iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79658

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Development of a Full-Stack Student Marketplace Platform Using React.js and Firebase

Satyam Gupta[1], Kumud Dixit[2]
[1]*Student,* [2]*Assistant Professor, Department of Information Technology, Madhav Institute of Technology & Science. Gwalior, India*

Abstract: ***Students still rely on informal methods like social media or peer networks to exchange academic resources such as peer networks, notice boards, and social media groups. These methods are unorganized, making it difficult to find items and interact reliably, lack of trust between participants, and inefficient communication processes. To address these challenges, this paper presents Studimart, a student-focused full-stack web platform designed to streamline the exchange of academic materials within campus communities.***

The system utilizes React.js for the front-end interface and Firebase for backend services such as authentication, real-time database management, and cloud storage [3], [4]. The application enables secure user registration, dynamic product listing, real-time updates, and efficient search functionality using multi-parameter filtering. Additionally, the platform supports image-based listings to improve user engagement and enhance product clarity. This paper outlines the architectural design, implementation strategies, technological choices, and system limitations of the proposed system. It further discusses the scalability potential of cloud-backed and serverless development approaches in building cost-effective and efficient student-centric digital marketplaces [2], [3], [9].

Keywords: Student marketplace, React.js, Firebase, full-stack web application, serverless architecture, Cloud Firestore, academic e-commerce, peer-to-peer platform.

I. INTRODUCTION

Students regularly buy books, notes, and other materials for each semester, but these are rarely reused after the course ends. such as textbooks, reference materials, laboratory manuals, printed notes, and essential stationery. These materials play a crucial role in supporting academic performance; however, their usage is often limited to a specific semester or course duration. Once the course is completed, a large portion of these resources becomes redundant, leading to inefficiencies in both utilization and cost. This creates a recurring need for systems that support redistribution of academic materials within student communities.Even though students need to exchange these items, they mostly rely on informal methods like word-of-mouth or social media. Existing methods such as word-of-mouth communication, notice boards, and social media groups lack structure, searchability, and reliability. As a result, students face difficulties in discovering relevant items and interacting with trustworthy sellers. Studies on student-oriented digital platforms indicate that usability, accessibility, and perceived usefulness significantly influence user adoption and engagement in academic environments [7]. As a result, it becomes difficult for students to find relevant items, trust sellers, and complete transactions smoothly.Existing platforms do not fully meet this specific need. Large-scale e-commerce platforms such as Amazon and Flipkart are designed for mass-market transactions and do not cater to localized, campus-specific exchanges. Their systems are optimized for new products, formal sellers, and logistics-driven delivery models, making them unsuitable for peer-to-peer academic trading. On the other hand, general classified platforms like OLX and Facebook Marketplace enable user-driven listings but lack domain-specific categorization and filtering tailored to academic content. Additionally, these platforms often introduce challenges such as irrelevant listings, geographical mismatches, and insufficient safeguards for student users, thereby reducing their effectiveness in an educational context.To bridge this gap, this work proposes Studimart, a dedicated full-stack web application specifically designed for student communities. Studimart addresses this gap by providing a structured platform for listing and discovering academic products. The system was built using React.js and Firebase, leveraging a cloud-based architecture to reduce infrastructure complexity and improve scalability [3], [4], [9]. Since the platform is focused only on student use within a campus, the platform enhances relevance, improves discoverability, and promotes a more efficient transaction process. From a technological perspective, Studimart is implemented using React.js to build a dynamic and responsive user interface, combined with Google Firebase as a Backend-as-a-Service (BaaS) solution. The adoption of a serverless architecture eliminates the need for traditional server management while providing scalable real-time data synchronization and secure authentication services.

This approach not only simplifies development but also ensures cost-effectiveness and ease of deployment, making it suitable for student-oriented applications in the future.

The primary contributions of this work are the designing of student centered marketplace with the server-less cloud stack also security is major factor to overcome this authentication is used for secure role based access with real time listing management with cloud Firestore

II. RELATED WORK

The design of Studimart is based on concepts from usability, cloud computing, and web development. Although the present system addresses a specific student marketplace use case, its technical foundation is supported by a combination of widely accepted design and development principles.Usability plays a central role in the design of interactive systems. Nielsen emphasizes that consistency, visibility, feedback, and user control are essential to reducing friction in user interfaces and improving learnability [1].From a frontend development perspective, React provides a component-based approach for building user interfaces that are modular, reusable, and efficient to maintain [4]. Such architecture is well suited to single-page applications in which interface sections, such as navigation, authentication forms, filters, and product cards, must be dynamically updated. Research on higher-education platforms highlights the importance of intuitive design and efficient navigation for student users [11]. Additionally, recent studies on student marketplace systems have demonstrated the feasibility of campus-based resource exchange platforms [12], [13].Research on digital collaboration platforms among students highlights that usability, interactivity, and affordability are key factors influencing adoption in academic settings. These findings indicate that students prefer platforms that are simple, efficient, and accessible, which directly supports the design considerations of student-centric systems, such as Studimart [7].

