I.J. Modern Education and Computer Science, **2025, 6, 76-96**
Published Online on December 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.06.06

A Structured Project-Based Learning Pedagogy to Bridge the Theory–Practical Gap in Blockchain Education

Rachana Yogesh Patil*
Department of Computer Enginerring, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India
Email: rachana.patil@pccoepune.org
ORCID iD: https://orcid.org/0000-0001-9973-6998
*Corresponding Author

Rahul Kulkarni
Co-founder and Chief Technologist at DoNew Innovations
Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India
Email: rahul@godonew.com
ORCID iD: https://orcid.org/0009-0007-0462-504X

Received: 11 November 2024; Revised: 19 January, 2025; Accepted: 27 May, 2025; Published: 08 December, 2025

Abstract: Traditional educational institutions prioritize theoretical education over hands-on practical skills which produce a gap between classroom learning and industry requirements particularly in the fast-growing blockchain sector. A structured case study demonstrates how Project-Based Learning (PBL) was implemented in an undergraduate engineering course which focused on blockchain technology. The educational approach evolved through four stages that combined theoretical instruction with collaborative solution creation and DApp programming and assessment evaluation. Performance metrics from students including testing coverage, GitHub contributions, documentation quality and research paper output are carefully analyzed through algorithmic guidance of each phase. The paper demonstrates the development of teaching methods through traditional practices and outcome-based instruction up to project-based learning supported by visual timeline comparisons. Student feedback demonstrates that the education methods led to enhanced technical abilities together with teamworking and increased student confidence. The case study demonstrates how PBL functions as an educational connection between academic learning and practical blockchain development needs because most teams (over 75%) finished functional DApps alongside several groups producing research suitable for publication.

Index Terms: Blockchain Education, Project-Based Learning (PBL), DApp Development, Smart Contracts, Outcome-Based Education, Engineering Pedagogy

1. Introduction

The quick advancement of digital technologies demands a fundamental educational transformation which moves engineering education from traditional content teaching methods toward experiential skill-based approaches [1]. Blockchain technology requires engineering students to develop both theoretical knowledge and hands-on abilities in DApp creation and deployment and testing [2]. Traditional educational methods that rely on lectures and textbooks and summative evaluations fail to provide students with the complex set of multidisciplinary abilities needed to handle such technological systems [3,4].

The decentralized infrastructure of blockchain supports numerous industries through its trustless operations in finance and supply chain management and identity control and healthcare delivery systems [5]. Modern blockchain courses at universities mainly concentrate on cryptographic principles and consensus algorithms but provide minimal opportunities for students to design and implement blockchain systems [6]. The educational content delivered in academic settings remains distant from what business organizations need through their workforce.

PBL has proven successful as a pedagogical framework which combines real-world application with problem-solving and collaboration through iterative design. Traditional teaching approaches stand contrast to PBL which gives students control to both discover problems and solve them through technical designs and critical assessment of their learning process [7]. Researchers show PBL creates more engaged learners who develop deeper comprehension along with transferable competencies such as teamwork and innovation and communication abilities.

This research documents the design process and evaluation of a blockchain-based PBL activity which was implemented in an undergraduate computer engineering program. The students built full-stack DApps with industry-standard tools through a phased methodology which consisted of conceptual clarification and design development followed by demonstration. The activity delivered comprehensive learning benefits and functioned as a venue for technical guidance alongside research investigation and collaborative student learning.

To analyze how students engaged with the project-based blockchain activity as self-regulated learners, we apply the COPES model by Winne and Hadwin [29]. This model frames learning as a process involving five interrelated components: Conditions (task features and learner characteristics), Operations (cognitive strategies), Products (outputs of learning), Evaluations (self-monitoring of progress), and Standards (criteria learners use to judge success). We revisit this framework in the discussion section to interpret how students navigated the learning environment, structured their problem-solving, and responded to the challenges posed by decentralized application development.

The research aims to achieve three main objectives:

1. Develops a scalable method for teaching blockchain through PBL education delivery.
2. The activity's technical and pedagogical results evaluation.
3. Examine student feedback together with reflective data to improve the curriculum in a continuous manner.

