Grenze International Journal of Engineering and Technology, June Issue

# CampusOrbit – An Intelligent Decentralized Placement ERP using Hash-based Integrity Verification, AI Shortlisting, and NFT Credentials

Anushka Dondal[1], Aparna Joshi[2], Arpita Gaikwad[3], Rutwika Ganer[4] and Shreya Gunjkar[5]
[1-5]Department of Computer Engineering Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India
Email: anushkadondal37397@gmail.com, aparna.joshi@pccoepune.org, arpitagaikwad93@gmail.com, rutwikaganer25@gmail.com, gunjkarshreya08@gmail.com

***Abstract*— Managing campus placements in higher education institutions always depends on scattered, unclear and easily tamperable system like paperwork and basic spreadsheets. CampusOrbit is a complete placement ERP system that solves this problem by combining all things under a single platform, by using secured verification methods, AI based candidate analysis, blockchain backed records, NFT credentials and smart resume and skill gap analysis. This system ensures data security by converting records into fixed JSON hashes and linking them through a Merkle tree on blockchain and making every placement activity trackable and tamper proof. This application uses Gemini 2.5 Flash rank candidates from their skills and CGPA and also giving students personalized feedback on their resumes and suggest areas to improve their skills. The Placement records are issued as NFT tokens which give students a permanent and a vali proof of their selection. This system also has RBAC (Role based access control) in which particular users can perform certain roles which ensures that students, TPOs and admin can only perform the task assigned to them. The tested results clearly show improvements in how quickly candidates are shortlisted and how securely data is stored and handled and how transparent the overall process is as compared to traditional systems. This paper explains the overall architecture, working methodology add the design of CampusOrbit with the special focus on its layered security features and the use of AI in the system.**

***Index Terms*— Blockchain, Placement ERP, NFT Credentials, AI Shortlisting, Integrity Verification, Merkle Tree, IPFS, Gemini AI, Decentralized Systems.**

## I. INTRODUCTION

Campus Placements is a crucial phase in every student's academic journey, where even small error in data or communication can lead to serious consequences on careers and the institution's credibility. Despite this many colleges still depend on outdated and traditional methods such as spreadsheets, emails, etc and manual shortlisting. These approaches are very time consuming and can also lead to accidental data loss, unauthorized edits and nonreliable tracking. In many cases, the students are left without any solid proofs of their placement status and when the conflicts arise regarding result or job offer there is no clear and trusted way to resolve them.

To overcome all these challenges, CampusOrbit introduces a modern and blockchain based Placement ERP system that reshapes the entire placement process and make it more clear and simpler.

*Grenze ID: 01.GIJET.12.2.183_26*

It brings together all stages of recruitment into a single streamlined platform from creating drives, checking eligibility, AI powered shortlisting, managing multiple selection rounds, handling disputes and issuing placement credentials. Each change made in the record is securely converted into a cryptographic hash and is digitally signed using the TPO's wallet and then recorded on public blockchain using a Merkle root structure. This approach ensures that every action is recorded permanently and can be verified at any time.

This system also uses Google Gemini 2.5-Flash to make the process smarter and faster. It evaluates candidates for specific drives using a factors like skills, academic performance, branch which helps recruiters to shortlist students more effectively. Also, it provides students with detailed resume and skill gaps analysis which points out gaps in their skillset and offer practical suggestions for improvement. This dual use of AI increases the efficiency of placement activities and also give students meaningful guidance for their preparation. Overall, CampusOrbit aims to create a transparent, secure and efficient placement ecosystem.

## II. Literature Survey

The centralized database is used by most of the academic and placement systems these days. These systems are easy to use and manage but they have a major problem which is, the data can be changed without proper proof. This becomes a serious issue during placements, where decisions like eligibility, shortlisting and final selection directly affect students' careers.

Rustemi et al. [1] studied different blockchain-based verification systems and found that many of them do not maintain proper audit trails. Because of this, if any record is changed later, it becomes very difficult to detect. Another important gap is that most systems only verify data after it is finalized but the data is not protected while processing when most placement decisions happen.

Tariq et al. [2] proposed a system called Cerberus. In that system, records are converted into hashes and stored on the blockchain. This makes verification simple and secure. However, the system assumes that records are already finalized before being stored hence it does not protect changes that are during the process. Similarly, a system using blockchain with IPFS [3] showed that storing only the hash on-chain and keeping full data off-chain is efficient but it is not useful for handling the situations where the same record is updated multiple times. NFT-based systems [4][5] also show that certificates or offers can be issued as tamper-proof digital tokens but they focus mainly on the final output rather than the process. They do not cover the whole placement process like eligibility checks, criteria, shortlisting, interview rounds.

CampusOrbit system improves this by securing the entire placement process rather than only focusing on the final result. Every time a record is created or updated, it is converted into a hash which is signed by the TPO and linked to the previous version. This creates a continuous chain of records. So when the final NFT offer is issued, it is backed by a complete and verifiable history of all decisions that have been made.

The system has to protect the history of all the changes. Ali et al. [8] suggested that even authorised users like administrators can misuse the access if security logging system is not available. Their work highlights the importance of tamper-proof logs but it deals with systems in which data does not change frequently. In placement systems, records keep changing across multiple stages like applications, shortlisting, interviews and results. Román-Martínez et al. [9] created a system in which every action is recorded on the blockchain but it requires users to directly interact with blockchain technology. It's not practical for our system.

Traditional systems store all student data in a centralised system which raises privacy concerns. Dieye et al. [5] proposed a self-sovereign identity model where users control their own data but such systems are complex to implement. A survey on decentralized identity [12] shows that modern systems often use IPFS with blockchain verification but they still lack a proper way to handle disputes if something goes wrong.

Can et al. [6] proposed a hybrid system which combines the blockchain with traditional database but their focus was on healthcare and does not fully address the placement situation. Dallel et al. [7] also pointed out that relying only on access control is not enough. System must be able to detect the tampering but they did not provide a solution for resolving such issues.

CampusOrbit addresses this issue by adding a feature named dispute resolution system. If a student feels a record is not correct, they can raise a dispute by providing hash-based proof. The issue is then reviewed by the TPO and admin and the final decision is recorded securely. This creates accountability and ensures fairness.

Another challenge is about deciding how much blockchain should be used in a system. Alghamdi et al. [8] found that fully blockchain-based systems are very slow and not scalable for real-world use. Da Costa et al. [9] also showed that blockchain improves security but it is not practically possible to store everything on-chain. Similarly, other frameworks [13] focus only on verification and not on managing the entire system.

CambusOrbit solves these issues by using a normal database (MongoDB) for storing data and blockchain is used only for verification through hashing and Merkle roots. This gives strong security without slowing down the system. CampusOrbit is a complete placement platform and not just a verification tool.
Lastly, AI is becoming an important part of placement systems. Research on AI-based resume screening [10][11] shows that machine learning can help shortlist candidates but these systems often lack transparency. Sometimes Students cannot clearly understand why certain decisions are made. CampusOrbit improves this by providing detailed skill gap analysis and giving the resume score so that students can see exactly why a candidate was not shortlisted or rejected. This makes the CampusOrbit system more transparent.

## III. Proposed System

CampusOrbit is a blockchain-enabled placement management ERP developed to make the campus recruitment process more secure, transparent and efficient. The system provides a single platform where students, TPOs, companies and administrators can manage all placement related activities digitally.
In the proposed system, companies and TPOs can create placement drives, define eligibility criteria, manage interview rounds and track student applications. Students can register, upload their profiles and resumes, apply for eligible drives, monitor their application status and receive placement offers through the platform.
For ensuring data integrity, every important record like applications, interview results and selection decisions is converted into a cryptographic hash and stored in an integrity chain. These hashes are further organized into a Merkle Tree and anchored on the blockchain, making the records tamper-evident and verifiable. If any record is modified without authorization, the system can immediately detect the change through its verification mechanism.
The system also integrates AI to automatically rank and shortlist candidates based on their skills, academic performance, achievements and job requirements. Additionally, selected students are issued NFT-based digital placement credentials that can be publicly verified and cannot be forged.
An audit logging mechanism maintains a complete history of all critical actions, also a dispute resolution module allows students to raise concerns regarding placement records and ensures transparent handling of disputes.

## IV. System Architecture

Fig. 1 Shows the layered architecture pattern of the system flow. Frontend and Backend communicate through authenticated HTTP requests that exchange data in JSON format. On the server side, each protected request passes through a middleware pipeline. The process begins with verifying the JWT token to confirm the user's identity. After authentication, the system checks whether the user's role is allowed to access the requested route. When both checks succeed, the request reach the controller that performs the main operation.
All blockchain-related functionality in the system is handled using ChainCraft. It acts as a blockchain abstraction layer. It provides services for SHA-256 hashing, wallet-based signing, Merkle tree generation, blockchain anchoring, IPFS storage, NFT minting and governance proposals.

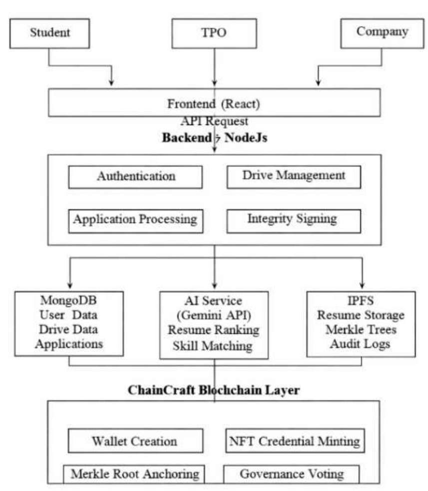


Fig. 1 System Architecture

Features:

*A. Multi-Layer Integrity System*

Data integrity is insured by 3 layers:
Layer 1 – Per-Record Hashing (Immediate Protection):
Whenever any changes are made in the record, the system creates the security hash of that data. Before hashing the data is organized in the proper format to ensure same hash is produced for same record.
This hash is then:

- Signed using the TPO's blockchain wallet
- Stored with a timestamp and signer details
- Added to an integrityChain[] inside the record

Over the time chain of changes are built, making easy to trace every modification.
Layer 2 – Merkle Tree Anchoring (When a Drive Ends):
When the drive completes, all record hashes are grouped into a Markel Tree. The root of this tree acts as a single proof of all record. It is stored in a public blockchain using ChainCraft. The complete tree is saved on IPFS.
An integrity anchor record keeps the track of Marel root, Blockchain transaction ID and IPFS links to ensure that large sets of data are securely verified.
Layer 3 – 4-Step Public Verification:
Anyone can verify whether the record is genuine or not.

- Hash Check – Recalculate the hash and compare it with the stored one
- Merkle Verification – Check if the record exists in the Merkle Tree
- Blockchain Check – Checks the Merkle root is stored on-chain
- Signature Verification – Validate the TPO's digital signature

If all these checks are passed the record is VERIFIED otherwise TAMPERED.

*B. AI Features (Gemini 2.5-Flash)*

The system uses AI to make the process smarter and more efficient.

*AI-Based Shortlisting:*
TOP and companies can automatically view shortlisted candidates for a drive.
The AI assurance:

- Job requirements (role, skills, CTC, eligibility)
- Student profiles (CGPA, skills, achievements, branch)

It then provides:

- A ranked list of candidates
- Scores (0–100)
- Reasons for selection
- Missing skills for each student

Scoring Criteria is:

- Skills Match – 40%
- Achievements – 25%
- CGPA – 20%
- Branch Relevance – 15%

*Resume & Skill Gap Analysis:*
Students can upload their resume and AI generates the feedback report.
The system assures:

- Current skills
- Industry expectations
- Target career domain

It then gives:

- Skill gap analysis
- Suggestions (courses, projects, certifications)
- ATS score

This helps the students to understand what and where they have to improve.

### C. NFT Placement Credentials

When the students are placed, the system generates the digital certificate in the form of NFT(ERC-721).

### D. Dispute Resolution

Whenever the student faces any difficulty while accessing the system, they can raise dispute. The system uses transparent blockchain process.

- A dispute creates an on-chain proposal
- TPOs and admins vote on it
- All votes are permanently recorded
- A resolver finalizes the decision

The outcome is stored in the database and blockchain. This ensures that the dispute is handled fairly.

### E. Technology Stack Summary

TABLE I. TECH STACK

| Layer | Technology | Purpose |
|---|---|---|
| Frontend | React 19, Vite, Tailwind CSS v4 | SPA, state management, styling |
| Backend | Express 5, MongoDB, Mongoose | REST API, data persistence |
| Auth | JWT (15m/7d), bcrypt.js salt-12 | Secure access and session management |
| Blockchain | ChainCraft API | Hashing, signing, anchoring, NFT, governance |
| Storage | IPFS via ChainCraft | Resumes, Merkle trees, audit logs |
| AI | Google Gemini 2.5-Flash | Candidate ranking, resume & skill-gap analysis |
| Forms/Validation | React Hook Form, Zod | Client and server schema validation |

## V. METHODOLOGY

The development of the System follows a structured approach that focuses on building a complete placement automation platform and strengthening it with tamper evident integrity layers.

The first step was studying how placement workflow operates in real environment and identifying the drawbacks in traditional system. In traditional ERP systems important data such as eligibility results, shortlists and final selection can be edited without any reliable proof, leading to data tamper and unreliable reports.

The next step was to develop system architecture, identifying roles, functionalities, system workflow and tech stack. The frontend section was developed using React.js to create role specific access for Students, TPO, Companies. The backend section was designed using Node.js and Express.js which includes REST APIs for authorization, authentication and business logic. For database storage MongoDB Atlas was used. JWT authentication and RBAC were used to ensure secure and role-based system access.

Next, the tamper evident integrity layer was introduced. Critical records such as job posting, applications, status were considered ad high risk record. Every time when the record was created or modified, the system generates a SHA256 hash. It maintains the data consistency.

The hash generated was stored in an integrity log along with the metadata record type, record ID, user identity, action performed and timestamp. This mechanism is called as hash chaining. This mechanism linked every new entry with the previous entry, forming a continuous sequence like in the blockchain logs. If any unauthorized modification is identified the sequence chain breaks to ensure data integrity.

For verification system recomputes the hash and compares it with the previously stored value. If the hash matches, the record is considered as valid record otherwise tampered.

### A. System Representation

Let the system be defined as:

$$S= \{U, R, A, DB, H, C, V\}$$

Where:

- U = Set of users (students, TPO/admin)
- R = Set of placement records
- A = Actions (create, update, view, verify)
- DB = Database storing operational records
- H = Hash generation function (SHA-256)
- C = Hash chain mechanism

- V = Verification function

Let:

$$U = \{u_1, u_2, \dots, u_n\}$$

Each user has a role:

$$\mathrm{role}(\mathrm{u_i}) \in Student, TPO$$

Access permission function:

$$Perm(u_i, a_j) = \begin{cases} 1, & if\ role\ allows\ action \\ 0, & otherwise \end{cases}$$

*B. System Representation*

Let the system be defined as:

$$S = \{U, R, A, DB, H, C, V\}$$

Where:

- U = Set of users (students, TPO/admin)
- R = Set of placement records
- A = Set of actions (create, update, view, verify)
- DB = Database storing operational records
- H = Hash generation function (SHA-256)
- C = Hash chain mechanism
- V = Verification function

Let:

$$U = \{u_1, u_2, \dots, u_n\}$$

Each user has a role:

$$role(u_i) \in Student, TPO$$

Access permission function:

$$Perm(u_i, a_j) = \begin{cases} 1, & if\ role\ allows\ action \\ 0, & otherwise \end{cases}$$

*C. Record Model*

Let placement records be:

$$R = \{r_1, r_2, \dots, r_m\}$$

Each record:

$$r_i = (data_i, t_i, u_i)$$

Where:

- $data_i$ = record content
- $t_i$ = timestamp
- $u_i$= user who performed action

*D. Hash Generation Model*

For every record creation/update:

$$h_i = H(data_i)$$

Where:

$$H : data \rightarrow \{0,1\}^{256}$$

*E. Hash Chaining Model*

Each log entry links with the previous hash:

$$c_i = H(h_i | c_{i-1})$$

Where:
- $c_i$ = current chain hash
- $c_{i-1}$= previous chain hash
- |= concatenation operator

*F. Integrity Verification Model*

For verification:

$$V(r_i) = \begin{cases} Valid, & if\ H(data_i) = h_i\ and\ chain\ intact \\ Tampered, & otherwise \end{cases}$$

*G. Security Constraint Model*

Unauthorized modification detection:

$$Tampered(r_i) \Rightarrow H(data_i') \neq h_i$$

Where:
$data_i'$= modified

## VI. RESULTS

*A. Comparative performance*

The system was compared with a normal spreadsheet-based placement system used in a college. Key metrics measured include shortlisting time, dispute resolution time, credential verifiability, Student Transparency Score and Resume Feedback Turnaround.

*B. AI Shortlisting Accuracy*

In a test conducted on 3 placement drives, there was total 147 students, the AI-generated shortlist was compared with TPO experts ranking. The results were showing high similarity between expert rankings with correlation score of 0.87, that means the AI is making similar decision very close to TPO experts with high accuracy.
The company HR rechecked the candidates resume who were marked by AI as having some missing skills for that particular job and then HR agreed with AI with 91 % of time. This shows that AI is closely accurate in identifying skill gaps in candidates with less time.
Additionally, this system was able to process and generate results for more than 50 applicants in around 9.4 seconds, which shows that it is also fast and efficient to use.

*C. Integrity Verification Results*

To check the tampering is working in the system, we checked across 6 simulated placement drives with total of 220 application records. We observed that there were no false positives, meaning no valid records were wrongly marked as Tampered. To check the system, we tested it by controlled tampering experiment where one field in a MongoDB document was manually changed. As a result, the system successfully detected the change as TAMPERED in all 15 test cases during step 1 (Hash Check).
To generate Merkle proofs, the time taken was around 340 milliseconds for drives for up to 50 applications.
The blockchain anchoring process using the ChainCraft oracle took an average of 4.2 seconds per transaction and it demonstrated that the system works reliably and within a reasonable time.

*D. System Output Illustrations*

TABLE II. PLACEMENT FUNNEL – DRIVE OUTCOMES BY STAGE (SAMPLE DATA, N=147)

| Stage | CSE | IT | ECE | MECH | Total |
|---|---|---|---|---|---|
| Applied | 48 | 32 | 27 | 40 | 147 |
| Eligible | 45 | 30 | 24 | 35 | 134 |
| AI Shortlisted | 22 | 15 | 12 | 10 | 59 |
| TPO Shortlisted | 18 | 12 | 10 | 8 | 48 |
| In Rounds | 18 | 12 | 10 | 8 | 48 |
| Selected (Offer) | 11 | 8 | 6 | 4 | 29 |
| NFT Minted | 11 | 8 | 6 | 4 | 29 |

Table II. below (conceptual representation) illustrates the placement funnel analytics dashboard available to TPOs it is showing the progression from applicants to offers across branches:

*E. Application Screenshots (Description)*

The following describes key screens of the CampusOrbit application:

*Student Dashboard*

Fig. 2 shows student dashboard, student can see jobs available, total applied jobs, how many in progress jobs, and selected. In Open drives, under My Application student can view for which company they have applied. Candidates can fill the profile data and then apply for open drive.

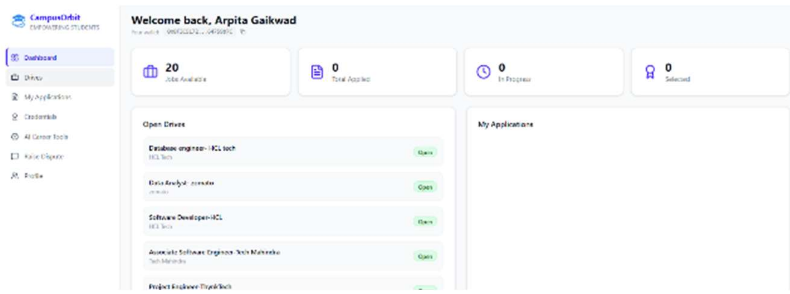


Fig. 2 Student Dashboard

*TPO Drive Detail & Integrity Dashboard*

Fig. 3 TPO DRIVE detail is shown, TPO can create drive, they have to fill given data fields such as Drive, title, company name with minimum to maximum salary in LPA, skills required for this job, job description. Also, TPO add last registration date and drive date. For eligibility, TPO add minimum CGPA and backlog data of students.

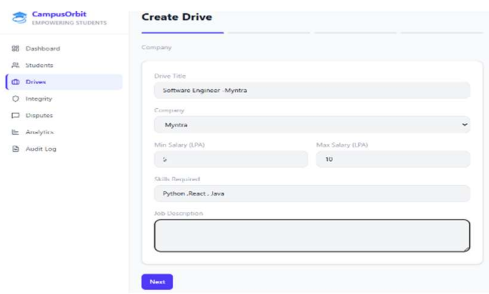


Fig. 3 Drive Creation

Fig. 4 shows a live integrity health timeline—each application row displays its hash, last signed timestamp, and verification status. It shows if there is any tampering in database or student data. In backend, 'Trigger Anchor' button initiates the Merkle tree pipeline. After anchoring, the drive card shows the Merkle root, IPFS CIDs for the audit logs or tree and the blockchain transaction hash as a clickable explorer link.


Fig. 4 Audit Logs

*AI Shortlist Panel (Company/TPO)*

Fig. 5 shows the List pf candidates ranked by AI score of each candidate who applied for the drive to company dashboard. By using Gemini API, the system generates these AI score to make the process of hiring the candidates faster. As shown in image it is showing AI score percentage of each candidate under different drives with missing skills.

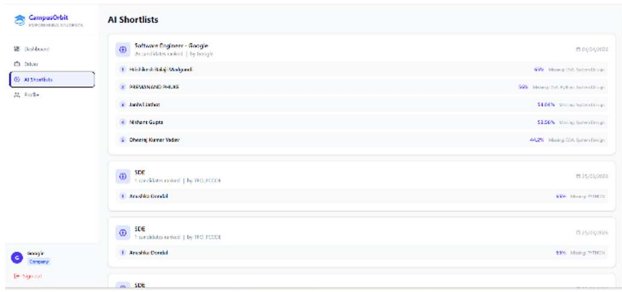


Fig. 5 AI Shortlist Panel

*Student's Skill Gap Analysis:*
Fig. 6 illustrate student side dashboard. After uploading a resume, the student accesses a Gemini-generated report card. Sections include: Identified Skills (tags), Skill Gaps (relative to target role), Suggested Courses/Certifications (with priority labels), Project Ideas to bridge gaps and an ATS Score with actionable formatting feedback. The report is exportable as PDF. It gives detailed of what are your strong skills, missing skills.

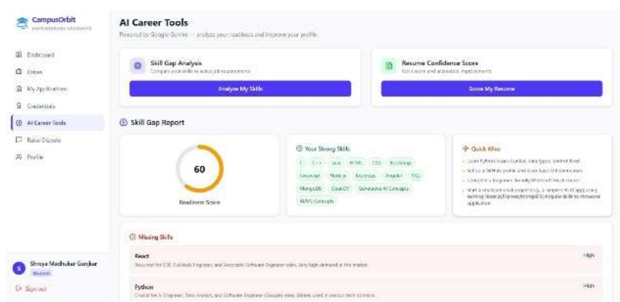


Fig. 6 Skill Gap Analysis

## VII. Conclusion

CampusOrbit shows that a placement system can combine various features like security, blockchain and AI. It has a multilayer integrity system which is helpful in detecting any changes made in data hence it is more reliable than the traditional system. The feature named AI shortlisting reduces the work of placement officer and makes fair selections. The resume score and skill gap analysis features help students to improve their skills.
The system is not only used for managing placements but also used for career growth.
NFT credentials are a secure and portable proof of placement that can be verified anytime and anywhere. The dispute system is a proper and accountable process which is transparent than most of the tradition and informal methods. Role-based access controls the permissions of each user according to their role. The system is secured using JWT authentication, encrypted keys and proper validations.

## Future Scope

Multiple College Placement Network: CampusOrbit can be expanded such that multiple colleges can use the system. Companies can post jobs which are visible to all the college. The data of each college is controlled by their own system.
AI Mock interviews- Using this feature students can practice for the interviews with the help of AI. They can get quick feedback regarding their performance and tips for improvement.
Predictive Placement Analysis- The past placement data can be used to predict a student's chance of getting placed which helps the placement officer to guide students better.
Smart Contract Automation: Use the blockchain smart contracts directly for verification and authentication which will reduce the dependency on external services.
Anomaly Detection System: Add a feature which detects unusual activities such as sudden changes in records or multiple updates in a short period of time which will help to identify possible errors or misuse.

## REFERENCES


[1] F. Rustemi, F. Dalipi, V. Atanasovski, and A. Risteski, “A systematic literature review on blockchain-based systems for academic certificate verification,” IEEE Access, 2023.

[2] Tariq, H. B. Haq, and S. T. Ali, “Cerberus: A blockchain-based accreditation and degree verification system,” IEEE Transactions on Computational Social Systems, vol. 10, no. 4, Dec. 2023.

[3] Ali, Y. Yasinghadi, T. J. Alahmadi, H. K. Alkahtani, and A. Raza, “Blockchain-based logging to defeat malicious insiders: The case of remote health monitoring systems,” IEEE Access, 2023.

[4] Román-Martínez, J. Calvillo-Arbizu, V. J. Mayor-Gallego, G. Madinabeitia-Luque, A. J. Estepa-Alonso, and R. M. Estepa-Alonso, “Blockchain-based service-oriented architecture for consent management, access control, and auditing,” IEEE Access, 2023.

[5] M. Dieye, P. Valiorgue, J.-P. Gelas, and E.-H. Diallo, “A self-sovereign identity based on zero-knowledge proof and blockchain,” IEEE Access, 2023.

[6] O. Can, T. Dağ, and M. Kantarcioglu, “A blockchain-based hybrid architecture for auditable consent management,” IEEE Access, 2024.

[7] O. Dallel, S. Ben Ayed, and J. Bel Hadj Tahar, “Blockchain-based authorization mechanism for educational social internet of things,” IEEE Access, 2024.

[8] T. A. Alghamdi, R. Khalid, and N. Javaid, “A survey of blockchain-based systems: Scalability issues and solutions, applications and future challenges,” IEEE Access, 2024.

[9] L. Da Costa, B. Pinheiro, W. Cordeiro, R. Araújo, and A. J. G. Abelem, “Sec-health: A blockchain-based protocol for securing health records,” IEEE Access, 2023.

[10] P. Gupta and N. Agarwal, “AI-Based Resume Screening Using Natural Language Processing and Machine Learning,” in Proc. IEEE Int. Conf. Artificial Intelligence and Smart Systems (ICAIS), 2024, pp. 220–226.

[11] R. Mehta and S. Kulkarni, “Automated Resume Shortlisting Using NLP Techniques,” in Proc. IEEE Int. Conf. Data Science and Advanced Analytics (DSAA), 2024, pp. 98–105.

[12] C. Mazzocca, A. Acar, S. Uluagac, R. Montanari, P. Bellavista, and M. Conti, “A Survey on Decentralized Identifiers and Verifiable Credentials,” IEEE Communications Surveys & Tutorials, vol. 27, no. 6, 2025.

[13] T. Nguyen and M. Park, “Blockchain-Based Academic Credential Framework for Secure Verification,” in Proc. Int. Conf. Information Systems Security, Springer, 2022, pp. 55–67.