International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

CampusShare: A Web-Based College Resource Sharing Platform for Buying, Renting and Borrowing Academic Resources

Priyanka S[#1], Rekha M[#2], Shamaa M[#3], Vinoth Kumar S[*4]

[#]Department of Computer Science and Engineering, Adhiyamaan College of Engineering (Autonomous), Hosur, Tamil Nadu, India

***Abstract*— Students need textbooks, calculators, laboratory equipment and project components for short periods, yet buying every item new is costly while many useful items lie unused once a semester ends. Informal exchange through friends, social-media groups and notice boards is slow, unstructured and untraceable. This paper presents CampusShare, a web-based College Resource Sharing Platform that lets students list, search, buy, rent and borrow academic resources within a college community. The application uses HTML, CSS, JavaScript and Bootstrap on the front end, Python Flask on the back end and SQLite for storage. Its modules cover registration and authentication with hashed passwords, resource listing with multiple images or videos, keyword and category search, buy and rental request management with owner approval, and handover verification through a unique transaction ID and QR code. A working implementation is demonstrated through its user-interface screens. The platform reduces the cost of academic materials, promotes reuse, and replaces scattered messaging with a single, organised and traceable process.**

***Keywords*— college resource sharing, academic resource marketplace, buy rent and borrow, peer-to-peer sharing, Flask, SQLite, QR code handover, resource reuse**

I. Introduction

Students in engineering and other programmes depend on a wide range of academic resources such as textbooks, scientific calculators, laboratory equipment, reference material and electronic project components. Many of these items are needed only for a semester, a laboratory course or a single project, but they are priced as permanent purchases. Once the requirement ends, the same items often remain idle with their owners. The result is avoidable expense for one group of students and under-used resources for another.

Within a campus, the usual way of bridging this gap is word of mouth, messaging groups and notice boards. These channels have no common catalogue, give no reliable view of availability or condition, and keep no record of who gave what to whom. Students may also hesitate to lend items without a clear way to arrange a handover or to confirm a return.

Online sharing platforms have shown that matching people who own idle assets with people who need them can lower costs and extend the useful life of goods [1]-[4]. General marketplaces, however, are not built around an academic community, and they rarely distinguish between buying, renting and borrowing within one workflow. This paper presents CampusShare, a college-specific platform that does so. The main contributions are:

- a unified web application in which students list resources and choose among buying, renting and borrowing;
- an owner-approval request workflow with status tracking and contact disclosure only after acceptance;
- a transaction ID and QR code generated on accepted rentals to support verified physical handover; and
- a lightweight Flask and SQLite implementation that can be deployed on modest college infrastructure.

II. Literature Review

Research on the sharing economy describes platforms as intermediaries that connect providers of under-utilised assets with users who need temporary access [5]. Belk argues that access, not ownership, is becoming central to online consumption [2], and Hamari et al. identify sustainability, enjoyment and economic benefit as principal motivations for participating in collaborative consumption [3]. Benjaafar et al. model peer-to-peer product sharing and show that, under suitable conditions, access to shared goods can substitute for individual ownership and increase social welfare [6].

The environmental case for reuse is supported by Dhanorkar, whose study of the staggered introduction of Craigslist across US markets associates online consumer-to-consumer trading with a reduction in municipal solid waste [7]. Platform coordination has been examined by Zeng et al., who analyse how sharing-economy firms orchestrate resources from many providers [8]. Trust is a recurring concern: Ert et al. show that visual information about hosts influences perceived trustworthiness on a peer-to-peer platform [9], and Möhlmann reports that satisfaction and trust affect whether users return to a sharing option [10].

These studies establish that centralised digital platforms can improve access and reduce waste, but they address general consumer goods or accommodation. They do not describe a single campus application that combines buy, rent and borrow options, owner-controlled requests and verified handover for academic materials. CampusShare is designed to fill that gap.

III. System Analysis

A. Limitations of Existing Methods

Current practice relies on personal contacts, social-media groups and notice boards. Five limitations follow. First, there is no centralised place to discover resources, so students search several channels. Second, the choice between purchase, rental and borrowing is made ad hoc, and students may buy items they need only briefly. Third, there is no catalogue showing condition, price and availability. Fourth, requests and arrangements are scattered over multiple messaging tools, which leads to missed messages and confusion. Fifth, there is no transaction history or tracking of rented and borrowed items.

B. Proposed System

CampusShare brings listing, searching, requesting and transaction management into one platform. Students register and log in, list resources with name, category, description, price, images and sharing type, and browse or search the listings of others. A student submits a buy or rental request; the owner reviews it and accepts it. On acceptance the requester can see the owner's contact details to arrange a

handover on campus, and, for rentals, a transaction ID and QR code are generated for check-in. The principal advantages are centralised discovery, lower cost through second-hand and rental options, a structured request process, and a traceable record of exchanges.

IV. System Architecture and Modules

A. Architecture

CampusShare follows a client-server architecture, shown in Fig. 1. The browser-based front end presents the interface, the Flask back end applies business logic and the SQLite database stores users, resources, media and requests. A student first authenticates, then moves through resource listing, search and filter, the buy, rent or borrow choice, and request handling. The overall process produces three outputs: resource exchange details for accepted requests, transaction and handover information, and database updates to user records, availability and history.