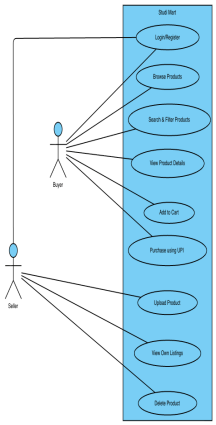


Fig. 1: Use Case Diagram of Studimart

The backend design of systems like Studimart is supported by cloud-based and data-intensive application principles. Modern systems need scalable data handling and efficient data updates, efficient synchronization, and support for changing application state [2]. Cloud-based platforms offer flexible infrastructure and service abstraction, reducing deployment overhead and enabling rapid application development [9]. Firebase, in particular, provides integrated support for authentication, database services, and storage, making it suitable for full-stack applications with real-time interaction requirements [3].Responsive design is another important consideration for modern student-facing applications, since users may access the platform from laptops, tablets, or smartphones. Finally, intelligent recommendation techniques are increasingly relevant in digital platforms that need to improve discoverability and personalization. AI-supported recommendation approaches can help users find relevant content more efficiently by analyzing patterns, preferences, or contextual information [10]. This provides a meaningful direction for future enhancement of the proposed system.

StudiMart follows a three-tier architecture with presentation, logic, and data layers, a presentation layer, an authentication/logic layer, and a persistence layer all realized using React.js and Firebase managed cloud services. The system is stateless on the server side, with application state managed at the client level or within Firebase.As illustrated in Fig. 2 the architecture follows a client-server model using React for the frontend and Firebase services for authentication, database, and storage operations.

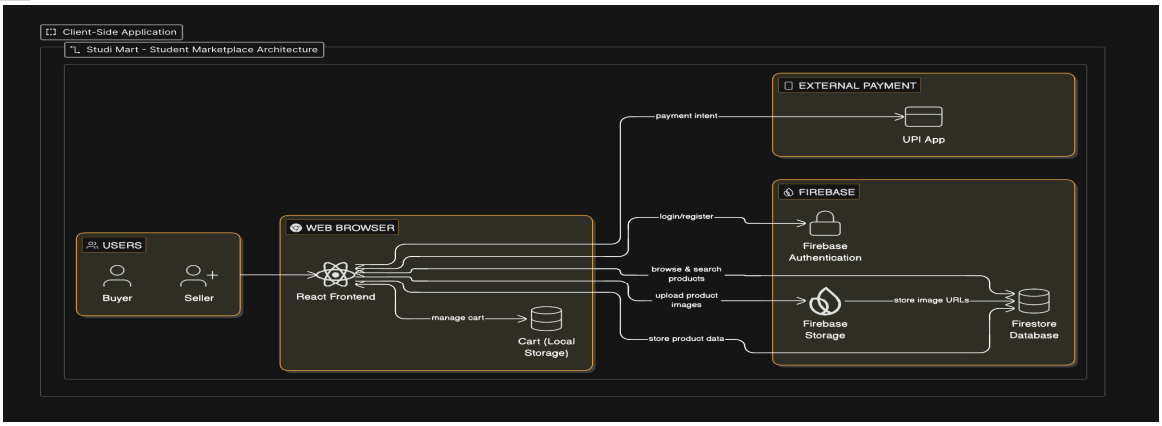


Fig. 2: System Architecture of Studimart Platform

TABLE I
SYSTEM ARCHITECTURE AND TECHNOLOGY STACK

Layer	Technology	Role
Presentation	React.js (Hooks, Router)	UI rendering & routing
Auth	Firebase Authentication	Login, session, OAuth
Database	Cloud Firestore (NoSQL)	Real-time listings & profiles
Storage	Firebase Storage	Product image uploads
Hosting	Firebase Hosting / Vercel	Global CDN deployment

A. Presentation Layer

The frontend was built in React.js using functional components with Hooks (useState, useEffect, useContext) for state and lifecycle management. React Router v6 enables client-side navigation for SPA. The component hierarchy is modular, with discrete components for navigation, listing cards, search, product detail view, and authentication forms[4].

