International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Design and Architecture of a Secure Campus Marketplace Integrated with a Gamified Circular Reward System

Dr.S.Kowsalya1, Aswin S[2], Jony V[3], Tharun V G[4], Mohan K[5], Rishvaneh D[6]

[1]Assistant Professor, Department of Computer Applications, Sri Krishna Arts and Science College, Coimbatore.
[2,3,4,5,6]Students, Department of Computer Applications, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

Abstract
Large college campuses generate a constant stream of small-scale commerce: textbooks, lab equipment, and hostel accessories change hands every semester, yet most of this trade is still coordinated through notice boards, informal chat groups, or open classified sites that were never built for a closed, trust-based community. Campus Bazaar is a full-stack, single-institution, peer-to-peer trading platform restricted to verified student accounts, layering a gamified Campus Credits engine on top of a conventional buy-and-sell marketplace to reward reuse rather than treat it as an incidental side effect of resale. This paper documents the platform's three-tier architecture, its fault-tolerant dual-adapter persistence layer, its JWT-based authentication model, and its supporting Lost-and-Found, messaging, and AI-assisted help modules. It then evaluates the system against measured performance and reliability criteria, weighs its practical trade-offs against the constraints of limited institutional infrastructure, and outlines a roadmap toward AI-assisted pricing, encrypted peer negotiation, and multi-campus federation.

Keywords: verified student marketplace, gamified rewards engine, circular economy, fault-tolerant persistence, full-stack web application, JSON Web Tokens.

1. Background and Motivation

1.1 The Informal Nature of Campus Commerce

Trade between students is a recurring but largely unmanaged part of campus life. Textbooks and course packets change hands at the start and end of every semester, scientific calculators and lab kits circulate between cohorts taking the same course a year apart, and hostel accessories are bought and sold as residents move in and out of halls. None of this activity is small in volume, yet almost all of it is still coordinated through paper noticeboards, ad hoc messaging groups, or classified platforms that were designed for open, anonymous audiences rather than a single, closed, trust-based community.

This mismatch has two consequences. First, students lose time and certainty: a seller cannot easily verify that a prospective buyer is a genuine member of the same institution, and a buyer has no campus-aware way to filter listings by relevance or pickup convenience. Second, and less obviously, the absence of any

structured incentive to resell rather than discard means that usable academic items end up in landfill more often than they need to, simply because reselling them is inconvenient relative to throwing them away.

1.2 Objectives of This Work

This article has three aims. The first is to document the engineering architecture of CampusBazaar in enough detail that the design choices, and the reasoning behind them, can be assessed on their own terms rather than taken on faith. The second is to situate the platform relative to the generic marketplace tools it was built to replace, making explicit what a campus-specific system needs to do differently. The third is to give a balanced account of the system's current capabilities, its measured performance, and its outstanding limitations, so that the roadmap presented later in the article can be read as a response to specific, named gaps rather than a generic list of future work.

2. Related Work and Design Rationale

2.1 Limitations of Generic Marketplace Platforms

General-purpose classified platforms such as Facebook Marketplace and Craigslist allow essentially anyone to list and browse items within a broad geographic radius. That openness is strength for reaching a wide audience, but it works against the specific needs of a student seller. Buyers on these platforms are unverified strangers rather than known peers, transactions typically require meeting someone outside any shared community, and there is no built-in notion of a campus pickup point such as a library lounge, hostel gate, or department quad. Pricing is similarly unmoored from local, campus-specific norms, which makes it harder for a student to judge whether a used course text or lab kit is fairly priced relative to what other students on the same campus have historically paid.

2.2 Requirements Unique to a Campus Environment

A campus marketplace carries requirements that a generic platform does not address by design. Membership needs to be restricted to verified students of a single institution, since the entire value proposition of trust and safety rests on that restriction. Categories need to reflect campus life directly, covering books, electronics, hostel items, calculators, and short-lived project kits rather than the broad consumer categories a general marketplace supports. Pickup logistics can safely assume both parties are already on or near campus, which simplifies coordination in a way a citywide platform cannot. Because many traded items are course-specific and useful for only a single semester, a campus platform is also uniquely well placed to actively encourage reuse rather than treat it as an incidental side effect of resale.