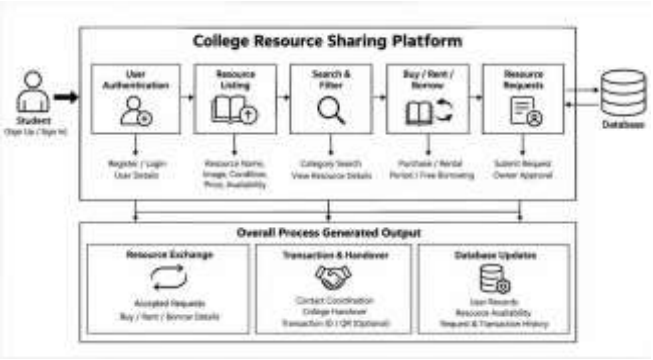


Fig. 1 Architecture of the College Resource Sharing Platform, showing the five functional components and the outputs generated from the overall process.

B. Modules

1) User registration and authentication: Students register with name, college register number, e-mail and password. Passwords are stored as salted hashes, credentials are verified at login and a session identifies the user afterwards.

2) Resource listing and management: An owner adds a resource with name, category, description, sharing type, buy price and rent price, and can attach several photos or videos. Owners can view and delete their own listings.

3) Search and filter: Students search by keyword across name, category and description and narrow results with category filters such as textbooks, calculators, notes, lab equipment and electronics.

4) Buy, rent and borrow: The listing states which options are offered. A buyer purchases a second-hand item, a renter uses it for an agreed period, and a borrower receives it free of charge where the owner allows.

5) Request management: Requesters send requests to owners. The owner accepts or rejects them, and the status is shown to the requester under My Requests.

6) Transaction and handover: After acceptance, the students coordinate through the disclosed contact details. A unique transaction ID and QR code support check-in at the physical handover of a rented item.

7) Database management: SQLite stores all application data and is queried whenever users log in, add resources, search or submit requests.

V. Database Design

The database maintains the relationships between users, resources and requests. One user can list many resources, and one resource can receive many requests. Primary keys identify each record, and foreign keys link a resource to its owner, a media file to its resource, and a request to both the requested resource and the requesting user. Table I summarises the main entities.

TABLE I
Main Database Entities

Entity	Principal attributes
User	user id, name, register number, e-mail, mobile number, bio, password hash
Resource	resource id, owner id, name, category, description, share type, buy price, rent price
ResourceMedia	media id, resource id, media path, media type
Request	request id, resource id, requester id, request type, status, transaction ID

VI. Implementation

A. Technology Stack

The platform is implemented with the tools listed in Table II. Bootstrap provides responsive layouts, Flask handles routing and form processing, and SQLite offers a self-contained relational store suitable for a small campus deployment [11], [12].

TABLE II
Software Used for Development

Component	Technology
Operating system	Windows 10/11 or equivalent
Front end	HTML, CSS, JavaScript
CSS framework	Bootstrap
Back end	Python, Flask
Database	SQLite
Editor and VCS	Visual Studio Code, Git, GitHub
Browser	Google Chrome or equivalent

B. Authentication

The login route reads the e-mail and password, retrieves the matching user and compares the supplied password with the stored hash using a password-hash check. On success the user identifier is placed in the session and the user is redirected to the home page. Routes that modify data, such as adding a resource or sending a request, redirect unauthenticated users to the login page.

C. Resource Listing and Media Upload

The add-resource route collects the form fields, allows one or more sharing types to be selected, and stores the record. Uploaded images and videos are read from the multi-file field, file names are sanitised with a secure-filename routine, the files are saved in a static uploads directory and each is recorded in the media table with its path and content type. The same listing can therefore carry several photos or a video.

D. Search and Browse

The browse route builds a query over the resource table. A keyword is matched case-insensitively against name, category and description, an optional category filter is applied with an exact match, and results are ordered with the newest listings

first. The result page shows image, category, title, description, available options and price for each item.

E. Request Handling and QR Verification

A signed-in student can send a buy or rental request for a listing. The request is stored with its requester, resource and status. When the owner accepts a rental, the system generates a transaction ID (for example, CS-RENT-5DD60E) and a QR code that the requester can display at handover. The owner's status view distinguishes accepted from checked-in rentals.

VII. Results and Discussion

A. Functional Results

The implemented system was run locally and exercised through its user interface. The home page (Fig. 2) offers browsing, sharing and a search bar. The browse page (Fig. 3) lists resources with category chips and a search box, and the add-resource form (Fig. 4) captures name, category, description, buy and rent prices and media. Listings seen in testing include a Python programming book, an ESP32 development board and an ultrasonic distance sensor, which are typical of the textbook and electronics categories targeted by the platform.