Responsive design principles are applied to ensure compatibility across devices using flexible layouts and media queries [5].

This layer ensures:

- Smooth navigation using client-side routing
- Responsive design across multiple devices
- Interactive UI components for better user engagement

B. Authentication Layer

Firebase Authentication was used to manage user identity and session persistence [3]. Protected routes ensure that only authenticated users can access the restricted features. A React Context API authentication context is exposed globally, enabling any component to determine the user state and restrict access to protected routes via a ProtectedRoute component.

Key features include:

- Email/password-based authentication
- Persistent login sessions
- Role-based access control for protected routes

C. Data and Storage Layers

Cloud Firestore serves as the primary database.Cloud Firestore is used as a NoSQL database to store listings and user data. Its real-time update capability ensures dynamic content updates [3], [8]. Product documents within the 'listings' collection store the title, category, description,price, seller ID, contact information, timestamp, and image URL. Firestore's real-time listeners (onSnapshot) automatically update the product feed without page refresh. Firebase Storage persists product images, with download URLs stored in the corresponding Firestore documents.

The backend relies on Firebase services:

- Cloud Firestore: Stores product listings, user data, and metadata
- Firebase Storage: Stores product images

III. IMPLEMENTATION

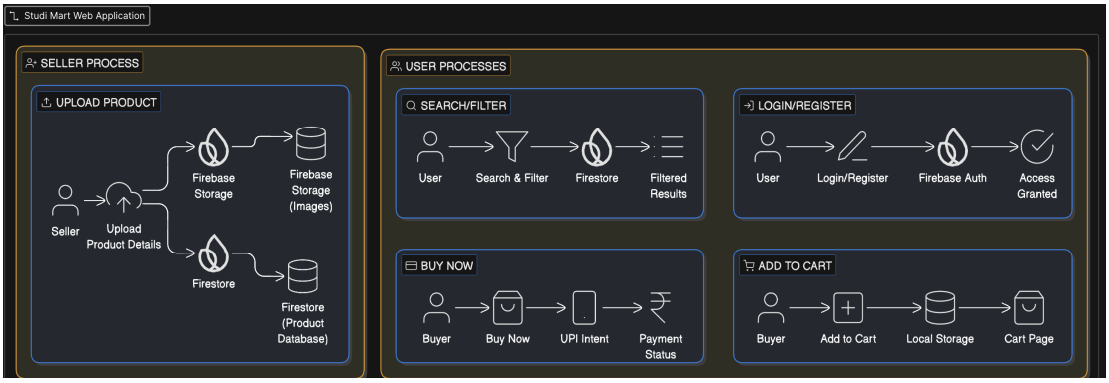


Fig. 3: Data Flow Diagram of Studimart

A. Development Environment

The StudiMart application was developed using a web stack focused on performance and ease of development. The frontend was initialized using Create React App, which provided a customizable environment with Webpack for module bundling and Babel for JavaScript transpilation. This setup enabled the use of modern ES6+ syntax while ensuring compatibility across browsers. Firebase was chosen because it integrates easily with React and simplifies backend development with frontend frameworks and its real-time capabilities. The Firebase JavaScript SDK (v9 modular architecture) was used to optimize the performance through tree-shaking, thereby reducing the overall bundle size. Core Firebase services integrated into the system include Authentication, Firestore Database, and Cloud Storage.Cloud-based architectures reduce infrastructure overhead and improve deployment efficiency [9].

Git was used to manage the code, where features were developed separately and then merged after testing. Each feature or Each feature was developed separately and merged into the main branch after testing. This made the code easier to manage, debug, and update.

B. Authentication Flow

User authentication in StudiMart is handled through Firebase Authentication, providing a secure and reliable identity management system. New users can register using their email and password, where input validation is automatically enforced, including password strength requirements and email format verification.Upon successful registration, user credentials are securely stored, and a session token is generated.

The system then redirects the user to the main marketplace interface. For returning users, Firebase maintains persistent login sessions, allowing automatic authentication without repeated login prompts.To protect restricted sections of the application, a custom ProtectedRoute mechanism is implemented. This component checks the authentication state before rendering specific pages such as product uploads or user-specific content. If a user is not authenticated, they are redirected to the login page, ensuring controlled access and improved security.

C. Product Listing Creation

The product listing feature is one of the main parts of StudiMart. Users can create listings by entering product details and uploading images. Data are stored in Firestore, and real-time updates ensure immediate visibility [3], [8]. enabling users to act as sellers on the platform.

