INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 5, May 2026 | ISSN: 2456-4184 | IJNRD.ORG

IndoLink: A Multi-role E-Commerce Framework Implementing Payment for Order Flow with AI-Based Product Discovery

[1]Bhuvan, [2]Chandan C V, [3]Deekshith Sai Raj S P, [4]Dhruthik S Gowda, [5]Ranjith J

[1]Student, [2]Student, [3]Student, [4]Student, [5]Associate Professor,
Department of Information Science and Engineering,
SJB Institute of Technology
Bengaluru, Karnataka, India

Abstract: Online shopping platforms have grown rapidly over the past decade, yet most of them still follow a simple model where buyers and sellers interact directly. This basic setup does not support features like controlled pricing, intermediary roles,or intelligent product search. This paper describes IndoLink, a new marketplace platform built around four user roles: Buyer,Seller, Admin, and Broker. Each role has its own dashboard and sets of responsibilities. The platform introduces a Payment forOrder Flow approach where money moves from the buyer to the admin and then to the seller, keeping all transactions traceableand structured. A product search feature powered by the Google Gemini AI model lets users find items by uploading a photoinstead of typing keywords. A referral system allows brokers to earn a five percent commission on profits from transactionsmade by users they bring to the platform. Payments are handled securely through Razorpay. The backend is built with Node.js and Express.js, the frontend with React and Vite, and MongoDB is used for data storage. Testing results show that the system works correctly across all roles and supports a transparent, scalable commerce environment.

IndexTerms **- E-Commerce, Payment for Order Flow, Multi-Role Marketplace, Artificial Intelligence, Image Search, Commission Systems, Digital Marketplace, Brokerage Model**

I. INTRODUCTION

Online commerce has transformed the way people buy and sell products. Platforms like Amazon and Flipkart have made it easy for buyers to discover products and complete purchases from anywhere. However, these platforms are primarily designed around a two-party model, where a seller lists a product and a buyer purchases it. The interaction typically ends there, leaving no room for a third party to supervise pricing or for individuals to earn by referring new users to the platform. This design creates several limitations. Sellers independently set prices, which can lead to inconsistencies. There is no centralized authority to verify whether pricing is fair or whether products meet quality standards before reaching buyers. Additionally, buyers often face difficulty in searching for products when they lack precise keywords or brand names. Furthermore, businesses that rely on referral-based growth lack built-in mechanisms to reward individuals who bring new users to the platform. To address these challenges, IndoLink was designed as a structured four-role marketplace. The platform introduces four key roles: Buyer, Seller, Admin, and Broker. Each role has a distinct responsibility. Sellers provide products, the admin reviews and reprices them before listing, buyers purchase from the curated catalogue, and brokers invite new users while earning commissions from their transactions. IndoLink stands out due to three core features. First, a Payment for Order Flow model routes all transactions through the admin layer, ensuring transparency and auditability. Second, an AI-powered image search system, built using the Google Gemini API, allows users to upload images and find matching products without relying on exact keywords. Third, an automated commission system tracks and distributes broker earnings based on completed transactions.

This paper presents the design, implementation, and evaluation of IndoLink. Section II reviews related work, Section III outlines the proposed system, Section IV details the architecture and methodology, Section V discusses results and observations, and Section VI concludes with future directions.

II. RELATED WORK

A. Limitations of Existing E-Commerce Platforms

Most e-commerce platforms in use today were designed for simplicity. They give sellers a way to list products and buyers a way to purchase them. This works well at a small scale but starts to show weaknesses as businesses grow. Pricing is inconsistent because each seller decides their own rates. There is no mechanism for a platform controller to verify product quality before it reaches buyers. Referral-based growth is either absent or handled externally, not within the platform itself [6]. Research on multi-sided platforms shows that adding structured roles to a marketplace can improve trust, revenue diversity, and user retention [7]. Studies on pricing in digital markets also confirm that centralized price oversight reduces buyer confusion and increases confidence in the catalogue [8]. IndoLink draws directly from these findings by placing the Admin at the centre of all pricing and procurement decisions.

B. AI Techniques for Product Search

Text-based search works only when the user knows what to look for. If someone sees a product in a photograph but does not know its name, keyword search is not helpful. Visual search systems address this by accepting images as input and returning similar items from a catalogue. Earlier work on image-based retrieval used convolutional neural networks to extract visual features and compare them against stored product embeddings [9]. These approaches achieved strong results but required specialized infrastructure. Newer large multimodal models like Google Gemini [10] provide a simpler path. They accept an image, identify what is shown, and return descriptive output that can be used to query a product catalogue. IndoLink uses this approach to build its AI search feature.

C. Payment Architectures in Marketplaces

Handling payments in a multi-party marketplace is more complex than in a direct sale. The money must flow in a controlled sequence, and each transfer must be verified. Research on multi-party payment flows shows that an intermediary payment model, where a platform holds funds before releasing them to sellers, reduces fraud and makes dispute resolution easier [11]. Razorpay [12] is a payment gateway that supports this kind of structured routing. It provides webhook-based payment confirmation, multiple payment modes including UPI, cards, and net banking, and a clean API for building custom payment logic. IndoLink uses Razorpay to handle both the buyer-to-admin and admin-to-seller payment paths.

