



Design, Implementation and Functional Validation of Closed-Community Digital Carpooling and Mobility Transaction Platform for Staff of Federal Polytechnic Ede, Osun State

Etudaiye A.I.^a, Fabiyi A.O.^a, Lawal M.O.^a, Ateko B.C.^b and Abdul-Ganiyu T. K.^{a,*}

^aDepartment of Cyber Security, Federal Polytechnic Ede, Osun State, Nigeria

^bDepartment of Artificial Intelligence, Federal Polytechnic Ede, Osun State, Nigeria

*Corresponding author: abdulganiyutoyeeb@gmail.com

Abstract – The study developed and functionally validated a closed-community digital carpooling platform for staff of Federal Polytechnic Ede, Nigeria. The platform addressed coordination, trust, payment, and accountability requirements associated with informal staff ride-sharing by integrating staff registration and authentication, ride publication and discovery, booking controls, fare presentation, online payment, server-side payment verification, driver approval or rejection, electronic notification, transaction logging, earnings visibility, and withdrawal requests. A three-tier architecture was implemented using HTML5, CSS3, JavaScript and Bootstrap 5.3 at the presentation layer, PHP at the application layer, and MySQL at the data layer, with Paystack providing payment processing and an e-mail service supporting notifications. Twelve black-box functional workflows were executed using test accounts and Paystack test transactions. All twelve workflows produced the specified outcomes, and the payment reference generated during testing corresponded with the reference contained in the driver's notification. The study demonstrated the functional feasibility of treating institutional carpooling as a closed-community transaction lifecycle rather than only as a ride-matching service. However, the evaluation did not measure user adoption, usability, response time, scalability, financial savings, congestion, or emissions. The results therefore provide implementation evidence for the platform's transaction workflow and identify constraint-aware matching, usability evaluation, security assessment, and real-world mobility evaluation as priorities for subsequent deployment.

Keywords: carpooling; digital platform; institutional mobility; ride-sharing; transaction system

1. Introduction

Daily commuting has become an increasingly consequential household expense in Nigeria. Following the removal of the petrol subsidy in 2023, fuel prices rose substantially, increasing the cost of routine travel for employees who depend on private vehicles. Recent evidence from Nigeria has linked financial pressure associated with subsidy removal to carpooling behaviour and has identified digital platforms as a mechanism through which informal ride-sharing can be organised (Maiguwa et al., 2026; Onikosi-Alliyu, 2025). The economic incentive is therefore increasingly visible; the technological challenge is to turn occasional informal sharing into a dependable institutional service.

Carpooling is not simply a transportation matching problem. A successful shared ride requires at least five interdependent conditions: participants must be eligible to use the service; their travel opportunities must be sufficiently compatible; the financial commitment must be understood; the transaction must be recorded reliably; and participants must have an accountable mechanism for resolving the booking. Classical ridesharing research has concentrated heavily on matching and optimisation (Agatz et al., 2012; Furuhashi et al., 2013), while institutional applications have demonstrated the value of restricting participation to defined communities (Bruglieri et al., 2011; Correia & Viegas, 2011).

Workplace and education-sector settings provide a distinctive opportunity because users share an organisational identity and often exhibit recurring commuting patterns. A closed community can reduce uncertainty about the identity of other participants, a factor associated with trust in carpooling participation (Correia & Viegas, 2011; Javid et al., 2025). At the same time, institutional systems introduce requirements that open ride-sharing platforms do not necessarily prioritise, including organisational eligibility, accountability, payment traceability, and privacy.

Federal Polytechnic Ede provides a relevant setting for examining this problem. Staff members commonly travel to campus using private vehicles, while informal ride-sharing arrangements can be difficult to coordinate and provide limited assurance regarding trust, payment, and accountability. The opportunity is therefore not merely to place a list of available rides online, but to construct a controlled digital transaction linking staff identity, ride publication, booking, payment, approval, notification, and settlement.