Authenticated users can navigate to the ”Sell” section, where they are presented with a structured form to input product details, which captures essential attributes, including title, category (Books, Notes, Stationery, Electronics, Others), product description, pricing information, and an optional image. When an image is uploaded, it is stored in the Firebase Cloud Storage, and the corresponding download URL is generated. This URL is then linked to the product data stored in Firestore, which is a NoSQL real-time database that allows instantaneous updates across all connected clients. Once a listing is submitted, it becomes immediately visible on the marketplace feed without requiring a page refresh or any other action. React‘s state management ensures that the UI is updated dynamically, reflecting newly added products in real time.

This feature stores data efficiently and can handle more users as the system grows, as Firestore efficiently handles large volumes of structured and semi-structured data.

D. Search and Filter

To make it easier for users to find products, StudiMart incorporates an efficient search and filtering system. When the homepage loads, all active product listings are fetched from Firestore and stored in the local state of the React component. The search functionality is implemented on the client side, where user-entered queries are matched against product titles and descriptions using string comparison techniques. This reduces the number of database requests and improves performance, thereby reducing latency and conserving Firebase read operations.In addition, a category-based filtering system is provided through a dropdown interface. Users can refine the results by selecting specific categories, such as books or electronics. The combination of search and category filters allows compound filtering, enabling users to quickly locate relevant products.Client-side filtering is implemented to improve performance and reduce database load. [1],[6]

TABLE II
FEATURE COMPARISON: STUDIMART VS. EXISTING PLATFORMS

Features	OLX/FB Mkt.	Amazon	Studimart
Student-focused platform	✗	✗	✓
Academic specific categories	✗	**Feature** Partial	✓
Free Peer-to-peer listing	✓	✗	✓
Subject-based filter	✗	Partial	✓
Zero commission	✗	✗	✓
Campus-specific relevance	✗	✗	✓

Table II compares StudiMart with existing platforms, highlighting its student-focused design with academic categorization and subject-based filtering not fully supported by general platforms

E. User Interface Design

The user interface was designed to be simple, easy to use, and responsive across devices to ensure a seamless experience across different devices. A card-based layout is implemented to present each product listing in a structured and visually organized manner[5]. Each card contains essential details such as the product image, title, category label, price, and seller contact information, allowing users to quickly scan and compare available items.The layout dynamically adapts to various screen sizes using responsive design techniques. On larger screens, listings are displayed in a multi-column grid to maximize content visibility, whereas on smaller devices, such as smartphones, the layout transitions into a single-column format for improved readability and touch interaction. This adaptability is achieved using CSS Flexbox along with media queries, ensuring consistent performance across devices.To help users find products easily bu using key elements such as product images and pricing information.

Clear typography, equal spacing, and contrasting color schemes were used to improve readability and reduce cognitive load. Interactive elements, such as buttons and clickable cards, are designed with sufficient size and feedback mechanisms to support interactive navigation.

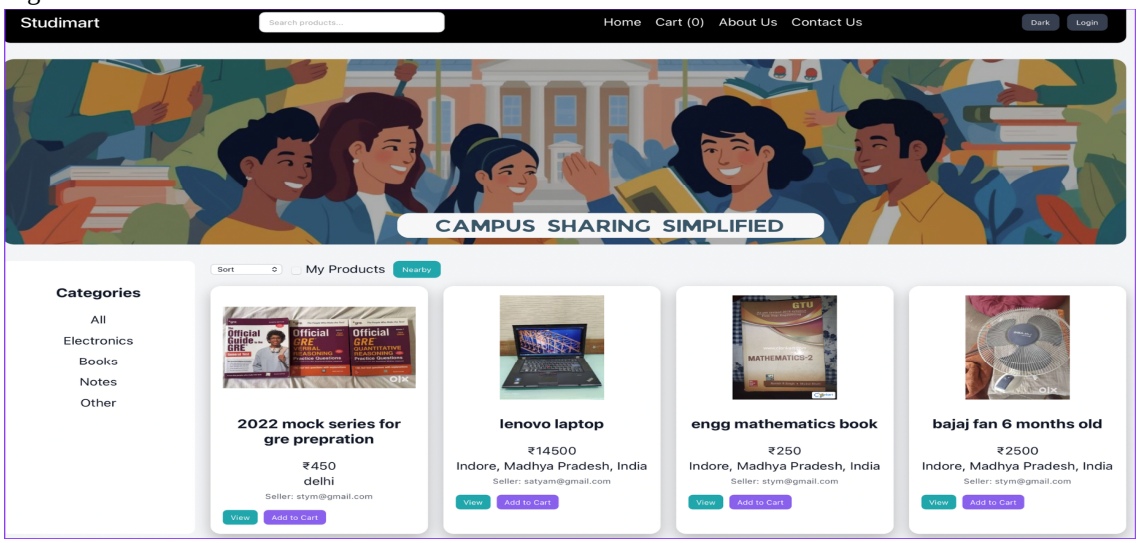


Fig 4: Home page interface displays product listings with filtering and navigation features.