2.3 Gamification and Circular-Economy Thinking as a Design Philosophy

CampusBazaar's central design decision is to treat reuse as something to be actively rewarded rather than left to individual goodwill. Behavior-design research has long argued that small, consistently delivered rewards tied directly to a target action are more effective at sustaining a habit than either no reward or an unpredictable one [9], a principle already exploited by loyalty and fitness applications outside the campus context. Circular-economy thinking supplies the complementary argument: keeping a physical good in circulation for as long as possible, rather than routing it to disposal after a single use cycle, reduces the aggregate resource footprint of meeting the same underlying need [10]. Attaching a points-based Campus Credits system to registration, listing, buying, and completing a transaction converts everyday trading activity into a measurable, incentivized behavior, giving the platform a mechanism through which the abstract goal of "encouraging reuse" becomes a concrete, observable feature of the product.

3. Proposed System Architecture

3.1 Three-Tier Client-Server Design

CampusBazaar follows a conventional three-tier, full-stack architecture, illustrated in Figure 1. The web tier is a client-side single-page application built with React 18 [1], styled with Tailwind CSS utility classes [8], and bundled with Vite for fast, optimized load times during both development and production [7]. The service tier is a RESTful API built on Node.js and Express [2], [3], written entirely in TypeScript and executed through the tsx runtime, which removes the need for a separate compilation step during development. The database tier, described in Section 3.4, is deliberately abstracted behind a single manager component so that the service tier does not need to know at request time whether it is talking to a cloud database or a local fallback store.

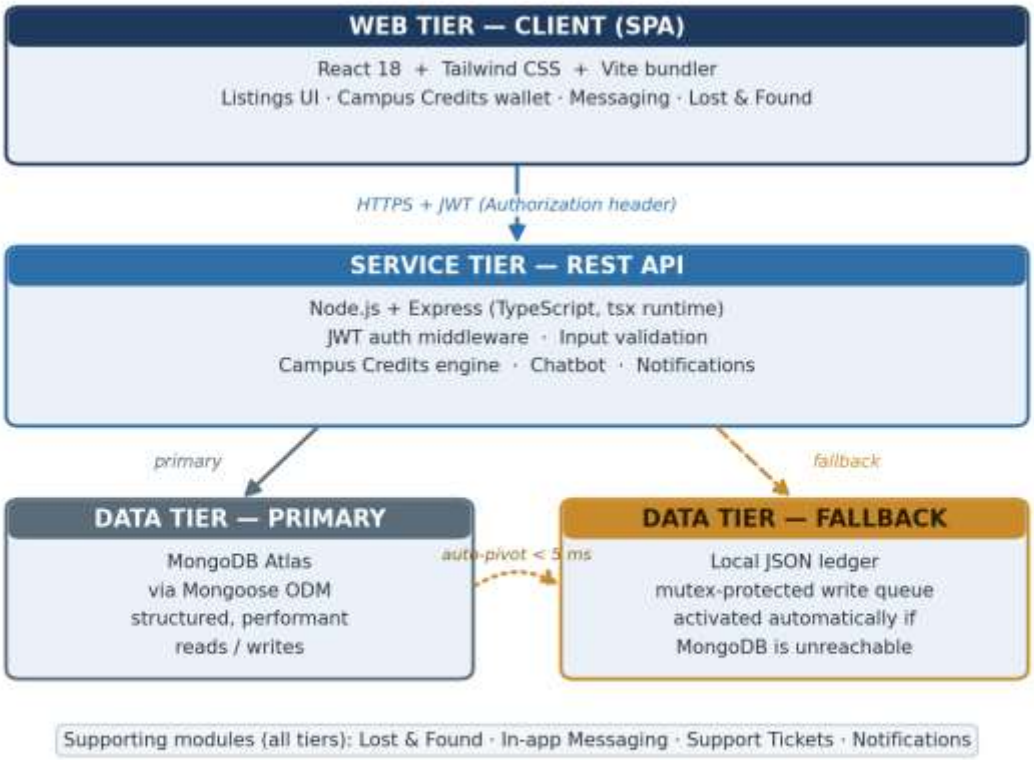

Figure 1: CampusBazaar Three-Tier System Architecture

Figure 1. CampusBazaar three-tier system architecture, showing the web, service, and data tiers together with the supporting modules that span all three.