The study introduced a closed-community institutional mobility architecture in which a ride was treated as a stateful transaction rather than an isolated listing. The design integrated staff-only access, ride publication, booking constraints, pre-trip fare presentation, external payment processing, server-side payment verification, driver authorisation, transaction records, notifications, and payout requests. The significance of the study lies in demonstrating how these controls can be combined into a single institutional workflow that provides an implementation foundation for subsequent matching and mobility analytics.

The objectives were to (i) derive functional and non-functional requirements for a staff-only institutional carpooling service; (ii) design a closed-community architecture integrating identity, ride, booking, payment, and settlement workflows; (iii) implement the architecture using widely accessible web technologies and an external payment gateway; (iv) evaluate the principal user journeys through black-box functional testing; and (v) identify architectural limitations and next-stage requirements for scalable institutional ride matching.

2. Materials and Methods

2.1. Development and artifact-oriented approach

The system was developed as an artifact-oriented institutional information system. Requirements were derived from three complementary sources: (i) the ridesharing literature, (ii) interviews with staff of Federal Polytechnic Ede, and (iii) review of existing carpooling platforms. The development process followed requirements analysis, architecture and data modelling (Elmasri & Navathe, 2016; Sommerville, 2016), implementation, integration, and functional testing. The system was decomposed into registration, authentication, user, ride, booking/payment, notification, and settlement functions so that individual workflows could be implemented and verified independently.

2.2. Functional and non-functional requirements

Table 1. Functional requirements (FR) and non-functional requirements (NFR) of the institutional carpooling platform.

ID	Requirement	Implementation intent
FR1	Staff registration	Staff ID, institutional e-mail, name, phone, department and password
FR2	Authentication	Credential validation and authenticated access
FR3	Ride publication	Route, date, time, seats, car details and notes
FR4	Ride discovery	Ride cards containing route, time, driver, seats and fare
FR5	Booking guards	Prevent self-booking and duplicate booking
FR6	Payment	Paystack payment followed by server-side verification
FR7	Booking decision	Driver approval or rejection of pending bookings
FR8	Notification	E-mail notification for booking and decision events
FR9	Earnings/payout	Driver transaction visibility and withdrawal request
FR10	Administration	Administrative accounts for oversight
NFR1	Access control	Restricted access for registered institutional staff
NFR2	Data integrity	Relational keys and persistent transaction records
NFR3	Usability	Consistent navigation and mobile-adaptive interface

2.3. System architecture

The application adopted a three-tier architecture. The presentation tier used HyperText Markup Language 5 (HTML5), Cascading Style Sheets 3 (CSS3), JavaScript, and Bootstrap 5.3. The application tier was implemented in Hypertext Preprocessor (PHP) and contained the registration, authentication, user, ride, and payment/settlement modules. The data tier used MySQL. Paystack (n.d.-a, n.d.-b) operated as an external payment service and an e-mail service provided notifications. Development and testing were performed on a hosting server because the integrated external services required an accessible deployment environment.