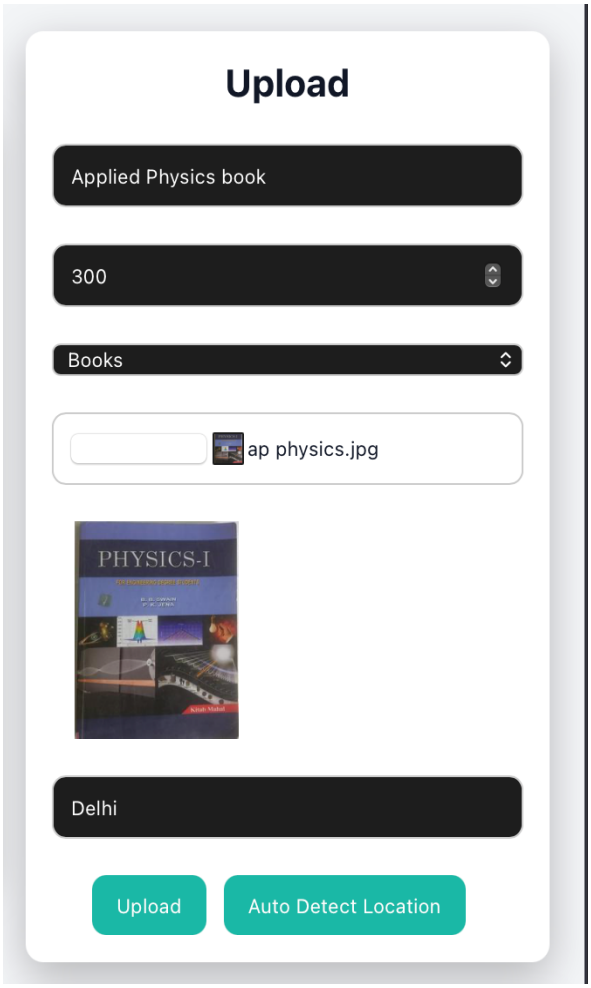


Fig. 5: Upload interface allows users to upload products with auto update location feature with category, price, and image.

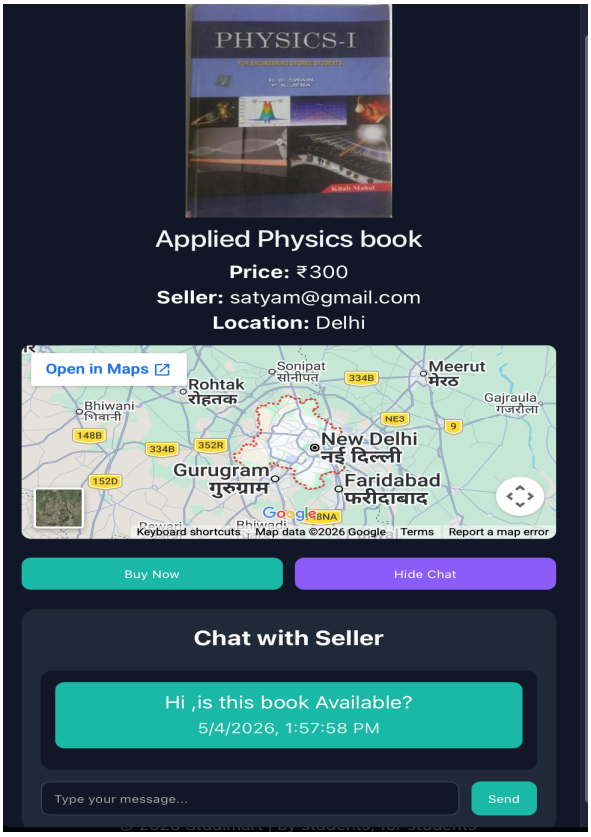


Fig. 6: Product detail modal Shows detailed product info with location and purchase option with chat with seller feature