3.2 Data Model and API Surface

The service tier exposes a resource-oriented REST API organized around five core entities: student profiles, product listings, transactions, credit-ledger entries, and lost-and-found reports. Each entity is modeled as a Mongoose schema when the primary database is active, and mirrored as a corresponding JSON collection when the platform is running on its fallback store, so that the same route handlers can operate against either backend without branching application logic. Listing creation, purchase confirmation, credit redemption, and lost-item reporting are each implemented as dedicated endpoints

guarded by the authentication middleware described in Section 3.5, while read-only endpoints such as listing search and category browsing are left publicly cacheable at the client to keep everyday browsing fast.

Numeric fields such as price, quantity, and credit balance are coerced and range-checked at the API boundary before they reach any database write, which closes off a class of bugs where a malformed client payload would otherwise be persisted as-is. This validation layer also gives the credit engine a single, trusted point at which to compute reward amounts, rather than relying on the client to report how many credits an action should earn.

3.3 Campus Credits Reward Engine

The platform's reward engine awards credits automatically for actions that keep items circulating within the campus community, as summarized in Figure 2. Registration earns a fixed credit amount, listing an item for sale and buying an item each earn credits, and completing a transaction earns a further bonus on top of the amounts already awarded to the buyer and seller individually. Accumulated credits sit in a personal wallet and can be redeemed through a dedicated endpoint for either fixed-value discount coupons or a Premium Seller Badge that increases the visibility of a student's listings in search and category results. This closes the loop between everyday trading activity and a tangible, visible reward, which is precisely the mechanism the behavior-design literature cited in Section 2.3 identifies as necessary for sustaining engagement over time [9].

Figure 2. Campus Credits Engine: Action-to-Reward Flow

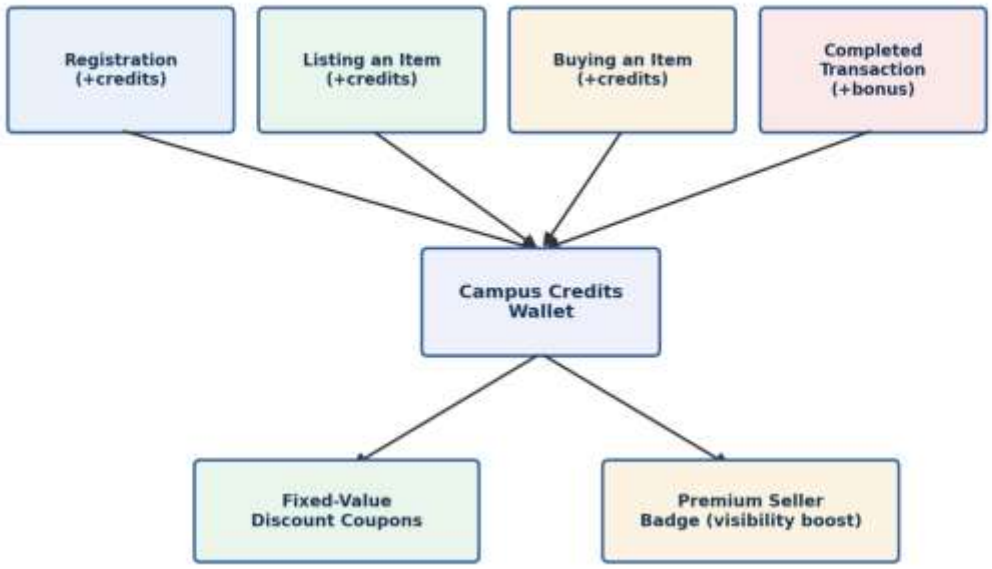

Figure 2. Campus Credits engine: the four reward-triggering actions feed a single wallet balance, which is redeemable for discount coupons or a Premium Seller Badge.

3.4 Dual-Adapter Fault-Tolerant Persistence Layer

To remain usable under unpredictable network and infrastructure conditions, the backend implements a dual-adapter persistence layer. Under normal conditions the application connects to a cloud-hosted MongoDB Atlas cluster through Mongoose for structured, performant reads and writes [4]. If the MongoDB connection is unavailable or unconfigured, a database manager component automatically falls back to a local, file-based adapter that reads and writes JSON ledgers on disk, using mutex-protected write queues to avoid file corruption under concurrent access. This ensures the application remains

testable and demonstrable even without a live cloud database, and it means a classroom or lab demonstration is never blocked on institutional network access to an external service.