D. Referral and Commission Systems

Referral systems have been studied as a cost-effective way to grow a platform's user base. When a current user refers someone new and that new user makes a purchase, both parties benefit. The referring user gets a reward, and the platform gains a customer it did not have to advertise for [13]. Most platforms implement referral systems as optional add-ons, not as core features. Commission calculations are often done manually or through separate tools. IndoLink integrates broker commission tracking directly into the order and payment flow, so calculations happen automatically each time a qualifying transaction is completed.

E. Networked System Design and Security

Work on IoT communication protocols, including MQTT over MPQUIC [4], shows how careful design of transport layers can improve reliability and reduce latency in distributed systems. Research on network traffic analysis for malware detection [5] demonstrates the value of monitoring structured data flows for security purposes. These principles influence IndoLink's backend design, where each module handles a single responsibility and all data flows are logged for traceability.

III. PROPOSED SYSTEM

A. Overview

IndoLink is a web-based marketplace with a client-server structure. The frontend is served to users through a browser. The backend processes all business logic and exposes a REST API. A MongoDB database stores all platform data. Two external services are integrated: Razorpay for payments and Google Gemini for AI image analysis. The platform supports four user roles. Each role has a separate dashboard and a defined set of actions. Roles are assigned during registration and enforced on every API call through JWT-based access control.

B. User Roles and Responsibilities

Buyer: A buyer registers on the platform, browses available products, adds items to a cart, and completes purchases. Payment is made directly to the admin through Razorpay. The buyer can also track order status from their dashboard.
Seller: A seller lists products with names, descriptions, prices, categories, images, and stock quantities. They do not sell directly to buyers. Instead, the admin purchases their products and resells them. The seller receives payment from the admin through Razorpay.
Admin: The admin is the central controller of the marketplace. They review products submitted by sellers, decide which ones to purchase, set a new selling price with markup, and make those products available to buyers. They also handle buyer orders, update delivery status, and monitor platform analytics.
Broker: A broker receives a unique referral code upon registration. They share this code with people outside the platform. When someone registers using that code and completes a transaction, the broker earns five percent of the profit from that sale. The broker dashboard shows total earnings, referred users, and commission status.

C. Payment for Order Flow Model

The PFOF model in IndoLink defines two payment paths. The first path runs from buyer to admin. When a buyer confirms an order, they pay the admin using Razorpay. The admin's Razorpay credentials are configured at the platform level. The second path runs from admin to seller. When the admin purchases a product from a seller, they pay the seller using the seller's own Razorpay credentials, which the seller configures in their dashboard. Every payment is verified through webhooks. A transaction record is created for each payment, storing the order ID, amount, gateway reference, and status. This means every rupee that moves through the platform is traceable.

D. AI Image Search

When a buyer wants to search using an image, they upload a photo through the search interface. The backend sends that image to the Google Gemini API. Gemini analyzes the image and returns a description of the object, including its likely category and relevant attributes. This description is then used to query the MongoDB product catalog and return a list of matching items. The buyer sees

the results just as they would with a normal text search. This feature is especially useful when someone knows what a product looks like but not what it is called.

E. Commission Calculation

Commission is calculated automatically at the time an order is completed. The system checks whether the buyer who placed the order was referred by a broker. If so, it calculates the profit on that transaction as the difference between the admin's selling price and the purchase price paid to the seller. Five percent of that profit is recorded as a broker earning. The Admin can see all pending and paid commissions and can release payments to brokers when appropriate.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Architecture

The IndoLink system is divided into four layers as shown in Fig. 1. The frontend layer is built with React and Vite. Tailwind CSS handles styling. React Router manages navigation between pages. Each role sees a different view after login. The backend layer runs on Node.js with the Express.js framework. It exposes REST API endpoints for all platform operations. JWT tokens are used for session management. Bcrypt handles password hashing. Multer processes image uploads. The database layer uses MongoDB with Mongoose ODM. Collections include users, products, admin products, orders, carts, payments, broker earnings, and categories. Data relationships are managed through document references. The external services layer connects to Razorpay for payment processing and to the Google Gemini API for AI image analysis.

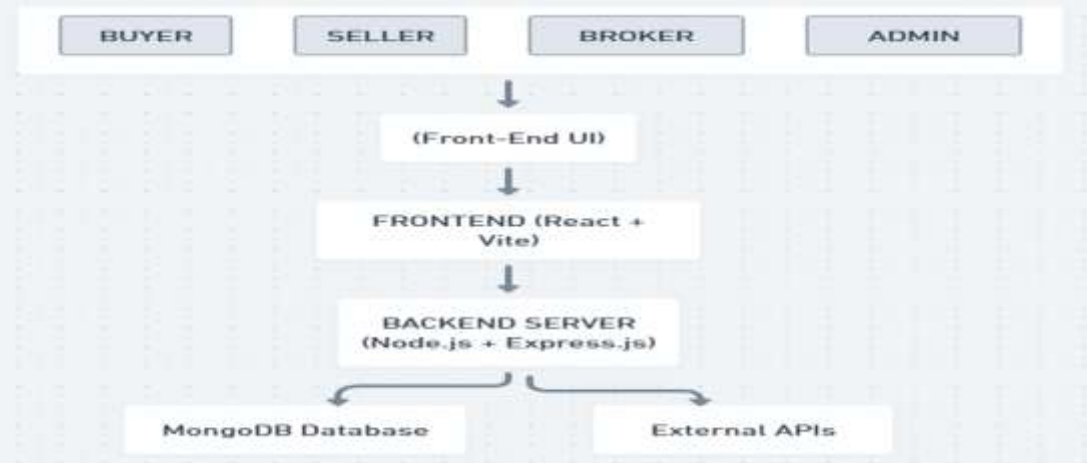

Fig. 1. IndoLink System Architecture

B. Data Model

The class diagram in Fig. 2 shows how the main entities relate to each other. The User entity is central. It stores the user's name, email, hashed password, role, broker code, and referral information. All other entities reference users by ID. The Product entity stores items submitted by sellers. When the admin buys a product, a new AdminProduct record is created that links back to the original product and stores the admin's purchase price and the markup selling price.The Order entity captures each transaction. It links to the buyer, admin, seller, and a list of OrderItem records. The Cart and CartItem entities hold a buyer's pending selections be-fore checkout. The BrokerEarning entity records commission events, linking a broker to a specific order, buyer, seller, and the profit and commission amounts.