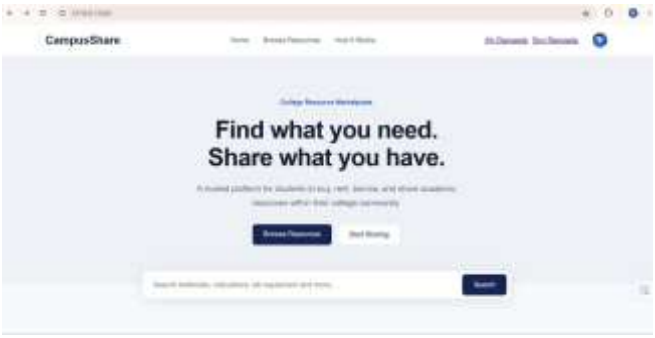


Fig. 2 Home page of the CampusShare platform.



Fig. 3 Browse page with keyword search and category filters.



Fig. 4 Add-resource form for listing a resource.

A resource detail page (Fig. 5) shows the description, available option, price and owner, with a Buy Now action. After a request is accepted, the requester sees the status and the seller's contact details under My Requests, and an accepted rental shows a unique transaction ID and QR code (Fig. 6). These screens confirm that the end-to-end flow of listing, discovery, request, acceptance and handover verification is working.



Fig. 5 Resource detail page with the Buy Now option.



Fig. 6 Accepted rental showing the QR code and transaction ID (owner contact details cropped).

B. Comparison with Informal Sharing

Table III compares the platform with the informal methods it replaces. The comparison is qualitative and is based on the features implemented; no user study or performance benchmark has yet been conducted.

TABLE III
Informal Sharing versus CampusShare

Aspect	Informal methods	CampusShare
Discovery	Multiple channels	Single searchable catalogue
Options	Ad hoc	Buy and rent; borrow as sharing type
Requests	Scattered messages	Owner accept with status
Handover	Unverified	Transaction ID and QR
Records	None	Request and rental status stored

C. Limitations

The current build handles buy and rental flows in full. Ratings, trust scores, damage reporting, reward points, return-date reminders and a complete transaction history are not yet implemented. Photos and contact details are stored on the local server, and the prototype has been tested with a small number of sample accounts rather than a campus-scale load.

VIII. Conclusion and Future Scope

This paper presented CampusShare, a Flask and SQLite web application that lets students buy, rent and borrow academic resources within a college. The implementation provides secure registration, multi-media resource listing, keyword and category search, owner-approved requests and QR-based handover verification. It replaces scattered informal

exchange with one organised process and encourages reuse of materials that would otherwise sit idle.

Planned work includes ratings and reviews, trust scores, damage reporting, a full transaction history, e-mail and in-app notifications for requests and return dates, QR confirmation of returns, and a usability study with students to measure adoption and savings.

ACKNOWLEDGMENT

The authors thank the Head of the Department of Computer Science and Engineering and the faculty of Adhiyamaan College of Engineering, Hosur, for their guidance and support during this work.

REFERENCES

[1] K. Frenken and J. Schor, "Putting the sharing economy into perspective," Environmental Innovation and Societal Transitions, vol. 23, pp. 3–10, 2017.

[2] R. Belk, "You are what you can access: Sharing and collaborative consumption online," Journal of Business Research, vol. 67, no. 8, pp. 1595–1600, 2014.

[3] J. Hamari, M. Sjöklint, and A. Ukkonen, "The sharing economy: Why people participate in collaborative consumption," Journal of the Association for Information Science and Technology, vol. 67, no. 9, pp. 2047–2059, 2016.

[4] A. Geissinger, C. Laurell, C. Öberg, and C. Sandström, "How sustainable is the sharing economy? On the sustainability connotations of sharing economy platforms," Journal of Cleaner Production, vol. 206, pp. 419–429, 2019.

[5] J. Wirtz, K. K. F. So, M. A. Mody, S. Q. Liu, and H. H. Chun, "Platforms in the peer-to-peer sharing economy," Journal of Service Management, vol. 30, no. 4, pp. 452–483, 2019.

[6] S. Benjaafar, G. Kong, X. Li, and C. Courcoubetis, "Peer-to-peer product sharing: Implications for ownership, usage, and social welfare in the sharing economy," Management Science, vol. 65, no. 2, pp. 477–493, 2019.

[7] S. Dhanorkar, "Environmental benefits of internet-enabled C2C closed-loop supply chains: A quasi-experimental study of Craigslist," Management Science, vol. 65, no. 2, pp. 660–680, 2019.

[8] J. Zeng, M. M. Tavalaei, and Z. Khan, "Sharing economy platform firms and their resource orchestration approaches," Journal of Business Research, vol. 136, pp. 451–465, 2021.

[9] E. Ert, A. Fleischer, and N. Magen, "Trust and reputation in the sharing economy: The role of personal photos in Airbnb," Tourism Management, vol. 55, pp. 62–73, 2016.

[10] M. Möhlmann, "Collaborative consumption: Determinants of satisfaction and the likelihood of using a sharing economy option again," Journal of Consumer Behaviour, vol. 14, no. 3, pp. 193–207, 2015.

[11] Pallets Projects, "Flask documentation." [Online]. Available: https://flask.palletsprojects.com/

[12] SQLite Consortium, "SQLite documentation." [Online]. Available: https://www.sqlite.org/docs.html