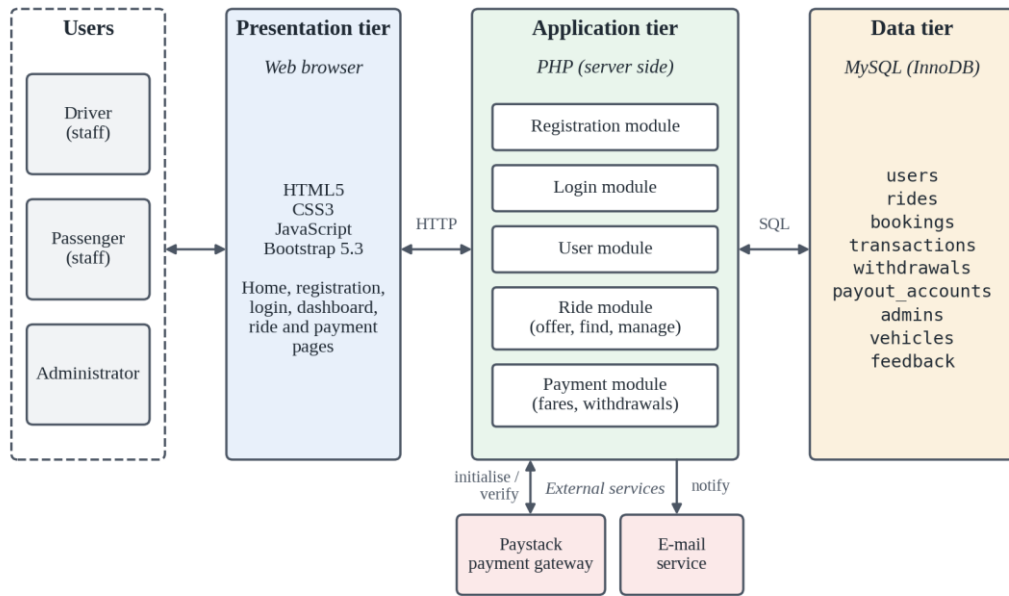


Fig. 1. Three-tier architecture of the carpooling system, with the Paystack payment gateway and the e-mail service as external services.

The architectural separation isolated presentation concerns from application logic and persistent data. More importantly, the payment boundary was explicitly separated from the application: payment credentials were handled by the external gateway while the application received and verified transaction references before creating a booking.

2.4. Closed-community transaction architecture

The core design was a stateful transaction lifecycle. A ride began as an offer, became discoverable, received a payment-backed booking request, was subject to driver approval, and then contributed to a traceable earning record. The architecture therefore connected five control planes: identity, availability, financial commitment, authorisation, and settlement.

Table 2. Control planes and integrity objectives of the transaction architecture.

Control plane	Primary mechanism	Integrity objective
Identity	Staff registration and institutional credentials	Only eligible users participate
Availability	Ride publication, seat count and route/time fields	Ride capacity and information remain explicit
Financial commitment	Fare displayed before payment	Passenger sees the amount before committing
Authorisation	Driver approval/rejection	Booking is not treated as complete merely because payment occurred
Settlement	Transaction record and withdrawal request	Financial events remain traceable to users and rides

2.5. Data model

The database contained nine tables: users, rides, bookings, transactions, withdrawals, payout_accounts, admins, vehicles, and feedback. The four core workflow tables—users, rides, bookings, and transactions formed the primary transaction path. A ride belonged to its publishing user; a booking connected a rider to a ride; and a transaction recorded a payment or payout event associated with a user.

Table 3. Database tables and their purposes.

Table	Purpose
users	Registered staff identity and contact information
rides	Published ride offers
bookings	Passenger requests and booking states
transactions	Payment and payout records with references
withdrawals	Driver withdrawal requests
payout_accounts	Driver payout details
admins	Administrative accounts
vehicles	Vehicle information
feedback	Future feedback and reputation information