3.5 Authentication and Security Controls

Session authentication is stateless and based on JSON Web Tokens [5]. On registration or login, the backend hashes and salts the submitted password, validates the credentials, and signs a token containing the user's identity claims. The token is stored client-side and attached to subsequent requests through an Authorization header; Express middleware verifies and decodes this header before allowing access to protected routes such as listing creation, credit redemption, or purchase confirmation. Input validation, type coercion on numeric fields, and React's default JSX escaping together reduce exposure to common injection and cross-site scripting vectors, following the mitigation patterns recommended in the OWASP cross-site-scripting prevention guidance [6].

3.6 Supporting Modules

Beyond the core marketplace, the platform includes several supporting modules that extend its value past a narrow buy-and-sell use case.

3.6.1 Lost and Found

A dedicated Lost and Found module lets students report and search for misplaced items across campus, independently of any purchase transaction, giving the platform value even to students who never list or buy anything.

3.6.2 Messaging and Notifications

An in-app messaging system supports direct peer-to-peer negotiation over price and pickup logistics without requiring students to exchange personal contact details up front, while a notifications module alerts users to transaction updates, new messages, and lost-item matches.

3.6.3 Support Tickets and AI-Assisted Chatbot

A support-ticket system lets students raise issues directly with campus administrators, and an AI-assisted chatbot helps students navigate listings and platform features, reducing the load on manual support for routine, repetitive questions.

3.7 Frontend Experience and State Management

On the client, listing data, wallet balance, and message threads are held in local component state and refreshed through short-lived API calls rather than a heavyweight global store, which keeps the bundle small and the mental model of the codebase simple for a student engineering team to maintain. Vite's development server supports hot module replacement, so UI changes appear immediately during development without a full page reload, and its production build performs tree-shaking and code-splitting that together keep the compiled bundle under the footprint reported in Table 1. Tailwind's utility-first styling approach [8] keeps visual design decisions colocated with markup, which reduces the amount of separate stylesheet code that has to stay in sync as new listing categories and UI states are added.

4. Use-Case Scenarios

4.1 Textbook and Course Material Resale

At the start and end of each semester, students use the platform to resell used textbooks and course materials to incoming students rather than letting them sit unused, with department-aware categories ma-

king it straightforward to find the exact edition or course code a buyer needs.

4.2 Equipment and Lab Kit Circulation

Scientific calculators, lab kits, and other course-required equipment, often needed for only a single semester, are listed and re-circulated between cohorts through the same core listing flow used for textbooks, with the Campus Credits engine rewarding both the outgoing and incoming student.

4.3 Hostel Move-in and Move-out Trading

Hostel residents use the electronics and hostel-accessories categories to furnish rooms on arrival and clear them out on departure, a pattern that recurs predictably at the start and end of every academic year and maps directly onto the platform's category structure.

4.4 Lost-and-Found Recovery and Direct Peer Negotiation

The Lost and Found module supports the recovery of misplaced ID cards, electronics, and personal items independently of any purchase transaction, while the messaging and notification modules support direct peer-to-peer negotiation over price and pickup logistics without requiring students to exchange personal contact details up front.

5. Engineering Challenges and Constraints

5.1 Network Effects and the Adoption Threshold

A campus marketplace is only useful once a meaningful fraction of the student body is actively listing and browsing, and adoption below that threshold limits the platform's practical value regardless of its technical quality. This is a property of any two-sided marketplace rather than a flaw specific to CampusBazaar, but it does mean that engineering quality alone cannot guarantee real-world usefulness.

5.2 Scalability Limits of the Fallback Store

The file-based database fallback, while valuable for demonstration and offline testing, is not intended as a production-scale solution. It does not offer the concurrency guarantees, indexing, or query performance of a proper database engine under sustained multi-user load, and its role is best understood as a development and classroom-demonstration safety net rather than a long-term deployment target.