The rest of the paper is structured as follows: Section 2 presents a review of the relevant literature on PBL and blockchain education. Section 3 discusses the evolution of teaching-learning approaches leading up to project-based models. Section 4 details the proposed methodology and phased implementation. Section 5 summarizes the key observations, assessment findings, and student reflections. Finally, Section 6 concludes the paper with insights and future directions.

2. Literature Review

Engineering education now operates through experiential and student-centered methods after transitioning away from traditional knowledge-delivery models. The educational transformation results from both pedagogical research progress and the increasing focus on practical skill development and real-world problem resolution [8].

PBL finds its theoretical foundation in constructivist and experiential learning theories because these methods promote active student knowledge construction over passive knowledge reception. Vygotsky and Piaget along with Kolb demonstrated that social interaction combined with experiential learning drives cognitive development according to their theories [9-11]. Their Experiential Learning Theory presents learning as a cycle that begins with concrete experience then progresses through reflective observation to abstract conceptualization and ends with active experimentation [12]. Through PBL and active learning approaches technical education implements these theories to create both deep knowledge acquisition and sustainable skill retention.

2.1 Evolution of Engineering Education Models

The implementation of PBL in current engineering education requires following the pedagogical development that produced this approach. The educational methodology has experienced substantial changes during the last twenty years as it moved progressively from instructor-led practices to student-led methods. Traditional Teaching methods from before 2010 focused on content delivery while instructors transmitted information to students who acted as passive learners. Assessment for learning occurred through summative examinations that ignored practical application and skill acquisition.

During the period from 2010 to 2015 Technology-Enhanced Learning (TEL) brought digital tools and multimedia content and Learning Management Systems (LMS) which transformed teachers into facilitators and students into content consumers [13]. Information acquisition received primary attention throughout this period although application remained minimal [14,15].

The implementation of Outcome-Based Education (OBE) represented a fundamental transformation during the period from 2015 to 2023. OBE frameworks needed educators to precisely connect course objectives to program outcomes which focused on measurable competencies in knowledge, skills and attitudes [16,17]. The assessment process evolved into a continuous and rubric-based system that fostered the connection between instruction methods and expected learning achievements [18]. Although the model succeeded at developing skills it struggled to effectively tackle real-world problems due to its lack of creative elements.

The emerging educational approach of PBL (2023 onwards) enables students to work together on authentic problem statements through an experiential framework. Teachers function as mentors or coaches who lead students through all stages of the creative process including idea generation and development alongside experimental testing and final presentation [19]. Assessment functions as both formative and reflective practice which leads to deeper learning opportunities and promotes innovation alongside teamwork development [20]. Figure 1 shows the progressive teaching-learning paradigm evolution across time as illustrated in a comparative timeline framework of four essential teaching methods.