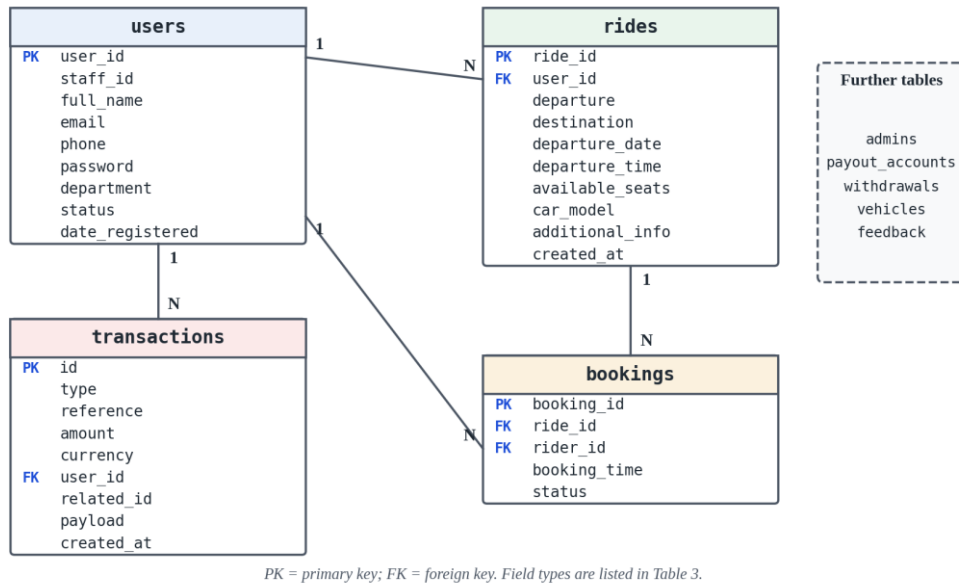
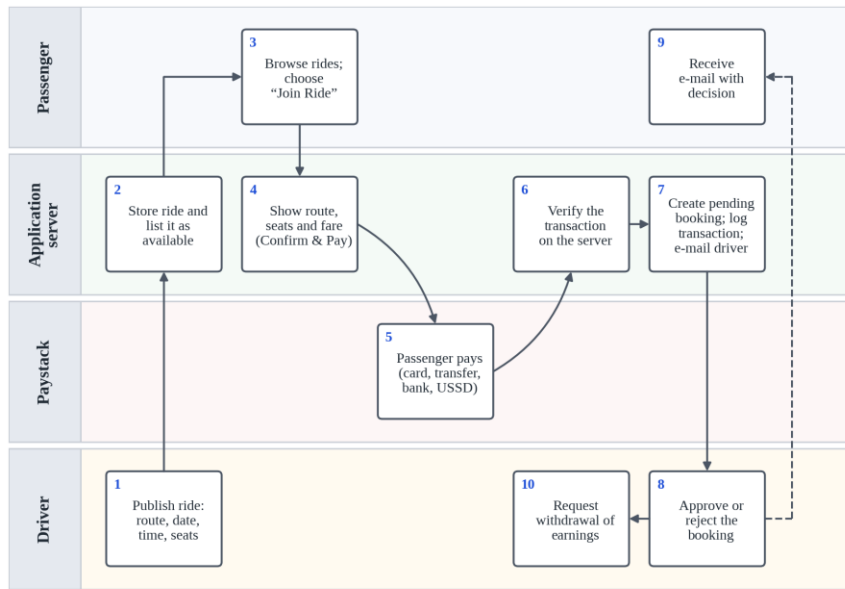


Fig. 2. Data model of the four core tables; the remaining tables are shown at the right.

2.6. Core transaction workflow

The transaction began when a driver published a ride with route, date, time, available seats, and fare. A passenger browsed available rides and selected a compatible offer. The system displayed a confirmation page containing the route, date, seats, and fare before payment. The passenger then completed payment through Paystack (n.d.-b).

Payment completion alone did not create the booking. The application received the transaction reference and verified the payment from the server using the gateway before creating a pending booking. The transaction was then logged and the driver was notified. The driver could approve or reject the request. The passenger received an e-mail containing the decision. Paid bookings contributed to the driver's earnings record, and the driver could submit a withdrawal request. Two explicit booking constraints were enforced: a driver could not join their own ride, and a passenger could not join the same ride more than once.



Solid arrows show the order of events; the dashed arrow is the e-mail sent once the driver decides.

Fig. 3. Sequence of events for one shared ride, from publication to withdrawal of earnings.

2.7. Payment integrity

The payment architecture followed a verify-before-commit pattern. The client-side payment event was treated as an external transaction signal rather than as sufficient evidence for booking creation. The server verified the transaction reference with the Paystack application programming interface (API) (Paystack, n.d.-b) before persisting the booking as pending. This separated payment initiation from the application-level state transition and reduced reliance on client-side confirmation.

2.8. Security, privacy and trust design

Access was restricted to registered staff. Registration used Staff ID and institutional e-mail, and authenticated users were required before accessing ride, booking, and payment pages. Card details were entered through the Paystack payment interface rather than directly into the application. The ride cards exposed the driver's phone number to authenticated colleagues to support coordination.

The system was designed with privacy and security considerations in mind, but production deployment would require formal assessment against the Nigeria Data Protection Act, 2023 (Federal Republic of Nigeria, 2023) and contemporary application-security guidance such as the Open Worldwide Application Security Project (OWASP) Top 10 (OWASP Foundation, 2021). The present study did not claim completion of a penetration test, formal privacy impact assessment, or security certification.