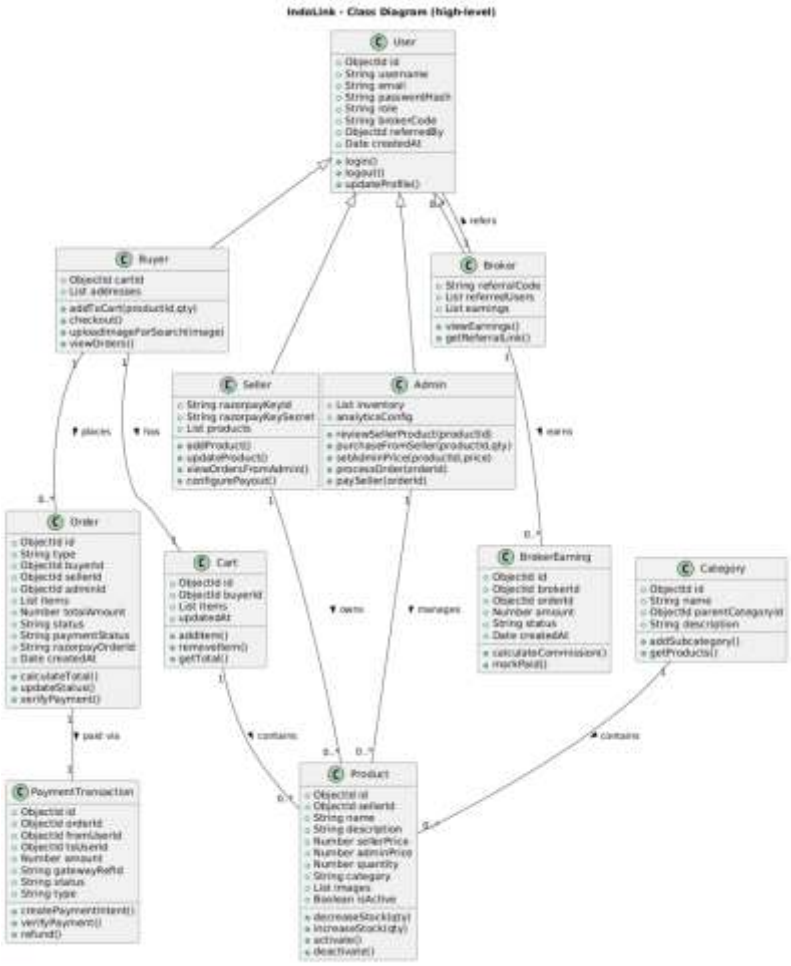

Fig. 2. IndoLink Class Diagram

C. Module Descriptions

1. User Management: This module handles registration, login, and role-based access. When a user registers, their password is hashed and stored. On login, the server issues a signed JWT. Every protected API route checks this token and verifies the user's role before processing the request.
2. Product Management: Sellers add products through a form that accepts images, price, stock, category, and a description. These products appear in the admin's review panel. The admin can buy any product, which creates a corresponding Admi Product record. The admin then sets a selling price before the product appears in the buyer catalogue.
3. Cart and Order Management: Buyers add products to their cart. At checkout, the system verifies that all items are in stock, calculates the total, and sends a payment request to Razorpay. If stock is insufficient, the buyer is notified immediately. On successful payment, the order is confirmed, inventory is reduced, and order records are updated.
4. Payment Processing: Razorpay handles all fund transfers. For buyer purchases, the system creates a payment order using the admin's API keys and redirects the buyer to the Razorpay interface. A webhook listens for payment confirmation. For admin-to-seller payments, the admin initiates a transfer using the seller's configured credentials. All payment events are stored in the Payments collection.
5. Inventory Management: Each product has a stock quantity field. When a buyer adds items to a cart and proceeds to checkout, the system temporarily reserves that quantity. If payment succeeds, the reservation becomes permanent and stock is decremented. If payment fails or the order is cancelled, the reserved quantity is returned to available stock.
6. AI Image Search: The buyer uploads an image using the search bar in the buyer dashboard. The backend reads the image file, converts it to base64, and sends it to the Google Gemini API with a prompt asking for product category and attributes. The API returns a text response. The backend parses this response and runs a text search on the product catalog using the extracted keywords.
7. Broker and Commission: When a new user registers with a referral code, their account is linked to the broker who owns that code. After every completed order, the system checks if the buyer is linked to a broker. If so, it computes five percent of the margin profit and creates a Broker Earning record. The broker's dashboard aggregates all earnings and shows each commission entry with its associated transaction.
8. Analytics and Reporting: The admin dashboard shows total revenue, active sellers, total orders, and total products. A line chart shows revenue over the past thirty days. A pie chart shows sales by category. Sellers see their active listings and sold item counts. The admin can also submit a product to the Gemini AI Market Analysis tool, which returns a recommended market price, a confidence score, and key demand insights.