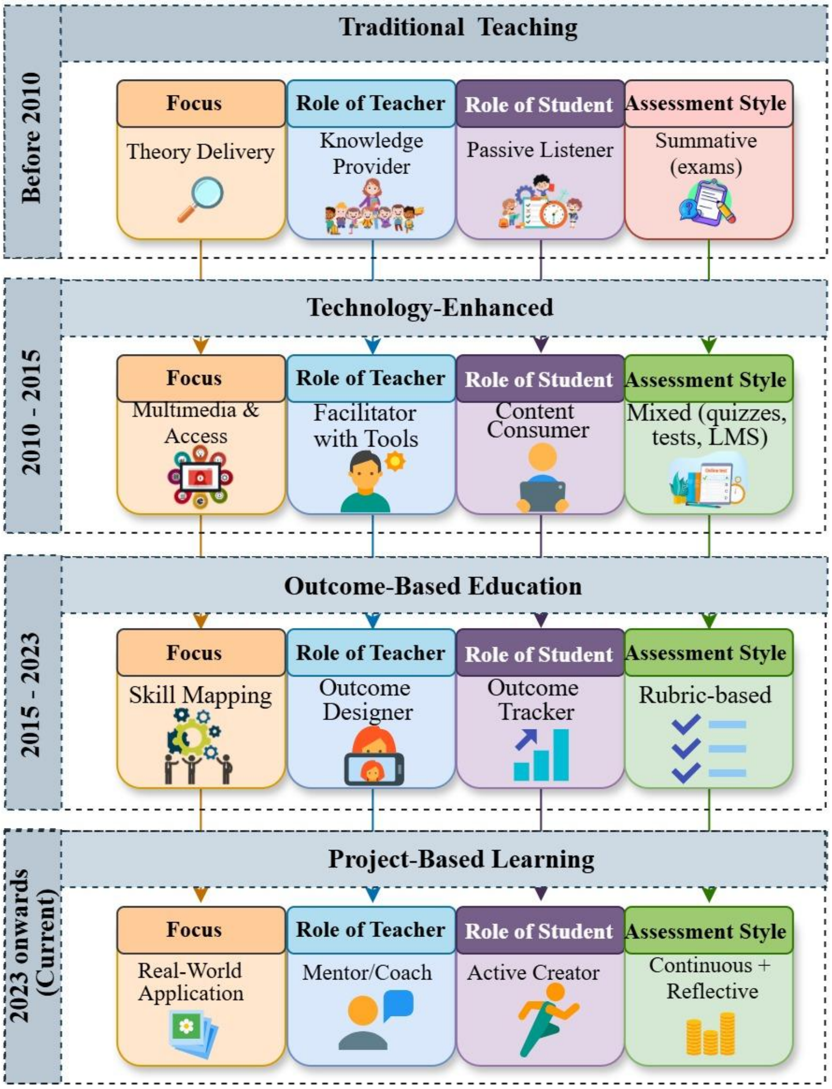


Fig. 1. Evolution of Teaching-Learning Approaches in Engineering Education

2.2 Blockchain Education: Current Practices and Gaps

Blockchain, being a rapidly evolving and multidisciplinary field, poses unique pedagogical challenges. Existing academic literature and syllabi tend to emphasize conceptual understanding—such as cryptographic primitives, consensus protocols [21,22], or decentralized architectures—often without equipping students with hands-on development experience. Some university initiatives and online platforms (e.g. Coursera, Udemy) introduce development tools like Remix, Truffle, or Ganache, but the implementation rarely follows an end-to-end, full-stack project model [23,24]. More critically, there is limited empirical analysis of student learning outcomes in such blockchain courses [25].

Studies suggest the potential of blockchain education in transforming digital infrastructures, yet the gap remains in translating this into practice-based academic programs. Additionally, aspects such as smart contract testing, security validation, frontend integration, and testnet deployment are often skipped or underemphasized [26].

This gap creates a compelling need to explore structured, project-based methodologies that allow students to engage in design, development, testing, and deployment of decentralized applications (DApps). Furthermore, the lack of formal assessment rubrics, difficulty calibration, mentoring mechanisms, and documentation readiness in existing blockchain education models points to significant improvement areas [27,28]. A structured comparison between traditional teaching methods and project-based learning (PBL) approaches in blockchain education further highlights these contrasts and opportunities as described in table 1.

Table 1. Comparison of Traditional and Project-Based Learning Approaches in Blockchain Education

Aspect	Observations / Issues	Gaps Identified	Recommendations / Contributions
Pedagogical Focus	Emphasis on conceptual understanding (e.g., cryptographic primitives, consensus, decentralized architectures)	Lack of hands-on development experience	Shift toward project-based learning (PBL) models
Tools Introduced	Platforms like Coursera/Udemy use tools such as Remix, Truffle, Ganache	Rarely structured as full-stack, end-to-end projects	Need for comprehensive DApp development exercises
Learning Outcomes	Limited empirical analysis of student outcomes in blockchain courses	Weak feedback loops and evaluation mechanisms	Introduce learning assessments and rubrics
Curriculum Gaps	Smart contract testing, security validation, frontend integration, testnet deployment are underemphasized	Critical components of real-world DApp development are ignored	Incorporate these elements into academic syllabi
Mentorship and Evaluation	Lack of difficulty calibration, mentoring mechanisms, documentation practices	Students lack guided support and readiness for real-world projects	Formalize mentorship and documentation standards
Study Contribution	Case study of a PBL-based blockchain education model	Bridges gap between theory and practice	Demonstrates technical/pedagogical outcomes using student feedback, testing coverage, research output

The present study addresses these gaps by documenting the design and execution of a PBL-driven blockchain learning activity. It evaluates both technical and pedagogical outcomes, supported by student feedback, testing coverage, and research output. This case study thus contributes to the evolving narrative of engineering pedagogy by grounding blockchain education within an actionable, student-centered framework.

