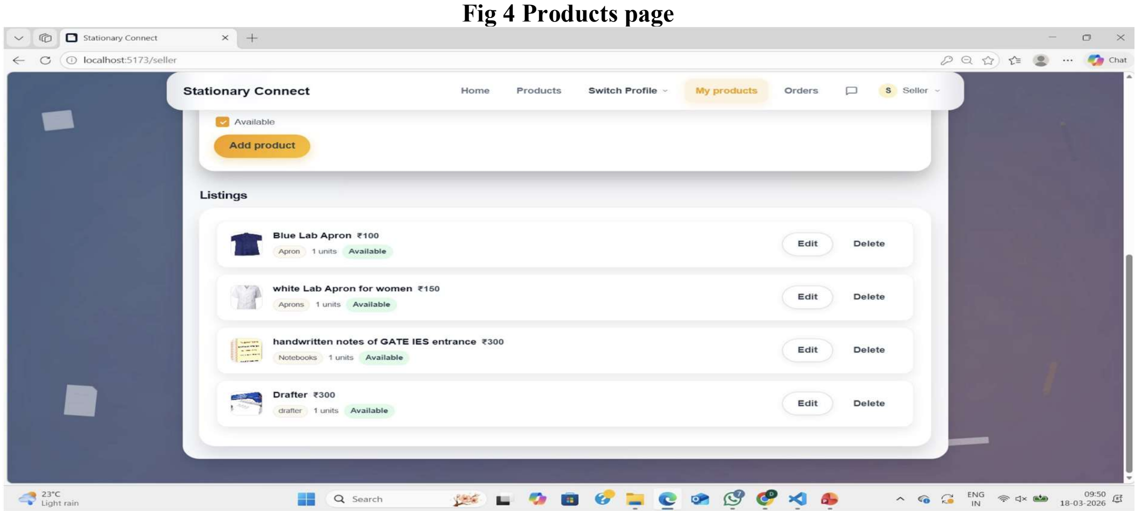

Fig 5 Seller product listings

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

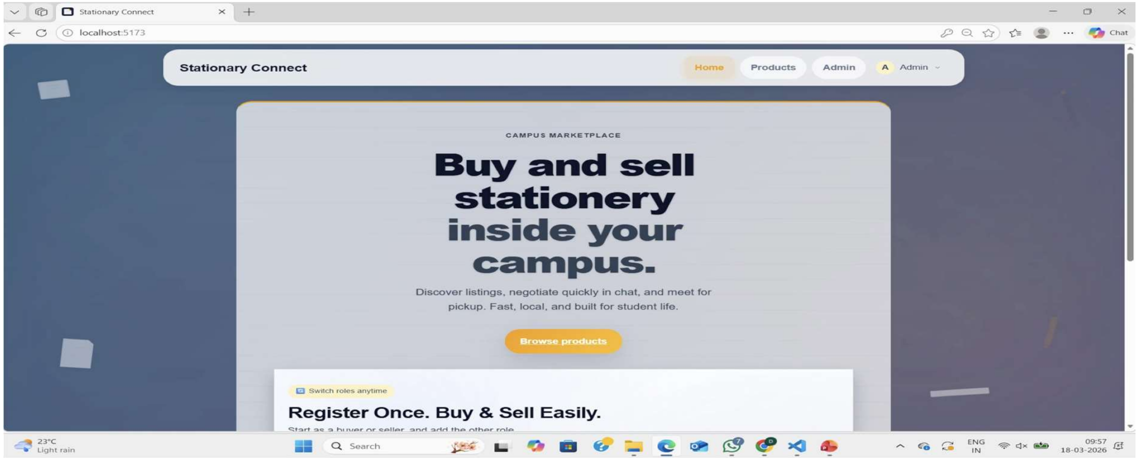

Fig 6 Admin dashboard

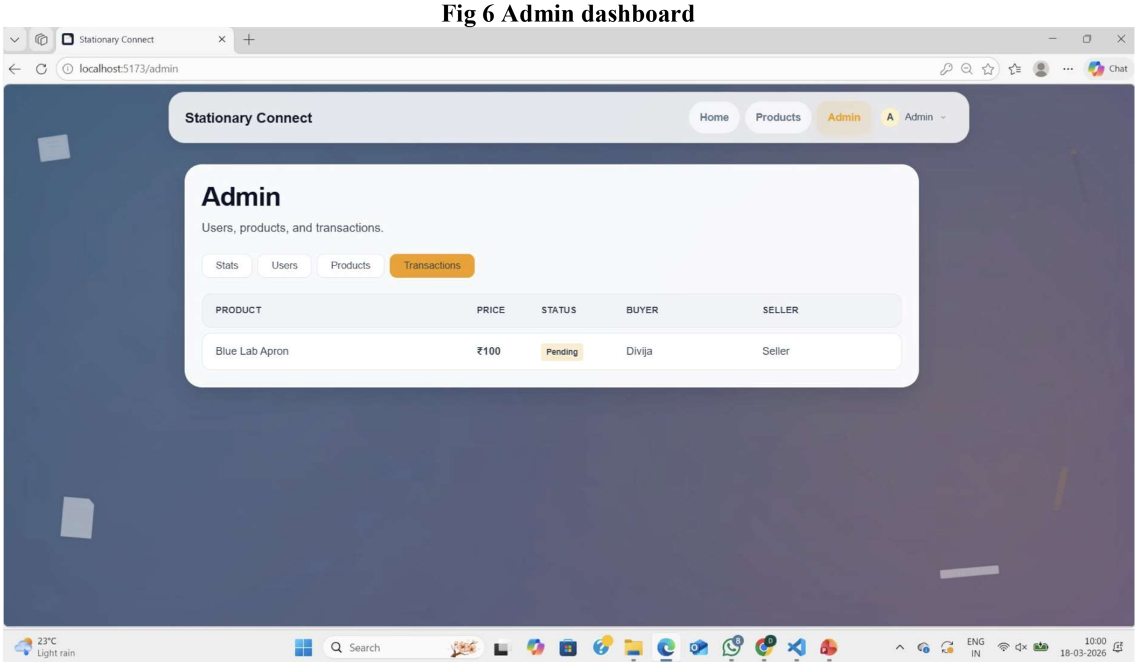

Fig 7 Transaction monitoring (pending state)

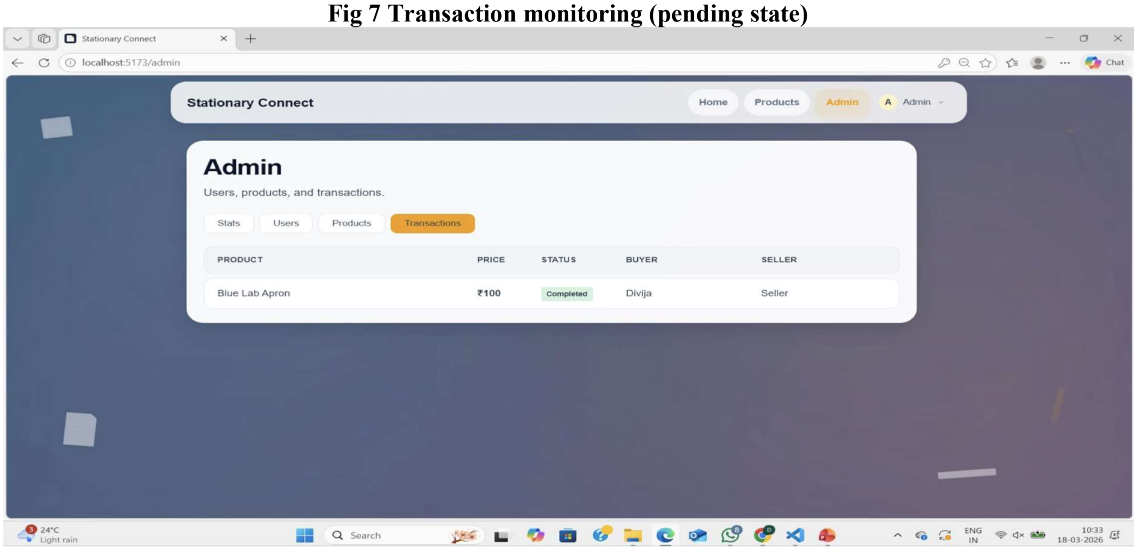

Fig 8 Transaction monitoring (completed state)

Conclusion

Stationary Connect presents an effective digital marketplace designed to simplify the exchange of stationery products within academic communities. The platform offers students and local users a convenient method to buy or sell required materials through a centralized online environment. By replacing unstructured offline transactions with an organized system, the application reduces time consumption, improves accessibility, and supports fair pricing.The integration of secure authentication mechanisms, transparent product listings, and real-time communication enhances user confidence and transaction efficiency. Buyers can quickly locate needed items, while sellers gain a dedicated space to reach potential customers. In addition to commercial benefits, the platform encourages the reuse of notebooks, calculators, drawing tools, and other educational materials, thereby reducing waste and promoting sustainable consumption practices.Built on scalable web technologies, Stationary Connect demonstrates strong potential for long-term adoption across colleges and universities. Overall, the project provides a practical, secure, and environmentally responsible solution for connecting stationery buyers and sellers through modern digital infrastructure.

Future Scope