D. Checkout Process

The buyer checkout process involves several steps. First, the buyer clicks checkout, and the Cart Service creates a draft order. The Order Service then validates each item and checks inventory. If items are available, the system calls Razorpay to create a payment intent and returns a payment URL to the buyer. The buyer completes payment on the Razorpay interface. A webhook confirms success or failure. On success, stock is decremented, the order status is set to confirmed, and a notification is sent to the buyer. On failure, stock reservation is released and the buyer sees an error. The full sequence is shown in Fig. 3.

E. Workflow Diagrams

Each user role follows a defined workflow. The buyer workflow shown in Fig. 4 moves from login through product search, cart management, payment, and order tracking. The seller workflow covers product listing, payment credential setup, and inventory updates. The admin workflow includes product review, procurement, pricing, and order management. The broker workflow includes referral sharing, user tracking, and commission monitoring.

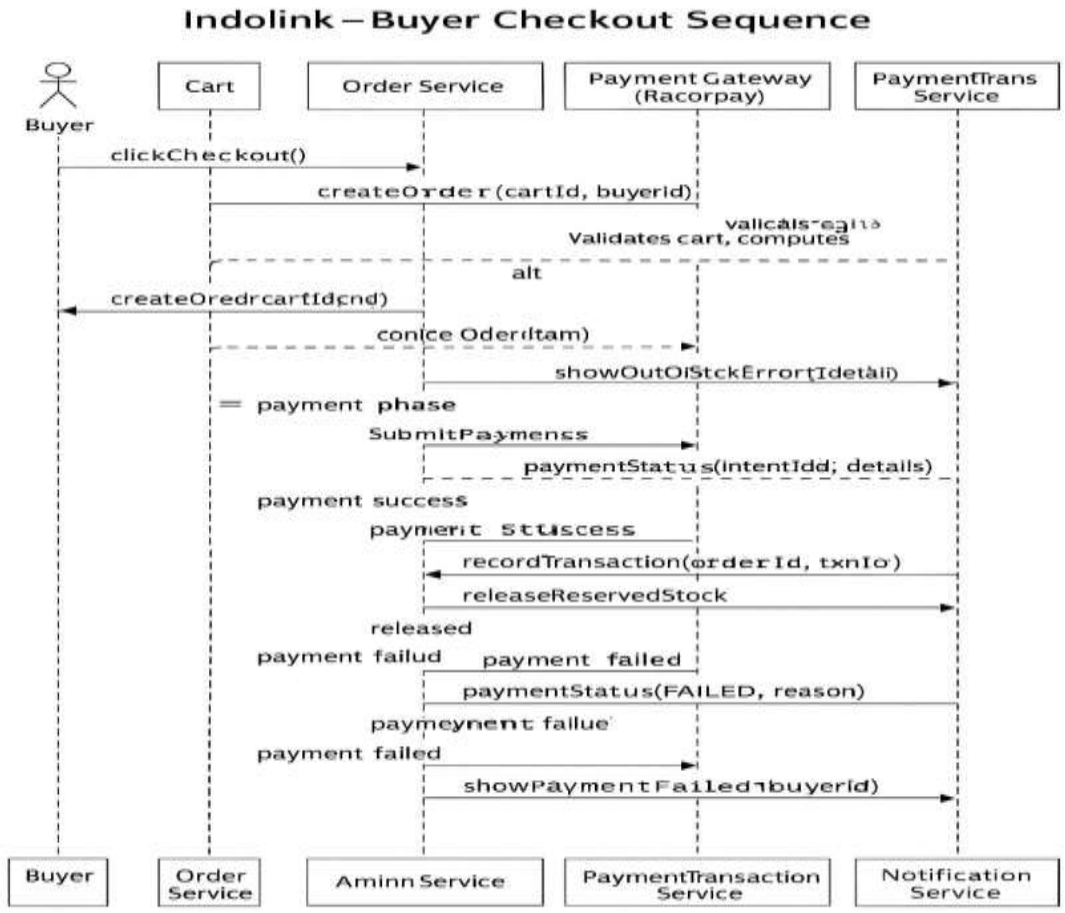

Fig. 3. Buyer Checkout Sequence Diagram

setup, and order monitoring. The admin workflow includes seller product review, procurement, markup pricing, buyer order processing, and analytics. The broker workflow covers referral code sharing, user tracking, and commission viewing.

F. Testing

Three testing levels were used. Unit tests written in Jest checked individual functions including password hashing, JWT token generation and verification, commission calculation, order total computation, and input sanitization. Key test cases are listed in Table I.

TABLE I
SELECTED UNIT TEST CASES

ID	Description	Expected Output
U-001	Hash and compare password	Match returns true
U-003	JWT sign and verify	Decoded payloads correct
U-005	Commission at 2 percent	Result equals 20.00
U-010	Price 10, quantity 5	Total equals 50
U-011	Script tag in input	Script is removed

Integration tests checked how modules work together. These tests covered the full signup and login flow, product creation and retrieval, order placement with commission ledger creation, payment webhook handling for both success and failure cases, and role-based access enforcement. A buyer token attempting to access admin route correctly received a 403 Forbidden response. Sending the same webhook twice produced only one state change, confirming idempotent handling. System tests ran the complete platform in a staging environment. The full Producer to Broker to Buyer workflow was executed and verified. Marketplace search and category filters returned accurate, paginated results. All functional test cases passed without errors.