2.9. Functional testing

Black-box functional testing was used to evaluate the workflows specified in the requirements. Test accounts were used through a web browser. Payment scenarios were executed in Paystack test mode (Paystack, n.d.-a), including simulated successful and unsuccessful outcomes. Evidence consisted of screen captures, delivered e-mails, and inspection of database records through phpMyAdmin. Response time, load, formal security, and instrumented usability tests were outside the scope of the evaluation.

3. Results

3.1. Implemented artifact

The implemented platform provided a public landing page, staff registration and login, an authenticated dashboard, ride publication, ride discovery, booking management, profile functions, and withdrawal requests. The Find a Ride interface presented route, date, time, driver, vehicle, remaining seats, fare, and driver notes. The Confirm & Pay page repeated the essential ride information before payment. The Manage Bookings page provided the driver with pending booking information and approval or rejection controls.

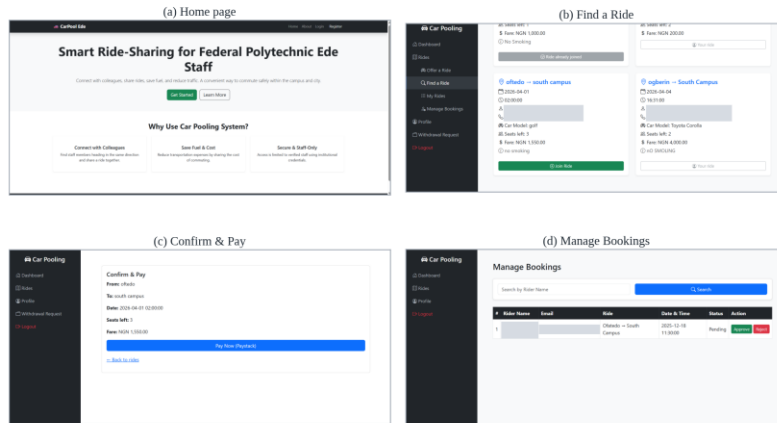


Fig. 4. Selected pages of the implemented system: (a) home page; (b) Find a Ride; (c) Confirm & Pay; (d) Manage Bookings. Names, phone numbers and e-mail addresses in (b) and (d) were masked.

3.2. Functional validation

Table 4. Functional test cases and observed outcomes.

Test ID	Workflow	Expected behaviour	Observed outcome	Result
T1	Staff registration	Account created with required staff information	Account created and persisted in users table	Passed
T2	Login	Registered staff reach dashboard	Dashboard displayed with user identity and department	Passed
T3	Publish ride	Ride stored and listed	Ride appeared with route, time, driver, seats and fare	Passed
T4	Browse rides	Available rides displayed	Ride cards displayed with Join Ride action	Passed
T5	Booking guards	Self and duplicate booking prevented	Buttons changed to 'Your ride' or 'Ride already joined'	Passed
T6	Confirm & Pay	Ride information shown before payment	Confirmation page displayed and Pay Now opened gateway	Passed
T7	Test payment	Payment verified and booking created	Success message and transaction reference recorded	Passed
T8	Driver notification	Driver notified after paid booking	E-mail received with rider, route, date and matching reference	Passed
T9	Booking decision	Driver can approve/reject	Pending booking status changed after decision	Passed
T10	Passenger notification	Passenger receives decision	Booking Status Update e-mail received	Passed
T11	My Rides	Offered and joined rides separated	Both ride categories displayed	Passed
T12	Dashboard/withdrawal	Earnings visible and withdrawal logged	Transactions displayed and withdrawal request recorded	Passed

All twelve tested workflows behaved as specified. In the payment test, the transaction reference displayed after the successful test payment matched the reference contained in the driver's notification. This agreement provided functional evidence that the payment-verification, transaction-recording, and notification sequence operated coherently under the test conditions.

3.3. Requirements traceability

Table 5. Requirements traceability and validation evidence.