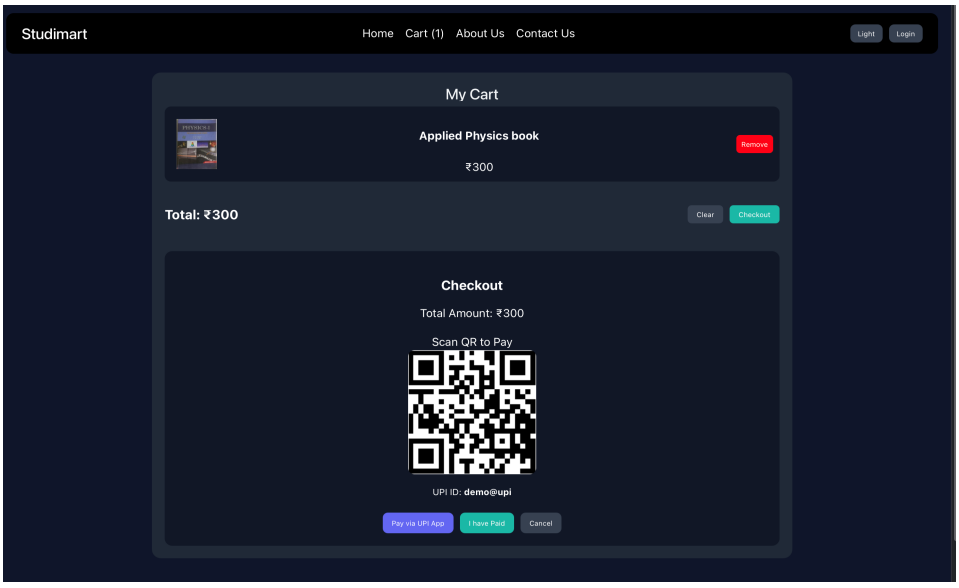


Fig. 7: Cart Page Displays selected items and total price for checkout.

The design also follows basic usability principles to improve user experience inspired by Jakob Nielsen‘s usability engineering guidelines, including consistency, visibility of system status, and user control. These principles help minimize user errors and improve overall efficiency while interacting with the platform. Basic user experience enhancements, such as hover effects, loading indicators, and simple notification messages, were incorporated to provide real-time feedback to users. The interface was kept minimalistic to avoid unnecessary complexity while ensuring that all essential functionalities remained easily accessible[1].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IV. CHALLENGES AND LIMITATIONS

During development, several practical and technical challenges were faced that influenced the design decisions and implementation scope. While the system achieves its core objectives, certain limitations remain owing to time constraints, resource availability, and the chosen technology stack.

One of the primary challenges was ensuring efficient real-time data synchronization using the Firebase services. Although Firestore provides real-time updates, managing a consistent state across multiple components in a React-based architecture requires careful handling of asynchronous operations and state management. Improper handling can lead to delayed UI updates or redundant data rendering. Scalability is another limitation, as increased user activity may lead to performance issues. Data-intensive system design requires optimization strategies, such as indexing and caching [2]. The current implementation is optimized for small-to medium-scale usage, such as within a college environment. As the number of users and listings increases, performance bottlenecks may arise owing to frequent read/write operations in the database and increased network requests. Advanced optimization techniques, such as indexing, caching, and pagination, have not been fully implemented in the current version, and the platform lacks advanced search capabilities. While basic filtering by category or keywords is supported, more sophisticated features, such as personalized recommendations, ranking algorithms, or relevance-based searches, have not yet been integrated. This limits the efficiency of product discovery for users when the dataset becomes larger.

Security is another area with specific constraints. Although Firebase Authentication ensures secure user login and session management, additional layers, such as role-based access control, data validation at multiple levels, and protection against malicious uploads, are minimal. This may cause issues if the system is used on a larger scale.From a user experience perspective, the application currently follows a simplified navigation model. While this reduces complexity, it also limits features such as detailed user profiles, chat systems between buyers and sellers, and order-tracking mechanisms. These features are commonly expected in mature marketplace platforms but are beyond the scope of the current implementation phase.Another limitation is the absence of location-based services. The platform does not yet integrate geolocation or mapping functionalities, which could help users identify nearby sellers and reduce transaction friction. Similarly, online payment integration is not included, requiring users to rely on external communication to complete transactions. Additionally, image handling is basic, with limited optimization for compression or multiple image uploads. This may affect the performance and storage efficiency when handling a large number of listings.