While various instructional frameworks exist for evaluating pedagogical design, COPES [29] focuses specifically on how students regulate their own cognition and effort during learning. This makes it particularly relevant to project-based environments where students define, monitor, and revise their learning goals dynamically.

3. Proposed Methodology

The implementation and evaluation of PBL in blockchain was conducted through a structured three-phase methodology. The sequential phases were developed to create a progressive buildup of students' conceptual knowledge and technical competence and research abilities. Through this method students learn through classroom instruction while developing technical skills through tool-based practice and team projects leading to outcome dissemination. The subsequent sections detail how each phase functions through its algorithmic framework which includes essential tasks and required tools and deliverables. Student artifacts and feedback were later analyzed using the COPES model to explore cognitive regulation and task engagement during the project.

3.1 Phase 1 - Conceptual Clarity

Algorithm: Phase 1 - Conceptual Clarity

Input: Students enrolled in the course.

1. **Course Initiation**
   - Introduce fundamental blockchain concepts.
   - Explain course roadmap, phases, and evaluation rubrics.
   - Showcase blockchain case studies and research papers to inspire students.
2. **Formation of Groups**
   - Form teams of 3–4 students based on interest or complementary skills.
   - Assign initial roles (e.g., researcher, developer, writer).
3. **Tool Familiarization**
   - Introduce key blockchain development tools:
     - Local blockchain simulator (Ganache).
     - Smart contract development frameworks (Remix, Truffle, Hardhat).
     - Wallet (MetaMask).
     - Blockchain APIs (Infura, Alchemy).
     - Frontend/Backend libraries (Web3.js, Ethers.js, Node.js).
     - Hosting and storage (IPFS, Vercel).
     - Testing frameworks (Mocha, Chai).
4. **Skill Building**
   - Teach students:
     - Types of Blockchain (Public, Private, Hybrid, Consortium).
     - Consensus Algorithms (POW, POS, POA, PBFT).
     - Smart Contract Languages (Solidity, Vyper, Chaincode).
     - Platforms for development (Ethereum, Polygon, Hyperledger).

Output:

- Identified Problem Statement
- Justification for Blockchain Use
- System Requirements Document

Phase 1 focuses on building a strong conceptual foundation. As detailed in Algorithm 1, the course begins with an orientation covering blockchain fundamentals, course roadmap, and evaluation rubrics. Case studies and research papers are shared to inspire students and provide real-world context. The stepwise activities conducted during the conceptualization phase are outlined in figure 2.

Students are grouped into teams of 3–4 based on interest or complementary skills, with roles such as developer or researcher assigned. Key blockchain tools—Ganache, Remix, MetaMask, Infura, Web3.js, and testing libraries—are introduced through hands-on sessions.

Parallel skill-building modules cover blockchain types, consensus algorithms, and smart contract languages. By the end of this phase, each group identifies a problem statement, justifies the use of blockchain, and prepares a system requirements document.