Capability	Requirement(s)	Validation evidence
Institutional identity	FR1, FR2, NFR1	T1–T2
Ride supply and discovery	FR3, FR4	T3–T4
Booking integrity	FR5, FR7	T5, T9
Payment integrity	FR6	T6–T7
Communication	FR8	T8, T10
Earnings and settlement	FR9	T12
Data persistence	NFR2	T1, T3, T7, T12
Administration and oversight	FR10	Administrator accounts implemented; no dedicated test case in this study
Mobile-oriented interface	NFR3	Implemented responsive interface; no instrumented usability score

4. Discussion of Results

The principal result was a functioning closed-community mobility artifact rather than a measured transport-system intervention. The twelve functional tests demonstrated that the platform could execute the principal transaction lifecycle under controlled test conditions. This is important because institutional carpooling requires more than ride publication: a user must be identifiable, a ride must have explicit capacity and fare, a payment must be verifiable, a booking must have an authorisation state, and resulting financial events must remain traceable.

The design also provided a practical response to the trust problem identified in previous carpooling research. Restricting access to institutional staff did not eliminate trust concerns, but it changed the trust model from anonymous participation toward organisational membership. This approach was consistent with the club-like institutional structure discussed by Correia and Viegas (2011).

The payment workflow was another distinguishing feature. Existing institutional carpooling systems often emphasise matching, whereas the present artifact treated financial settlement as part of the ride lifecycle. The verify-before-commit mechanism ensured that a client-side payment signal did not directly create a booking. The resulting workflow was: Eligible user → Ride offer → Ride discovery → Fare confirmation → Payment → Server verification → Pending booking → Driver decision → Notification → Settlement.

This transaction-oriented perspective distinguished the platform from a simple ride-listing application and created an extensible foundation for future matching algorithms, reputation services, and institutional mobility analytics.

4.1. Scalability and matching

The principal functional limitation was manual ride discovery. Passengers currently browse published rides rather than receiving ranked matches based on route overlap, departure-time tolerance, available seats, recurring schedules, or detour constraints. This design was suitable for an initial small-community deployment but may become inefficient as the number of staff and rides increases. A natural next stage is a constraint-aware institutional matching engine. Such an engine can represent a ride as a tuple of origin, destination, departure time, available seats, and fare, and a passenger request using an origin, destination, acceptable time window, and other preferences. Candidate rides can then be filtered by hard constraints and ranked using a weighted compatibility score. Because this extension was not implemented in the present artifact, it is treated as a future architecture rather than an evaluated result.

4.2. Economic and environmental implications

Carpooling can reduce the number of vehicles required to transport a given number of commuters and can distribute fuel and travel costs among participants. The present system created the digital mechanism through which such sharing could occur, but the study did not measure fuel savings, vehicle-kilometres avoided, congestion, emissions, or net monetary savings after payment-gateway charges. Those outcomes require a prospective deployment study with actual users and trip records.

4.3. Privacy, security and governance

Closed institutional mobility systems process personally identifiable information, contact information, travel patterns, and financial transaction records. These data create governance requirements beyond ordinary web

application functionality. A production deployment should therefore implement least-privilege access, secure credential storage, session protection, audit logging, input validation, rate limiting, privacy-aware contact disclosure, secure secret management, backup and recovery procedures, and formal security testing.

Future versions should also reconsider whether a driver's phone number needs to be visible to every authenticated colleague. A platform-level messaging or masked-contact mechanism could preserve coordination while reducing unnecessary disclosure of personal contact information.

4.4. Usability and adoption

Although the interface was designed for simple navigation and responsive display, no standard usability instrument was administered. Future evaluation should therefore include the System Usability Scale (Brooke, 1996) or a comparable validated instrument, together with task-completion time, error rate, perceived trust, perceived usefulness, and intention to continue using the service.

4.5. Generalisability

The system was designed around the institutional context of Federal Polytechnic Ede. Its architecture uses widely available web technologies and therefore has potential for adaptation to other institutions, but transferability should not be assumed without testing differences in staff structure, commuting geography, institutional policies, payment practices, and privacy requirements.