In summary, while StudiMart successfully demonstrates a functional student-centric marketplace system, it remains a foundational prototype. Future improvements can address these limitations by incorporating scalable architecture patterns, enhanced security mechanisms, intelligent search features, and additional integrations such as location services and digital payments.

V. FUTURE SCOPE

The current version of StudiMart provides a basic working system for a student-centric marketplace; however, several enhancements can be incorporated in future iterations to improve usability, scalability, and system intelligence.

One of the key planned improvements is the integration of a real-time in-app messaging system. This can be implemented using Firestore-backed message threads, enabling seamless communication between buyers and sellers without relying on external platforms. Such a feature would enhance user engagement and streamline the transaction process.To strengthen trust and transparency within the platform, a seller rating and review mechanism can be introduced. This system would allow users to provide feedback based on their transaction experience, thereby helping future buyers make informed decisions and encouraging responsible seller behavior.Another important enhancement involves the implementation of push notifications using Firebase Cloud Messaging (FCM). This would allow users to receive instant alerts regarding listing inquiries, price updates, or new relevant products, thereby improving responsiveness and overall user interaction with the platform.

Location-based services can further enhance the platform by enabling geolocation-based discovery of listings. By utilizing browser Geolocation APIs along with Firestore geoqueries, users can identify nearby sellers, making transactions more convenient and efficient, especially within localized environments such as college campuses.

Expanding the platform to mobile devices is also a significant future direction. A cross-platform mobile application developed using React Native would ensure wider accessibility and improved user experience, particularly for users who prefer mobile-first interactions.Additionally, the incorporation of an AI-powered recommendation engine can significantly improve content discoverability. Techniques such as collaborative filtering or content-based filtering can be employed to analyze user behavior and preferences, thereby suggesting relevant listings and enhancing personalization.Overall, these enhancements aim to transform StudiMart from a functional prototype into a more robust, scalable, and intelligent marketplace system capable of supporting a larger and more dynamic user base.

VI. CONCLUSION

StudiMart is a student-focused marketplace that helps in exchanging academic resources more efficiently focused at simplifying the exchange of academic resources such as books, notes, and essential materials. The system successfully integrates a modern web-based frontend with cloud-backed backend services to deliver a functional and accessible solution tailored to student needs.Through the use of React for the user interface and Firebase services for authentication, database management, and storage, the platform achieves seamless interaction flow from user registration to product listing and browsing. The implementation highlights the effectiveness of combining lightweight frontend frameworks with scalable backend-as-a-service solutions to rapidly develop full-stack applications.

The project also emphasizes usability and responsiveness by adopting a clean, card-based interface design that adapts across devices. Core functionalities such as real-time listing updates, structured product display, and secure authentication contribute to a reliable user experience. These features collectively validate the feasibility of deploying a student-centric marketplace within a controlled environment such as an academic institution.At the same time, the study identifies certain limitations related to scalability, advanced search capabilities, and extended feature integration. These constraints provide direction for future improvements and reflect practical considerations encountered during development. During development, some challenges were faced in handling real-time data updates and maintaining UI consistency.

In conclusion, StudiMart serves as a foundational prototype that not only fulfills its primary objective but also offers a scalable framework for future expansion. This project shows how a student focused marketplace can built using modern web technologies by demonstrating how targeted solutions can be developed to address specific community needs using contemporary technologies.

REFERENCES

[1] Jakob Nielsen, Usability Engineering. San Diego, CA, USA: Academic Press, 1993.

[2] Martin Kleppmann , Designing Data-Intensive Applications. Sebastopol, CA, USA: O'Reilly Media, 2017.

[3] Google, "Firebase Documentation," [Online]. Available: https://firebase.google.com/docs.

[4] Meta Platforms, "React – A JavaScript Library for Building User Interfaces, " [Online]. Available: https://react.dev.

[5] World Wide Web Consortium, "Responsive Web Design Basics, " [Online]. Available: https://www.w3.org

[6] Steve Krug, Don't Make Me Think: A Common Sense Approach to Web Usability, 3rd ed. Berkeley, CA, USA: New Riders, 2014.

[7] A. Singh, S. Sharma, and M. Paliwal, "Adoption intention and effectiveness of digital collaboration platforms for online learning," Interactive Technology and Smart Education, vol. 18, no. 4, pp. 493–514, 2021.

[8] S. Saraf, "A Review on Firebase (Backend as a Service) for Mobile Application Development, "International Journal for Research in Applied Science and Engineering Technology, vol. 10, no. 1, 2022.