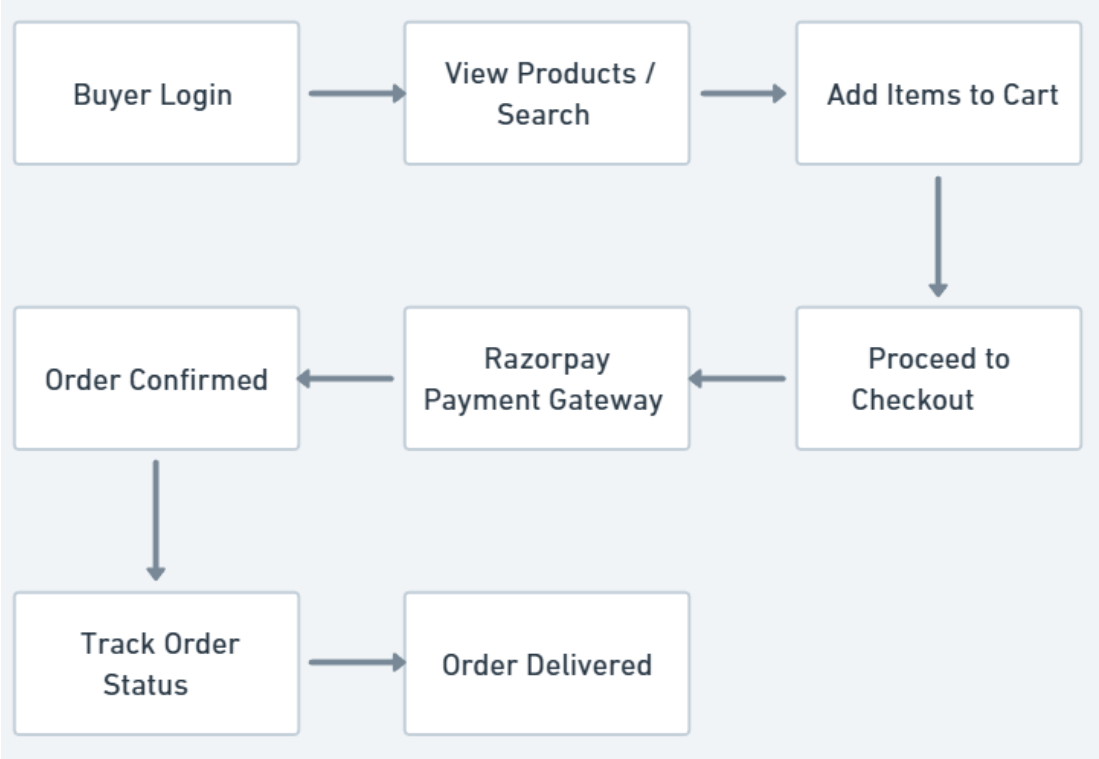

Fig. 4. Buyer Workflow

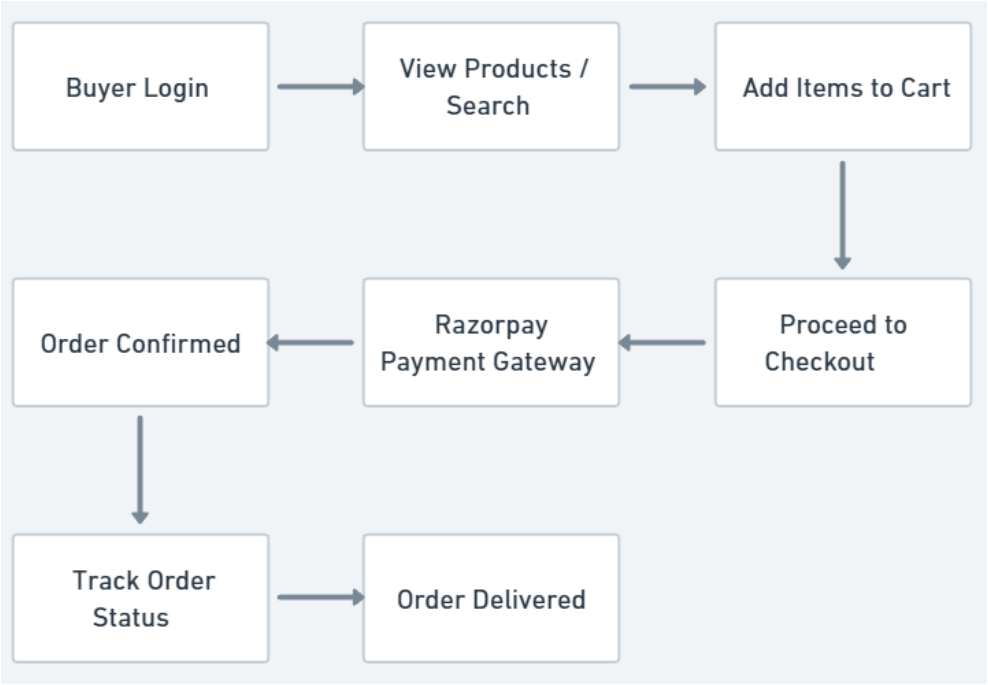

Fig. 5. Broker Workflow

V. RESULTS AND DISCUSSION

A. User Authentication

Login and role-based redirection worked correctly for all four roles. Each user was sent to the appropriate dashboard after authentication. Error messages appeared for incorrect credentials. JWT tokens were verified on every API request. A screenshot of the sign-in page is shown in Fig. 6.