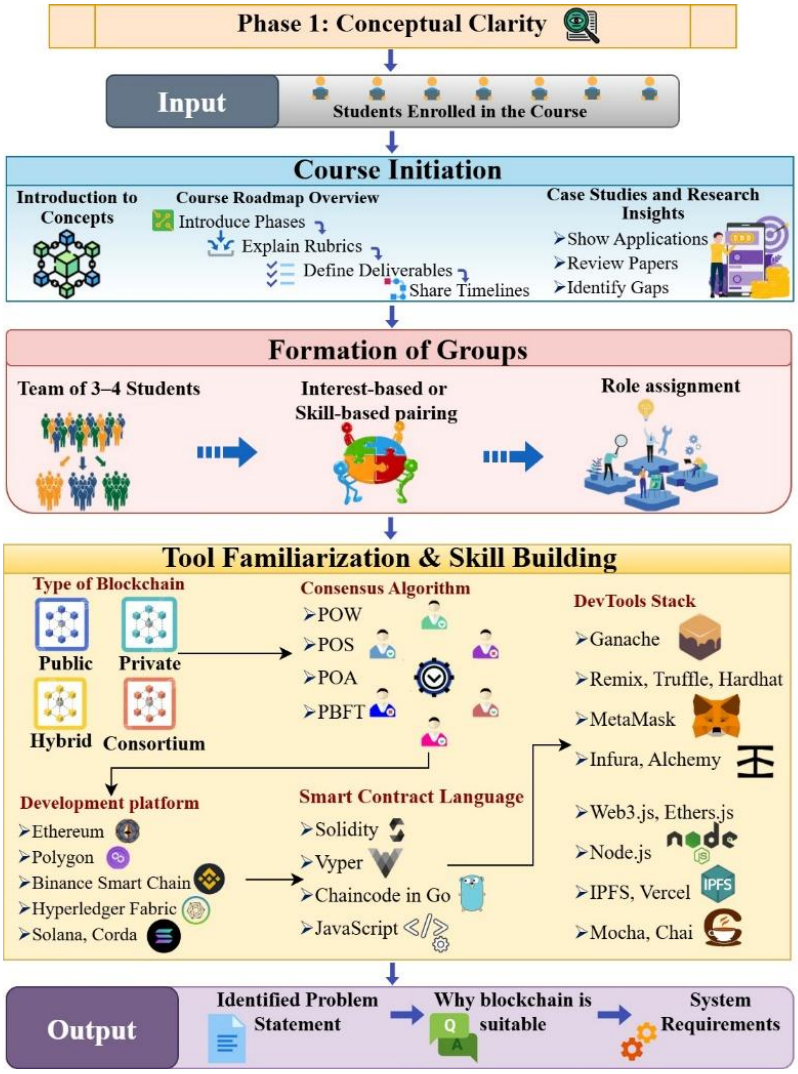


Fig. 2. Algorithmic Flow of Phase 1: Conceptual Clarity

3.1.1 Group Size Analysis

A total of 97 students enrolled in the course, who were distributed into 40 project groups for the activity conducted under Phase 1: Conceptual Clarity. Students were encouraged to form groups based on shared interests, balanced capabilities, and mutual availability. The grouping strategy allowed for flexibility, promoting collaboration while accommodating individual working preferences.

The group size distribution revealed a varied collaboration pattern among the students. Specifically, 10 groups consisted of four students each, while 9 groups were formed with three members. A significant number—15 groups—opted to work in pairs, and notably, 1 group consisted of a single student who chose to work independently.

This data is visualized in figure 3 using a donut chart, which clearly illustrates the distribution across group sizes. The largest proportion of students opted to work in pairs, followed by groups of four and three. Solo participation was rare but permitted under special circumstances.

The variation in group sizes provided a natural context to observe the impact of team size on task division, documentation quality, and conceptual depth. Larger groups (4 members) generally demonstrated stronger coordination across responsibilities—such as requirement analysis, tool exploration, and technical writing. In contrast, smaller groups often showed higher individual involvement per member, with some producing concise and focused submissions.

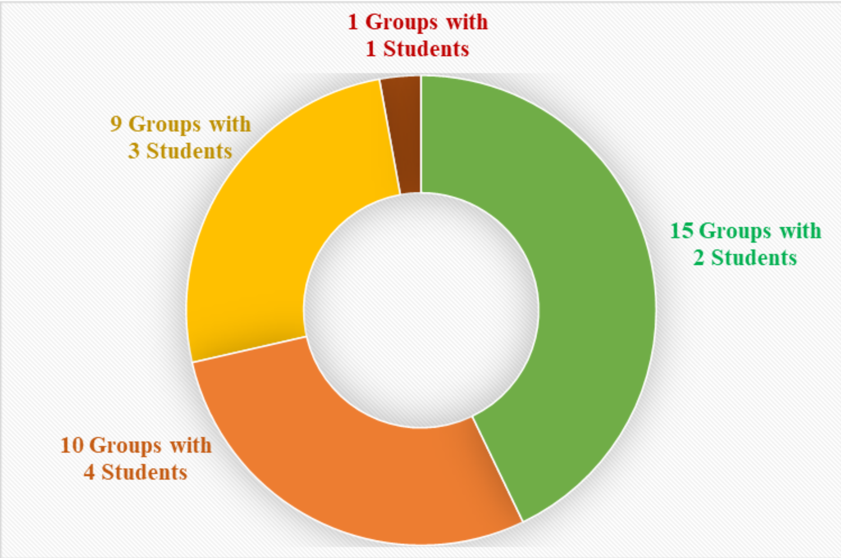


Fig. 3. Analysis of project group size

3.1.2 Domain-Wise Project Categorization

The analysis of 40 student projects demonstrates a wide variety of application domains that highlight blockchain's adaptability and students' differing interests. The assignments were split across 11 specific domains that combined real-world usefulness with decentralized capabilities.