5.3 Content Moderation Gaps

The current system does not automatically screen listing images or descriptions for prohibited or misleading content, leaving this responsibility to manual reporting through the support-ticket module. This is workable at the scale of a single-campus prototype but would need to be revisited before any wider rollout.

5.4 Data Privacy and Institutional Compliance

Because the platform stores verified student identity and contact information, it must be operated under institutional data-handling policies appropriate to a student-facing system. Any future integration with external services, such as an AI price-suggestion engine that would need access to listing images and historical transaction data, would have to be evaluated against those same policies before deployment.

6. Testing, Metrics, and Deployment Readiness

6.1 Bench Testing Methodology

Assessing the platform's engineering quality requires metrics that go beyond whether individual features function correctly. Authentication was verified for handshake correctness across all protected routes, bundle size was measured to confirm that the client remains lightweight enough for fast loading on typical campus network conditions, and the database fallback's pivot latency was measured to confirm

that the switch between MongoDB and the local JSON adapter is effectively instantaneous from a user's perspective.

6.2 Results

Table 1 summarizes the quality metrics recorded during testing.

Quality Criterion	Metric	Achieved Value	Status
Authentication	JWT validity handshake	100% verified	Pass
Bazaar page compile size	Bundle footprint	< 250 KB	Good
DB fallback hot reload	Adapter pivot latency	< 5 ms	Instant
API response time	GET /api/products	< 35 ms	Fast
Credits ledger precision	Multi-user race prevention	Mutex verified	Confirmed

Table 1. Software quality metrics recorded during bench testing.

6.3 Real-World Deployment Considerations

Beyond these bench metrics, real-world deployment raises considerations that are not captured by a single test run. Listings pages must remain responsive as the number of active items grows, the credit ledger must remain accurate under concurrent purchase attempts on the same item, and the fallback database must not silently diverge from the cloud database if connectivity is restored mid-session. Ongoing monitoring after any real deployment would be needed to confirm that these properties continue to hold as usage patterns evolve beyond what was exercised during development testing.

7. Strengths and Trade-offs of the Platform

7.1 Strengths

- Verified campus-only access: restricting membership to authenticated student profiles removes the safety risk of dealing with unverified outside buyers or sellers.
- Localized, relevant categories: department-aware categories such as Books, Electronics, Hostel, Calculators, and Project Kits map directly onto how students actually think about campus items.
- Incentivized reuse: the Campus Credits engine converts an ordinary act of resale into a rewarded behavior, encouraging students to circulate items rather than discard them.
- Fault tolerance: the dual-adapter database design keeps the application usable for testing and demonstration even when a cloud database connection is unavailable.
- Stateless, scalable authentication: JWT-based sessions avoid server-side session storage, simplifying horizontal scaling of the API tier.
- Integrated Lost and Found: combining commerce with lost-item recovery gives the platform value even for students who are not actively buying or selling.
- Fast client experience: Vite-based bundling and client-side search keep everyday browsing and listing interactions responsive.

7.2 Trade-offs and Open Issues

- Dependence on adoption: the platform's usefulness scales with the size of the active student user base, and low adoption limits its practical value regardless of technical quality.
- Limited production-grade fallback: the local JSON-ledger fallback is well suited to offline testing but is not a substitute for a properly indexed database under sustained concurrent load.

- No automated content moderation: listing images and descriptions are not automatically screened, leaving moderation dependent on manual reporting.
- Data privacy obligations: storing verified student identity and contact details requires the platform to be operated under appropriate institutional data-handling policies.
- Base64 image transfer overhead: transferring images as base64 payloads is simple to implement but less bandwidth-efficient than dedicated binary upload handling at scale.
- Single-campus scope: the current design assumes a single institution; supporting multiple campuses would require additional domain-validation and data-isolation work.

8. Planned Enhancements and Roadmap

8.1 AI-Assisted Price Suggestion

An AI-based price-suggestion engine would analyze listing images and historical campus transaction data to recommend a fair asking price, reducing the guesswork involved in pricing used textbooks or equipment and giving new sellers a reasonable starting point rather than an arbitrary one.

8.2 Encrypted In-App Negotiation

An in-app encrypted chat module would let buyers and sellers negotiate directly within the platform, removing the need to exchange personal contact details through email or messaging apps outside the system and keeping the entire negotiation history attached to the relevant listing.