The future development of Stationary Connect can significantly expand its usability and impact. A dedicated mobile application for Android and iOS platforms can improve accessibility and allow users to manage transactions from anywhere. Integration of secure online payment gateways would enable seamless financial transactions and reduce dependency on cash-based exchanges.Advanced search capabilities powered by artificial intelligence can be introduced to provide personalized recommendations, smart filtering, and demand forecasting. Enhanced seller dashboards may include sales analytics, inventory tracking, and promotional tools to improve business performance. Blockchain-based transaction records could also be explored to increase transparency and trust.Further sustainability initiatives may include reward systems for reusing products, carbon-saving metrics, and partnerships with campus recycling programs. With these enhancements, Stationary Connect can evolve into a comprehensive smart-commerce ecosystem for educational institutions.

9. REFERENCES

[1] V. Goyal, A. K. Mishra, and D. Singh, “Implementation and Comparison of MERN Stack Technology with HTML/CSS, SQL, PHP & MEAN in Web Development,” *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 5, no. 11, Nov. 2023.

[2] S. Raj and R. Dutta, “Design and Implementation of an E-Commerce Web Application,” *International Journal of Computer Applications*, vol. 179, no. 41, pp. 1–5, May 2018.

[3] “The MERN Stack Revolution: A Review of its Impact on Modern Web Development,” *International Journal of General Science and Technology (IJGST)*, Apr. 2024.

[4] “Exploring MERN Stack and Tech Stacks: A Comparative Study,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 10, no. 12, Dec. 2023.

[5] “Accelerating Full-Stack Development with the MERN Technology Stack,” *ResearchGate Publication*, Jul. 2025.