Seven of the 40 student projects concentrated on Government and Public Services applications that included decentralized FIR filing and government fund allocation together with blockchain-based land registries and KYC verification systems. The high number of students working on transparent governance projects demonstrates their commitment to accountability as well as trust and transparency in decision-making processes.

Educational applications ranked second in popularity among participants who developed solutions including credential verification systems together with attendance tracking through ElizaEdu and digital notekeeping and scholarship distribution systems. Academic institutions demonstrate increasing recognition of blockchain technology's potential to establish secure verifiable records.

Each of the Healthcare and Law/Forensics domains received five groups focusing on the challenges of medical record management and genomic data protection and digital evidence preservation operations that need immutability and traceability capabilities.

Table 2. Domain-wise Classification of Student Projects

Domain	No. of Groups	Example Projects
Healthcare	5	Healthcare management, Genomic data, Patient records
Education	6	ElizaEdu, Credentials system, Scholarship tracking, To-Do App, Digital notes
Government/Public Services	7	FIR system, Fund allocation, Voting, Land registry, KYC
Supply Chain / Traceability	4	Food traceability, Aquaculture, Drug authenticity
Finance / Banking	4	Savings/lending, Crowdfunding, Loyalty points
Law/Forensics/Evidence	5	Digital evidence, Chain of custody, Blockchain for defense
Energy / IoT / Grid	2	Energy market, Shipment tracking
Storage / Cloud / Filesystems	3	Cloud storage, Document storage, Disk renting
Transport / Mobility	2	P2P rideshare, Car sharing
NGO / Non-profit	1	NGO management system
Agriculture / Rural Tech	1	OxFarm protocol

The Finance/Banking and Supply Chain/Traceability domains each engaged 4 groups, with applications ranging from lending and savings platforms to food traceability and counterfeit prevention.

Other domains included Storage and Cloud Systems (3 groups), Energy/IoT (2 groups), Transport/Mobility (2 groups), and niche areas like NGO operations and Agriculture, each represented by a single project.

The distribution is summarized in table 2 and visualized in figure 4, which together highlight a well-rounded exploration of blockchain applications across multiple sectors.

This domain-level analysis demonstrates that students did not restrict themselves to popular or generic blockchain applications. Instead, many selected contextually relevant, high-impact use cases where decentralization offered measurable value—an encouraging outcome aligned with the instructional objective of meaningful technology adoption.