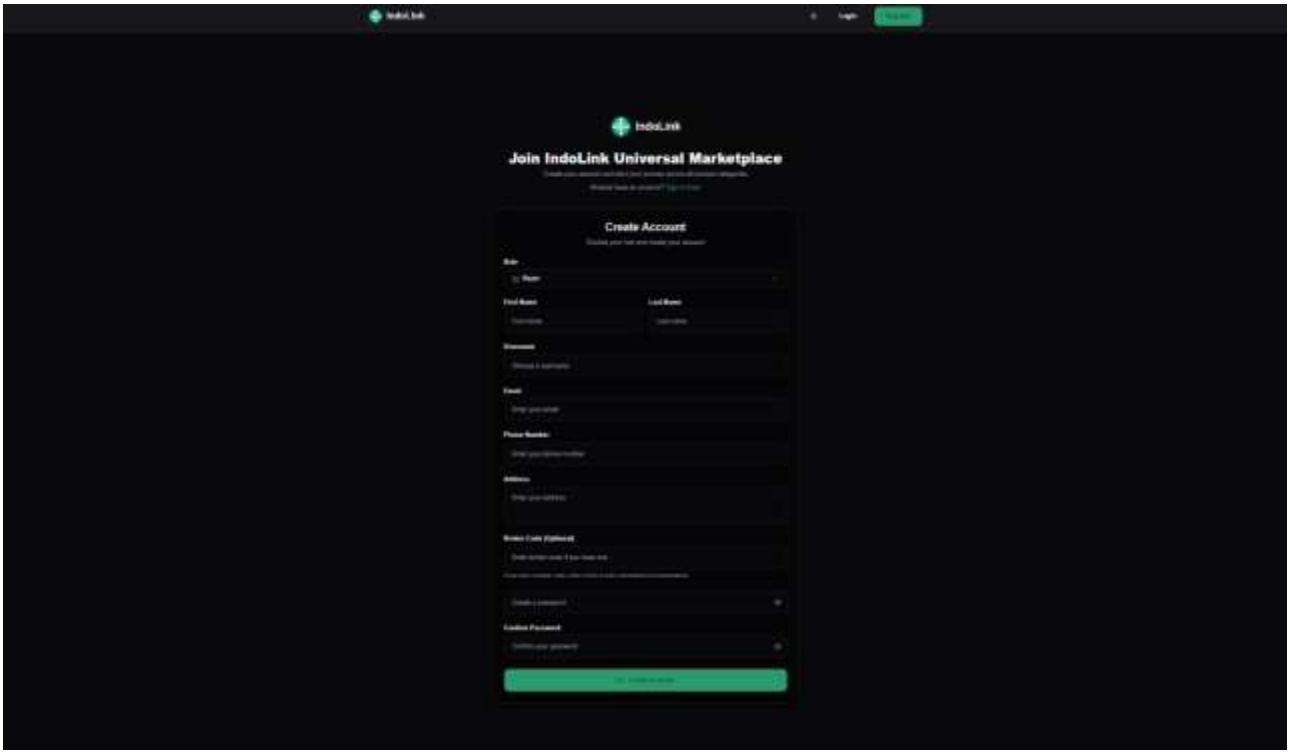

Fig. 6. IndoLink Sign-In Interface

B. Buyer Dashboard

The buyer dashboard showed a product catalogue with category filters along the top. Buyers could search by typing a product name or by uploading an image. During testing, uploading a photograph of black pepper correctly returned pepper-related products from

the catalogue. The AI search results matched visually similar items with good accuracy. Cart management, order placement, and order history all worked as expected. The buyer dashboard is shown in Fig. 7.

Fig. 7. Buyer Dashboard with AI Image Search

C. Seller Dashboard

The seller dashboard displayed a list of active product listings, a form to add new products, and an order history section showing purchases made by the admin. A test seller with ten active listings received purchase notifications from the admin. Payment confirmations appeared correctly after the admin completed the transfer. The seller dashboard is shown in Fig. 8.

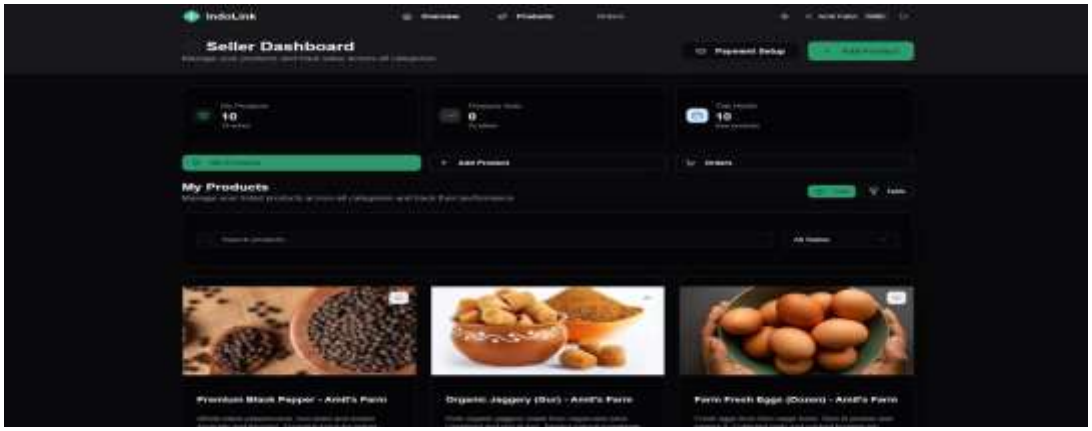

Fig. 8. Seller Dashboard

D. Admin Dashboard

The admin dashboard showed platform-wide statistics. In a test session with real data, the dashboard reported a platform revenue of Rs. 2,804,892, with 60 products, 6 active sellers, and 82 total orders. Revenue trend charts updated based on transaction records. The category distribution chart showed Grocery and Gourmet at 97 percent and Health and Household at 3 percent, matching the test data. The AI Market Analysis feature returned a recommended price and a demand summary for a queried product. The admin dashboard is shown in Fig. 9.

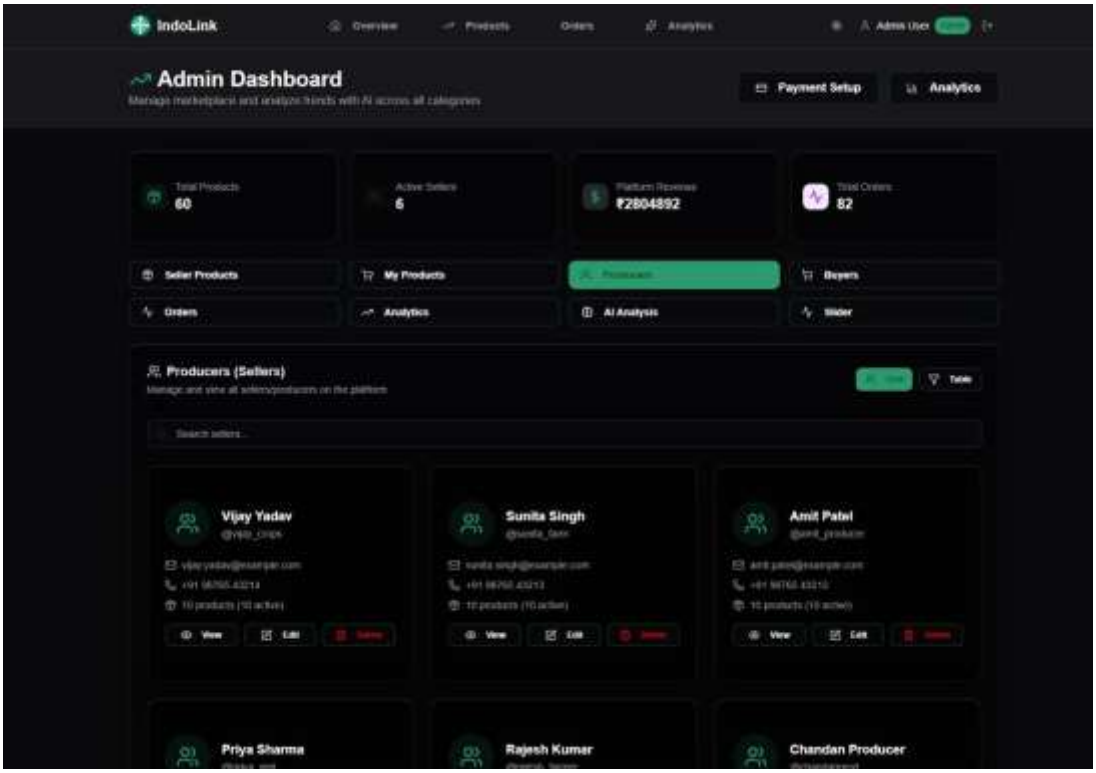

Fig. 9. Admin Dashboard with Analytics and AI Insights

E. Broker Dashboard

In testing, a broker with three referred users accumulated earnings of Rs. 1,111.00 from five qualifying transactions. Each commission entry displayed the seller, buyer, profit amount, and the 5 percent commission value. The referral code was correctly linked to all three users during registration. The broker dashboard is shown in Fig. 10.

F. Payment Processing

Payment tests covered UPI, card, net banking, and wallet modes. In all success cases, the order status changed to PAID, inventory was decremented, and the admin received confirmation. In simulated failure cases, reserved stock was returned and order status was set to FAILED. Sending the same webhook payload twice produced no duplicate changes, confirming that the idempotency check worked correctly.

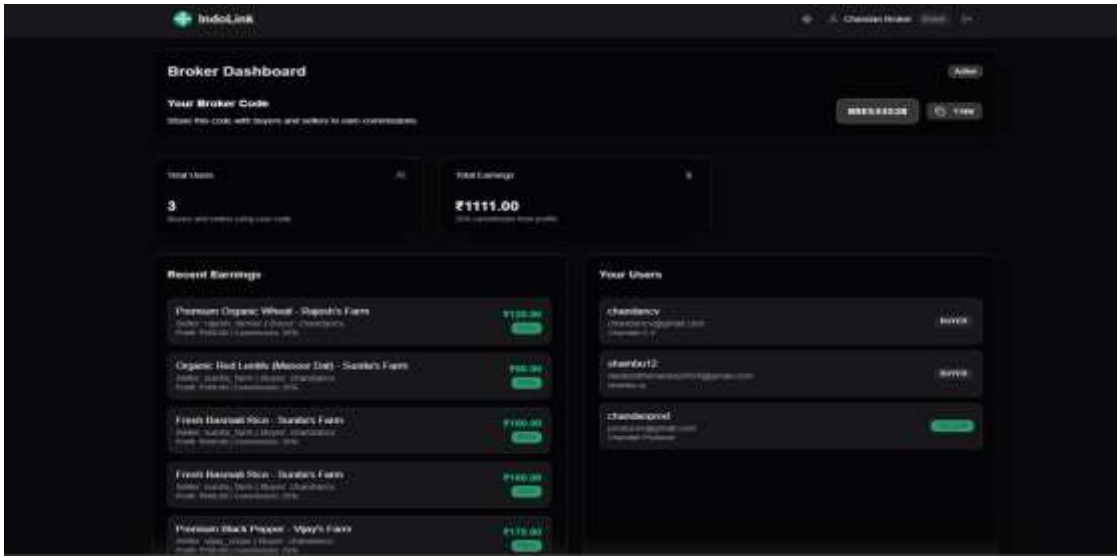

Fig. 10. Broker Dashboard with Commission Tracking

G. Feature Comparison

Table II compares IndoLink features against the capabilities of typical conventional e-commerce platforms. IndoLink adds multi-role participation, centralized admin pricing, AI image search, commission tracking, and structured two-path payment routing, none of which are standard in most existing platforms.

TABLE II
feature comparison: IndoLink vs. conventional platforms

Feature	**Conventional**	**IndoLink**
Multi-Role Support	No	Yes
Centralized Pricing Control	No	Yes
AI Image Search	No	Yes
Commission-Based Referral	No	Yes
Structured PFOF Routing	No	Yes
Role-Based Dashboards	Limited	Yes
Analytics with AI Insights	Limited	Yes
Secure Payment Gateway	Yes	Yes

H. Discussion
IndoLink shows that combining structured roles, AI features, and transparent payment routing is achievable within a standard full-stack web architecture. The admin-centric PFOF model creates a controlled pricing environment without making the platform difficult to use. The AI image search reduces the effort buyers need to find products they cannot name. The commission system gives brokers a concrete reason to promote the platform, which supports organic growth.

The modular backend design made it straightforward to test and extend individual components. Each module handles one responsibility, which reduced the number of bugs introduced when making changes. The payment webhook idempotency check proved important during testing, as repeated webhook delivery is a common behaviour of payment gateways.

One area for improvement is latency in the AI search path. The round trip to the Gemini API adds a few hundred milliseconds to the search response time. Caching responses for common image queries would reduce this delay. Commission calculation currently uses a fixed five percent rate, which could be made configurable per broker tier in a future version.

VI. CONCLUSION AND FUTURE WORK

This paper presented IndoLink, a multi-role e-commerce platform that brings together Payment for Order Flow, AI-powered image search, and a broker commission system in one marketplace. The platform gives buyers a better way to find products, gives sellers a structured channel to reach customers, gives brokers a reason to grow the user base, and gives the admin full control over pricing and fulfilment. All four roles were implemented with dedicated dashboards and tested thoroughly across unit, integration, and system levels.

The system was built using React, Node.js, Express.js, and MongoDB. Razorpay handled secure payments and Google Gemini provided AI inference for product search. Results from testing confirmed that transactions work correctly, commissions are calculated accurately, and all access controls are enforced as designed.

Future work will focus on several areas. A mobile application will make the platform more accessible to users on smartphones. Integration with logistics APIs will allow real-time delivery tracking directly from the dashboard. Support for automated GST invoicing will make the platform usable for professional and large-scale trade. Multilingual and multi-currency support will open the platform to international markets. A personalized product recommendation engine will improve the shopping experience for repeat buyers. Chat support and chatbot tools will help users get answers faster. A seller rating and review system will help buyers evaluate product quality and build trust in the marketplace over time.

REFERENCES

[1] K. Laudon and C. G. Traver, *E-Commerce: Business, Technology, Society*, 17th ed. Pearson Education, 2021.
[2] P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
[3] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*, 5th ed. Addison-Wesley, 2011.
[4] Y. Ren, Z. Du, Y. Lin, J. Wu, and G. Han, “MQTT over MPQUIC: A multipath transport architecture for IoT networks,” in *Proc. 2025 Int. Conf. on Meta-Networking (MEET)*, 2025, pp. 1–5.
[5] M. Kapoor, N. Patidar, and N. Arya, “Analyzing network traffic features for IoT malware detection through forensic methods,” in *Proc. 14th IEEE Int. Conf. on Communication Systems and Network Technologies (CSNT)*, 2025, pp. 334–339.
[6] B. Burt and D. Rosen, “Trust and transparency in digital marketplaces,” *Journal of Business Research*, vol. 89, pp. 292–301, 2018.
[7] T. Eisenmann, G. Parker, and M. Van Alstyne, “Strategies for two-sided markets,” *Harvard Business Review*, vol. 84, no. 10, pp. 92–101, 2006.
[8] E. Elmaghraby and P. Keskinocak, “Dynamic pricing in the presence of inventory considerations,” *Management Science*, vol. 49, no. 10, pp. 1287–1309, 2003.
[9] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *Proc. European Conf. on*

Computer Vision (ECCV), 2014, pp. 584–599.
[10] Google LLC, "Google Gemini AI API Documentation," *Google AI Developer Documentation*. [Online]. Available: https://ai.google.dev
[11] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Wiley, 2020.
[12] Razorpay Software Pvt. Ltd., "Razorpay Payment Gateway API Documentation." [Online]. Available: https://razorpay.com/docs/api
[13] G. Biyalogorsky, E. Gerstner, and B. Libai, "Customer referral management: Optimal reward programs," *Marketing Science*, vol. 20, no. 1, pp. 82–95, 2001.