5. Recommendations and Future Research

Based on the functional results and the identified limitations, the following recommendations are proposed for subsequent development and evaluation:

- Develop a constraint-aware route and schedule matching algorithm that ranks compatible rides instead of relying on manual browsing.
- Introduce optional recurring-commute profiles for staff travelling on predictable schedules.
- Evaluate a privacy-preserving contact mechanism that separates identity verification from unnecessary exposure of personal phone numbers.
- Implement reputation and feedback mechanisms after sufficient real-world usage data are available.
- Evaluate usability using a validated instrument and task-based measures.
- Conduct a prospective pilot measuring participation, repeat use, shared-trip counts, occupancy, fuel expenditure, and commuting time.
- Estimate changes in vehicle-kilometres travelled and associated emissions using actual trip records rather than assumed environmental benefits.
- Evaluate payment and payout reliability under production conditions and quantify transaction costs.
- Conduct formal application-security testing and a privacy impact assessment before institutional deployment.
- Develop an institutional analytics layer for monitoring ride demand, unmet requests, route concentration, and temporal demand patterns.

6. Conclusion

The study developed and functionally validated a closed-community digital carpooling platform for staff of Federal Polytechnic Ede, Nigeria. The system integrated institutional identity, ride publication, fare visibility, external payment, server-side transaction verification, booking authorisation, notification, transaction logging, earnings visibility, and withdrawal requests within a single workflow. All twelve functional test cases passed under the specified test conditions, demonstrating the feasibility of the transaction lifecycle from ride publication through payment, approval, notification, and settlement. The principal contribution was the treatment of institutional carpooling as a closed-community mobility transaction rather than solely as a matching service, thereby embedding identity control and financial traceability into the application workflow. The evidence did not establish user adoption, cost savings, congestion reduction, emissions reduction, usability, security assurance, or scalability. Further research should therefore combine the implemented architecture with constraint-aware matching, validated usability assessment, formal security and privacy evaluation, and longitudinal mobility data from real institutional users.

Acknowledgements

The authors acknowledge the staff of Federal Polytechnic Ede who contributed contextual information during the requirements analysis and functional evaluation of the system.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

- Agatz N, Erera A, Savelsbergh M, Wang X. Optimization for dynamic ride-sharing: A review. *Eur J Oper Res* 2012;223(2):295–303.
- Brooke J. SUS: A quick and dirty usability scale. In: Jordan PW, Thomas B, Weerdmeester BA, McClelland IL, editors. *Usability Evaluation in Industry*, London: Taylor & Francis; 1996:189–94.
- Bruglieri M, Ciccarelli D, Colomi A, Luè A. PoliUniPool: A carpooling system for universities. *Procedia Soc Behav Sci* 2011;20:558–67.
- Correia G, Viegas JM. Carpooling and carpool clubs: Clarifying concepts and assessing value enhancement possibilities through a stated preference web survey in Lisbon, Portugal. *Transp Res Part A Policy Pract* 2011;45(2):81–90.
- Elmasri R, Navathe SB. Fundamentals of Database Systems, 7th ed. Pearson, 2016.
- Federal Republic of Nigeria. Nigeria Data Protection Act, 2023.
- Furuhata M, Dessouky M, Ordóñez F, Brunet M-E, Wang X, Koenig S. Ridesharing: The state-of-the-art and future directions. *Transp Res Part B Methodol* 2013;57:28–46.
- Javid MA, Al-Shehhi MIS, Al-Youqabi MHS. Students' intentions toward carpooling in Sohar, Oman. *Trans Transp Sci* 2025;16(2):23–32.
- Maiguwa CC, Idoma K, Maina B, Adamu D. The implication of fuel subsidy removal on transportation mode choice in Gombe metropolis, Nigeria. *WIT Trans Ecol Environ* 2026;267:105–15.
- Onikosi-Alliyu SO. The impact of subsidy removal on the efficacy and adoption of casual carpooling schemes in Nigeria. *Al-Hikmah J Econ* 2025;6(1).
- OWASP Foundation. OWASP Top 10:2021. [Online]. Available: <https://owasp.org/Top10/>
- Paystack. (n.d.-a). *Test payments*. <https://paystack.com/docs/payments/test-payments/>
- Paystack. (n.d.-b). *Verify payments*. <https://paystack.com/docs/payments/verify-payments/>
- Sommerville I. Software Engineering, 10th ed. Pearson, 2016.