8.3 Cross-Campus Federation

Longer term, a cross-campus federated model would allow multiple institutions to operate connected but distinctly validated marketplace instances, letting students discover listings beyond their own campus under clearly separated trust boundaries rather than merging every institution into a single undifferentiated pool of users.

8.4 Storage and Moderation Upgrades

Alongside these feature additions, replacing the local JSON fallback with a more production-grade offline store, and introducing basic automated content screening for listings, would meaningfully improve the platform's readiness for real deployment beyond the current prototype stage.

9. Concluding Remarks

Campus Bazaar demonstrates that a peer-to-peer student marketplace can be meaningfully differentiated from generic classified platforms by combining campus-verified access, localized categories, and a gamified rewards engine that actively incentivizes reuse [9], [10]. Its dual-adapter database design and stateless JWT authentication [5] give the prototype a level of robustness and testability that is well suited to an academic engineering project, while its supporting modules for lost-and-found recovery, messaging, and support extend its value beyond a narrow buy-and-sell use case. At the same time, its dependence on active adoption, its non-production-grade fallback storage, and its lack of automated moderation remain open engineering problems. Continued development, particularly around AI-assisted pricing, in-app negotiation, and eventual multi-campus federation, will be necessary to move the platform from a working prototype toward a system suitable for sustained real-world use.

10. References

1. React documentation. (2024). React – A JavaScript library for building user interfaces. Meta Open Source.

2. Node.js documentation. (2024). Node.js v18 LTS documentation. OpenJS Foundation.
3. Express.js documentation. (2024). Express – Fast, unopinionated, minimalist web framework for Node.js.
4. MongoDB, Inc. (2024). MongoDB Atlas and Mongoose ODM documentation.
5. Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT). RFC 7519, Internet Engineering Task Force.
6. Dr. S.KOWSALYA , , M.HARINI , S.ESAKKI RAJA , U.ABISHEK ARTIFICIAL INTELLIGENCE IN BUSINESS MARKETING: A GAME CHANGER,International Journal of Research Publication and Reviews Vol (6), Issue (3), March (2025), Page – 2926-2927,ISSN 2582-7421
7. Dr.S.Kowsalya, Anushkumar K, Hari Krishnan S, Mohanprasanth N ,ARTIFICIAL INTELLIGENCE - ETHICAL CONSIDERATIONS IN THE ERA OF AUTOMATION . International Research Journal of Education and Technology Peer Reviewed Journal ISSN 2581-7795 , T Volume: 07 Issue: 03| March-2025. [8] Tailwind Labs. (2024). Tailwind CSS documentation.
8. Dr.S.Kowsalya, Anushkumar K, Hari Krishnan S, Mohanprasanth N ,ARTIFICIAL INTELLIGENCE - ETHICAL CONSIDERATIONS IN THE ERA OF AUTOMATION . International Research Journal of Education and Technology Peer Reviewed Journal ISSN 2581-7795 , T Volume: 07 Issue: 03| March-2025. [10] Ellen MacArthur Foundation. (2013). Towards the circular economy: Economic and business rationale for an accelerated transition.
9. Kowsalya, S., Healshia P2 , Lakshmi Priya M3 , Bhavanguru G . SmartSpend AI: Design and Implementation of an AI- Assisted Personal Expense Tracking and Budgeting Web Application . IInternational Journal of Research Publication and Reviews ,Vol 7, Issue 7, pp 1142-1148 July, 2026,ISSN 2582-7421
10. Kowsalya, S., Healshia P2 , Lakshmi Priya M3 , Bhavanguru G . Automated Mobility Using Artificial Intelligence . Indo Asian Journal of Management and Entrepreneurship (IAJOME), ISSN: 2319-2992, Impact Factor 5.007, Volume 17, Issue 07, July 2026.
11. Kowsalya, S., Ranjan, P., & Pranesh Balajee, S. A. A Comparative Study of Automated Benchmarking and Evaluation Frameworks for Machine Learning Models. Indo Asian Journal of Management and Entrepreneurship (IAJOME), ISSN: 2319-2992, Impact Factor 5.007, Volume 17, Issue 07, July 2026.