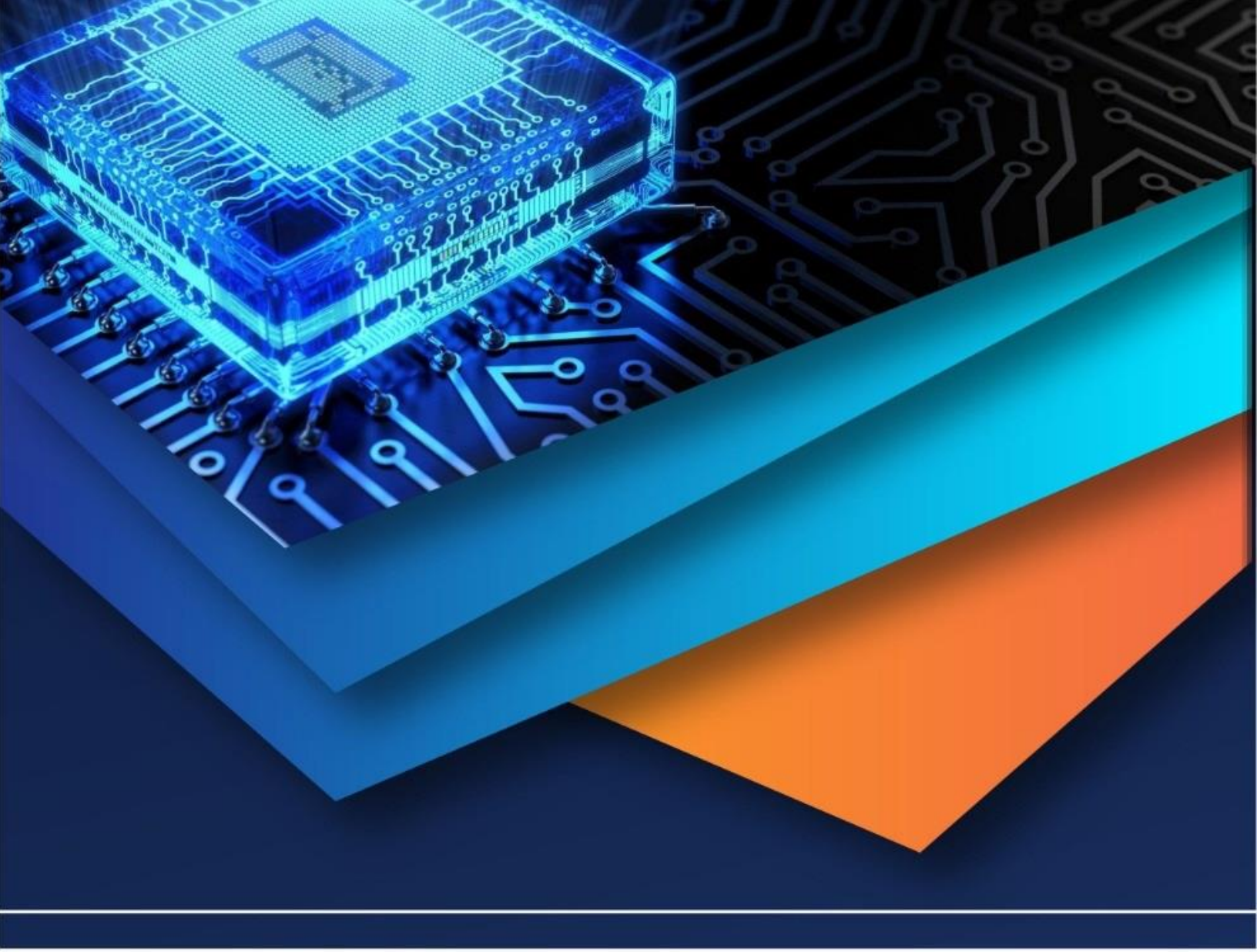
[9] Amazon Web Services, "Cloud Computing Concepts," [Online]. Available: https://aws.amazon.com.

[10] OpenAI, "Artificial Intelligence and Recommendation SystemsOverview, "[Online].Available: https://openai.com

[11] A. Vatankhah, "Design principles for e-learning platforms in higher education," Journal of Computing in Higher Education, 2024.

[12] R. Sharma et al., "Campus Mart: A student marketplace platform for academic resource exchange," International Journal of Research and Technical Innovation (IJRTI), 2023.

[13] S. Patel et al., "Design and implementation of a student marketplace system," International Journal of Emerging Technologies and Innovative Research (IJETIR), 2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY