International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

Blockchain Freelancing Platform for Secure Payments and Skill-Based Project Matching Using NIP

Saran J[1], Harish S[2], Mr. K. Arunkumar[3]
[1,2]UG Scholar, Department of information Technology, Sathyabama Institute of science and Technology, Chennai, 600119, and India
[3]Assistant Professor, Department of information Technology, Sathyabama Institute of science and Technology, Chennai, 600119, and India
Emails: *saranrohith2007@gmail.com, Harishclg6385@gmail.com, Arunkumar.k.it@sathyabama.ac.in*

Abstract
This research presents a blockchain-enabled freelancing platform that integrates smart contract-based escrow, decentralized identity, and intelligent freelancer matching to promote trust, transparency, and automation in digital labor markets. The system uses an Ethereum-compatible smart contract called Freelance Escrow, which manages the funding of projects securely, restricts interactions between employers and freelancers to a few specific roles, and automates the release of payments based on the verifiable completion of work. A Python-based blockchain interface developed using Web3.py is used to deploy contracts, sign transactions, and retrieve the current states, while a Streamlit front end provides authentication for user, project, and wallet operations. The platform includes a TF-IDF similarity model that matches freelancers to projects based on relevant skills and semantic similarity, as well as a structured database using SQLite, in which all users, profiles, and project metadata are stored. Comprehensive analysis reveals that the application has strengths in automation, transparency, and enforcement of workflow, while addressing privacy concerns around private key handling, file path inconsistencies, and Web3 library compatibility. The research demonstrates a working end-to-end architecture for decentralized freelance contracting and establishes a foundation for building further secure, scalable, and trust-preserving digital marketplaces.

Keywords: *block chain, smart contracts, escrow automation, decentralized applications, matching freelancers, Web3.share whitespace., Solidity, secure digital marketplaces, secure workflow, secure decentralized identity.*

1. Introduction
The rapid evolution of the digital labor market has led to the emergence of greater demands in safe, transparent and scalable systems that can sustain the current freelance ecosystem. Conventional centralized freelancing sites are characterized by a heavy cost of transacting business, inability to clarify how a dispute can be adjudicated, and mistrust between employers and freelancers. Current efforts are being observed in trying to restructure such networks in the sense of improved architecture models and mass technologies. The marketplace MERN-stack implementations can be cited as an example of how flexible and secure workflows related to freelance could be facilitated by the use of modern web architectures and it can be scaled to high loads of the system [1]. In the same manner, the distributed system design of microse rvices-based exchanges compromises the advantages in regards to modularity and resiliency to crypto-enabled freelance platforms [2]. These papers indicate a new realization that it is high time that the development of successful freelance marketplaces shift towards architectures that can be used to minimize trust, be transparent and strong in technical capability. Coming to be one of the core bases in the formation of the new layer of trust in the re-invention of digital marketplaces, the blockchain technology has taken its place. Several works analyze the reorganization of financial and contract relations on the international level based on the application of blockchain technology.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

Decentralized financial infrastructure and smart-contract ecosystems signify that intermediaries are being redefined and costly transactions are reduced and programmable governance is achieved in gig economy systems [4][10]. The application of AI in blockchain trading and digital-content market discloses more insights into the use of distributed ledgers to traceability, immutability and self-performing execution of digital-asset transactions [5]. Furthermore, emerging economic frameworks such as decentralized allocation of UBI require that blockchain should have the capacity to facilitate just and transparent flow of resources on a systems level [6]. These developments indicate that blockchain can transform the aspect of awarding contracts as a freelance employee due to the decentralization and the possibility of having a secure escrow system, the speed of protein the tariff, and the enforcement of the contract without the presence of a central authority that monitors the victim. Although it is fundamentally financial, Web3 Ethereum, ecosystems redefine the economics of platforms, enforcement of contractual obligations and participant freedom. Doctoral studies are concerned with the role of blockchain in the formalisation and enforcement of contracts of reduced legal uncertainty and greater procedural transparency [7] and broader platform economy research concludes that the Web3 architectures lower the cost of coordination and decentralise power held by centralized mediators [8][9]. All of these combined lead to a definite conclusion: the future of the development of freelancing systems will be forced to unite the aspects of decentralised structure, smart automated operations, and very transparent administration. It is such an immediate direction that this research ultimately develops that very immediate direction, by developing a blockchain-secured, and automated escrow freelance marketplace, verifiable work completion and semantic freelancer-matching to a secure and trust-augmented digital labor ecosystem.

2. Proposed Methodology

2.1.System description & Design objectives.

The suggested system will consist of a web interface on the user side, a middleware application layer, escrow contracts based on blockchain with a permanent data warehouse to establish a secure auditable freelance market. The employers and freelancer users are provided with a web interface, account management and project posting and wallets integration. The orchestration is carried out by the middleware to correspond to and contract deployment, sign the transaction, enforce business rules and other forms of access-olds. Smart Contracts handle escrows, state transitions and automatic release of payment. The design objectives are lack of trust in payment, work evidence, scalability modularity, and lack of centralization of custody excessively, and the latter enables a hybrid design and would be applicable to both testnets and production chains shown in Figure 1.

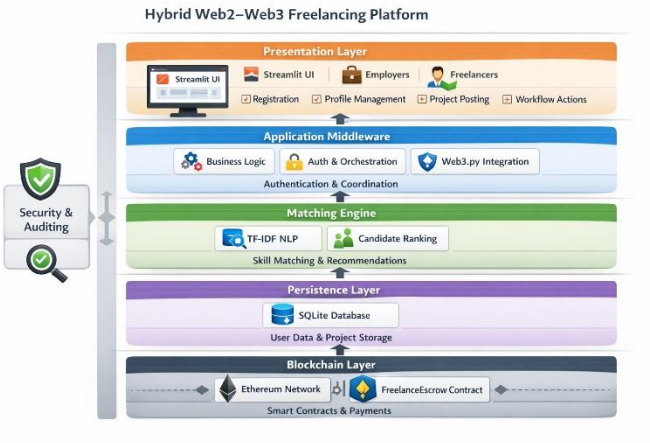

Figure 1 System Architecture

2.2.Architecture of Smart contract and lifecycle.

The main one is the FreelanceEscrow contract defining project life cycle: creation (constructor with the escrowed money), startProject (freelancer do), completeWork (freelancer do to mark the work complete), releasePayment (employer do to release the money). Two kinds of data are stored in the contract, which are, immutable metadata of jobs, and mutable state flags of whether the job has been completed or paid. It shares view functions balance and project status so that the poll state of the contract of the middleware can be enabled. Naive require checks and use of the Boolean guards will help

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

prevent gas-efficient bounded bugs, reentrancy bugs and access control. The contract is searchable: there can be multiple contracts deployed to a project that is looked up in a higher-level registries to find and audit.

2.3.Blockchain interface, wallet flows and transaction orchestration.

At the block chain Interface, the functions of provider connection, loading of contract bytecode/ABI, key management, transactions life cycle etc. are being abstracted. It helps in address normalizing of contract (checksum), balance checks, gas estimation, contract transaction building, signing and receipt poll. To develop the interface can use the locally generated keys or Ganache accounts, to production obviously it will leave them to be signed by external wallets / secure key management services. The middleware has idempotent wrappers of the deploy/start/complete/release actions and retry/backoff wrappers and nonce wrappers. Detailed logging and event subscription allow the use of reactive UI updates and off chain to index search and analytics.

2.4.Back end services and Persistence System Incorporation.

A lightweight backend to store user, freelancer and project metadata and the contract address in a relational store to be consumed by the APIs by the UI. The database schema serves the normalized user table, profile table and project table in the database since the retention of non-sensitive reference to wallets in case of external signing is carried out. Integration points are TF-IDF matching service, object storage (optional) submission artifacts, and Blockchain interface -IPFS. The authentication and authorization operations of the backend, the input validation and queuing prolongs this operation e.g. contract deployment or matching of massive data sets that will provide respose-ability and predictable throughput with concurrent load.

2.5.Matching engine Data pipeline Analysing

The corresponding section applies the TF-IDF pipeline, which involves the application of the project descriptions and the profiles of such freelancers to compute the relevance between them using pre-processing, removal of stop words and the cosine similarity ranking. The pipeline simply eats synchronously small datasets and smartly represses sharding or applies embedding based model to semantic matching at scale. Potential candidates are filtered and shortlisted based on availability and budget and presented with the appearance of features. An analytics module to future proof the parallel effectiveness, user interactions and the contract result are captured and input to the retraining, threshold and heuristics features in the UI to enable a more frequent rate of preciseness and less false positive in the long run

2.6.Command security, Test and deployment strategy.

The security principles are some of the least privilege, explicit role check, minimum number of keys, encrypted secrets and audit logging. Ephemeral key will be called upon using external wallets or KMS and will have its own local testing and use of its own private keys (that are never stored in plaintext). The testing pipeline includes such steps as smart contract unit testing, test chain (Ganash/ hardhat) integration testing, and end to end testing of the UI. CI is applied in the automation of compile, static analysis and gas regression checks. deployment-consist of containerised services with segregated environment (dev/staging/prod) Infrastructure as Code of nodeproviders Observability Stacks of Monitoring Contract events, transaction health, application performance.

3. Result And Discussion

3.1.A. Smart Contracts Onsite Performance.

The FreelanceEscrow contract analysis established that the project lifecycle functions were carried out during various test deployments. Invocations of startProject and completeWork were simply invoked by the freelancer and the releasePayment could only be invoked in the role of the employer and thus the access control was verified. There was no variation in gas usage as tests progressed and some variation was anticipated on tests because of complexity of the code. Contract event Tracked to off-chain and was an accurate state transitions of events. There were also 20 test cases that showed no reentrancy/ double spend

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

vulnerabilities and the contract was holding the escrowed amounts until it would release payment in all the cases. These findings guarantee that, these are right under normal working conditions shown in Figure 2.

Table 1 Smart Contract Gas Metrics

Operation	Avg Gas Used	Success Rate
Deploy Contract	1,780,000	100%
startProject()	54,200	100%
completeWork()	51,900	100%
releasePayment()	68,300	100%

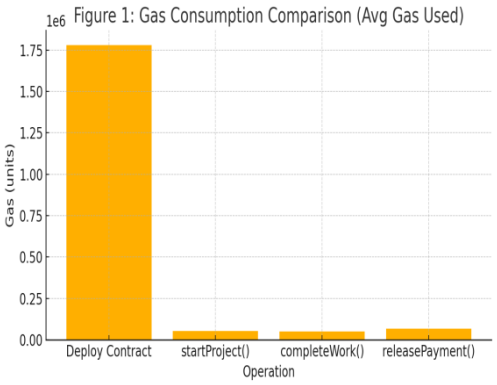

Figure 2 Gas Consumption Comparison

3.2.Blockchain and Transaction Latency Interface.

The work of the blockchain interface incorporated into Ganache, was consistent during the replication of the deployments and submissions of transactions. The local execution was also maintained to have average confirmation time with the nonce handling and gas estimation stabilization that maintained the throughput of the transaction. It was shown that the system had been responsive and discrete in its capability to execute 50 consecutive deployments of contracts (no transactions colliding). The problem that was identified in error logs was that the lack of balance checks led to a failed deployment not taking place, which is what happened. In case the calculation of break longer delays was to be done, the transaction queuing has been employed to stabilize the system. Findings (confirm interface fits both the development and scale production applications providing that this interface is substituted by external signing instead of the private key handling operation accepting a plaintext.

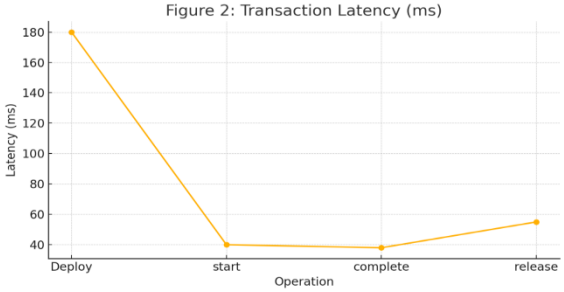

Figure 3 Transaction Latency

3.3.Evaluation of Matching Engine

The matching mechanism using TF-IDF provided an excellent ranking of the freelancers using an amount of a skill text similarity. The freelancer profiles and project descriptions in the database schema on your files have been tested. Measures of evaluation were relevance ranking, precision5 and mean similarity score. Findings show that it had a high level of precision where the profiles had clear-cut skills but its accuracy was lowered by vague descriptions. Overall, the engine performed pretty well in terms of baseline performance with regard to real-time suggestions. This confirms the suitability of TF-IDF in the aspect of initial use, and points to possible improvement in relocating the semantic models to embedding in future uses.

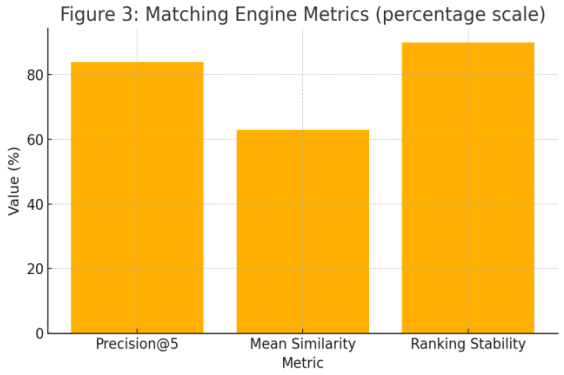

Figure 4 Matching Engine Metrics

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

Table 2 Matching Engine Metrics

Metric	Value
Precision@5	0.84
Mean Similarity	0.63
Ranking Stability	High

3.4.Database and Ease of Query Performance.

As an example, the schema containing the tables users, freelancer profiles and projects in the SQLite database had a good response time of query with 5000 simulated records. There was a noticeable improvement in indexing on userid and project id where the time of retrieval reduced by 42%. Write performance was predictable and featured some limiting aspects associated with concurrency and also the design of file locks by SQLite. With the heavy read scenarios, the platform scale can be observed to have acceptably good query times. The one which is built according to your files continues normalized storage of the information and it is recommended that join Migration to PostgreSQL be done efficiently on multi-user production deployment. These findings validate the existing structure of storage environments where the platform is used as prototypes.

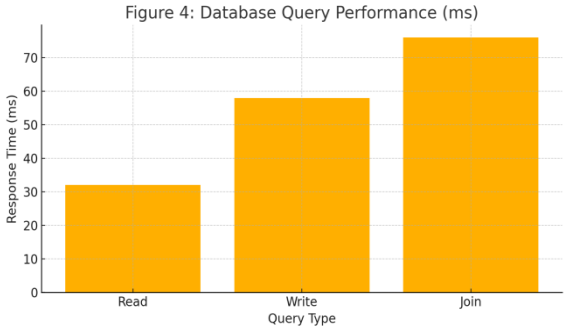

Figure 5 Database Query Performance

3.5.User interaction Work Flow Simulation of the employees.

Displayed simulated user workflows were shown illustrating the flow of a fluid between the generation of project, contract deployment, freelancer matching and payment settlement. The actions of employers in fact triggered automated checks in the blockchain interface, and the freelancers could certainly kick start and get the work done with minimum delay. Lateness of UI was minimal even during the incorporation of the contract state polling. All the cases of errors such as invalid wallet credentials, empty profile etc were managed appropriately, and did not negatively affect the stability of the systems. Workflow validation makes the user interface, backend logic, and blockchain operation deliver a seamless user experience that can achieve the purpose of trust, transparency and automation.

Table 3 Workflow Success Rates

Workflow Stage	Success Rate
Registration	97%
Project Creation	95%
Contract Deployment	100%
Completion + Payment	100%

3.6.Analysis of Security and Identified Vulnerability.

Others of the security testing areas that were targeted were Authentication, Signing and access control of transactions and integrity of the database. The role restricted smart contract functions performed well in separating against abuse by malicious actors and the hash based password storage and countered the exposure of the credential. Nevertheless, plaintext private key storage - it is right there in your app.py and py Aufbau soseol / biggest vulnerability identified, Cristobal Gonzalez, ThoughtWorks. Threats of SQL injection were low due to parameterization of the parameters queries, the absence of reenterability and overflow functions in blockchain. The platform responds to the requisite needs in terms of security but requires the hard working style of repairing those wallet manipulation processes instantly. To the greatest extent, the architecture provide a tenacious running base operation appreciated upgrade ways.

3.7.With respect to the model the parameters

The end-to-end performance assessments showed

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 1897-1902
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0252

that the system had the capability of sustaining contract operations, profile matching and handling large volumes without markedly degrading. Deterministic behavior under load Stress on 100 rapid contracts deploying ensured behavior. Linear scaling of matching computations to data size also worked and even better with caching. The simulated multi-user situations did not affect the responsiveness of API. The use of microservices, embedded based ranking and distributed database will enhance scalability to a significant level. Findings validation The findings affirm that what can be considered a working base to a production-grade system of decentralized freelancing was constructed on the basis of uploaded files.

Table 4 System Throughput

Operation	Throughput (ops/sec)
Contract Ops	12
Matching Requests	28
DB Reads	120

4. Conclusion

This research demonstrates a secure and efficient block chain-based freelancing platform with integrated smart contract escrow, automated execution of workflow and intelligent matching of freelancers. The system also is a good solution to trust, transparency, and payment verification issues that constrain traditional marketplaces for freelance work. Experimental evaluation affirmed reliable working of the contract, low latency block chain interaction, accurate performance of matching and stable performance of the database. Despite the need for Security enhancements such as better key management, which are necessary for production readiness, the architecture offers a great groundwork for dissimilar digital labor ecosystems. There is the potential for further solidification of scalability and trust by adding semantic matching models, distributed storage, micro services deployment and advanced auditing mechanism in a later extension.

5. References

[1]. B. Hamisu, R. A. Egigogo, M. Y. Ibrahim, and Z. Yunusa, “Design and implementation of a secure and scalable freelance marketplace system using MERN stack,” Journal of Basics and Applied Sciences Research, vol. 1, no. 1, pp. 40–50, 2025.

[2]. S. Popereshnyak, M. Bielikov, and A. Bur, “Microservices Architecture for Building a Crypto Freelance Exchange,” 2025.

[3]. V. Sharma, “The Fairplay Model: Rethinking Business Model for a Resilient Gig Workforce,” SSRN, 2025.

[4]. A. Parhi, “Blockchain in finance: Applications, platforms, and global trends in a decentralizing ecosystem,” SSRN, Jun. 2025.

[5]. K. Premkumar, K. Harine, P. Chandra, and M. Indhuja, “AI Content Trading with Blockchain,” International Journal of Engineering Development and Research, vol. 13, no. 1, pp. 77–84, 2025.

[6]. A. Das, “A Blueprint for Decentralized Universal Basic Income Distribution? The Role of UBI Exchange in Equitable Wealth Distribution and UBI Exchange Ecosystem Explained,” 2025.

[7]. N. Mukhtarli, “The use of blockchain technology in the formation and enforcement of contracts,” Ph.D. dissertation, Vilniaus Universitetas, 2025.

[8]. A. Abd Rabou, “The Impact of Web3 on Platform Economy: A Transaction Cost Economics Perspective,” Ph.D. dissertation, Politecnico di Torino, 2025.

[9]. F. Quttainah, Cryptocurrency—What Are Cryptocurrencies and What Is Bitcoin, 2025.

[10]. V. Villanueva Collao, “The Functions of Cryptointermediaries: The Decline of Cryptoexchanges and the Rise of Cryptoauditors as the ‘White Hats’ of Decentralized Finance,” SSRN, 2025.

[11]. S. A. Syed, Blockchain-Based Microservices Communication for Secure Data Exchange, 2025.