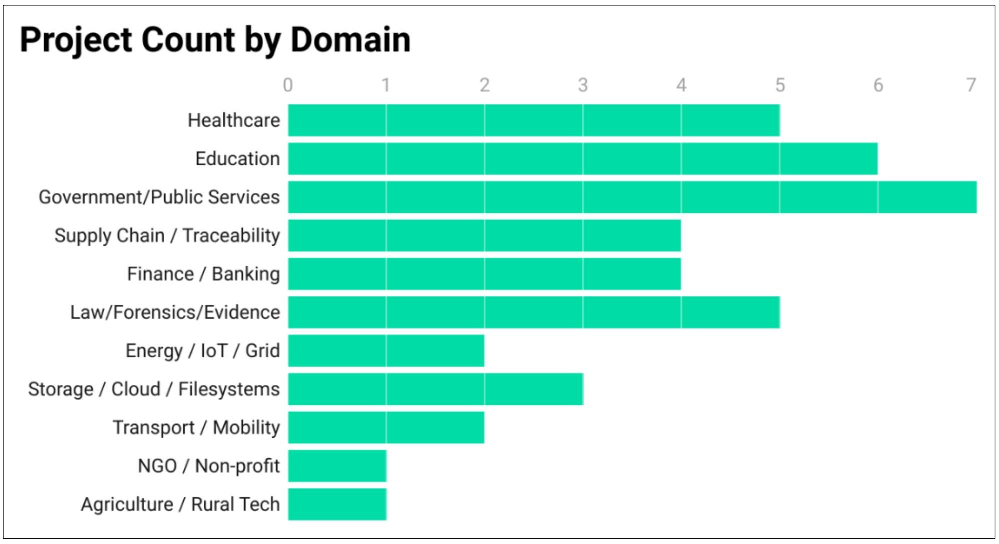


Fig. 4. Bar Graph of Project Distribution Across Domains

3.1.3 Evaluation of Blockchain Applicability Justification

To ensure that students did not apply blockchain superficially or as a buzzword, a detailed rubric was developed to assess the justification of blockchain applicability for each proposed project. The rubric aimed to evaluate whether students demonstrated a sound understanding of when and why a decentralized solution was necessary, and whether they could meaningfully relate blockchain features to their chosen problem statement. Each group was evaluated across four key dimensions:

- Relevance – whether blockchain was a logical fit for the selected problem domain.
- Decentralization Need – whether the problem inherently required distributed, trustless architecture.
- Feature Mapping – the extent to which students aligned blockchain features (e.g., immutability, transparency, traceability) with specific system needs.
- Clarity – the quality and coherence of the explanation provided.

Each criterion was scored on a 5-point scale, leading to a maximum of 20 points per group. The rubric encouraged critical thinking by requiring students to explicitly articulate the value blockchain brings to their solution, thereby discouraging the application of blockchain where it may not be necessary.

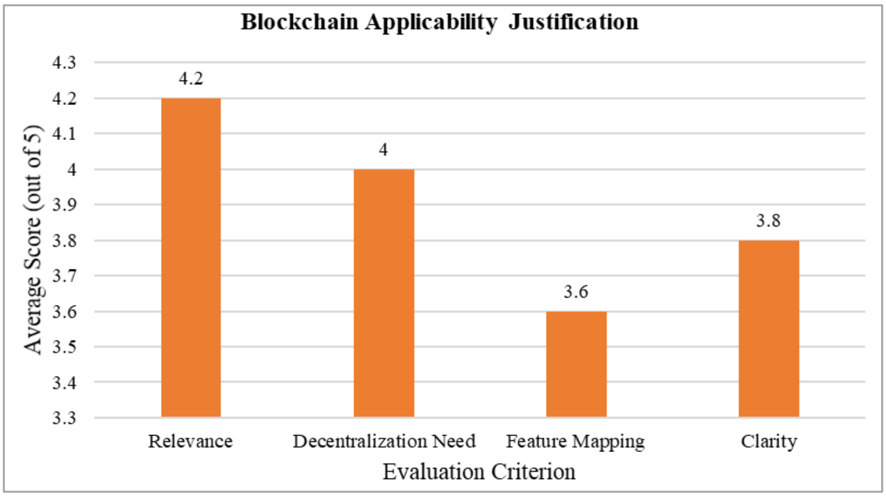


Fig. 5. Criterion-wise Analysis of Student Justifications for Blockchain Use

The evaluation results are summarized in figure 5, which presents the average score across all 40 groups for each criterion. Students demonstrated strong understanding in recognizing blockchain's relevance and the role of decentralization in their projects. The average score of 4.2 in Relevance indicates that most groups selected problem statements where blockchain offered a genuine advantage over traditional systems. Similarly, a score of 4.0 for Decentralization Need shows that students were able to identify use cases that required trust minimization and distributed control.

However, the relatively lower score of 3.6 in Feature Mapping suggests that while students understood blockchain conceptually, some struggled to explicitly tie specific blockchain characteristics to technical aspects of their system design. Clarity of explanation averaged at 3.8, indicating reasonably well-structured justifications, though a few required refinement in articulation and technical framing.

3.2 Phase 2 - Solution Design and Implementation

In Phase 2, the focus shifts from foundational understanding to structured solution development. As described in algorithm 2, students translate their identified problem into a feasible blockchain-based architecture. This involves the design of system flow diagrams, smart contract interfaces, and integration points. A structured representation of the solution design and implementation methodology is provided in Figure 6.

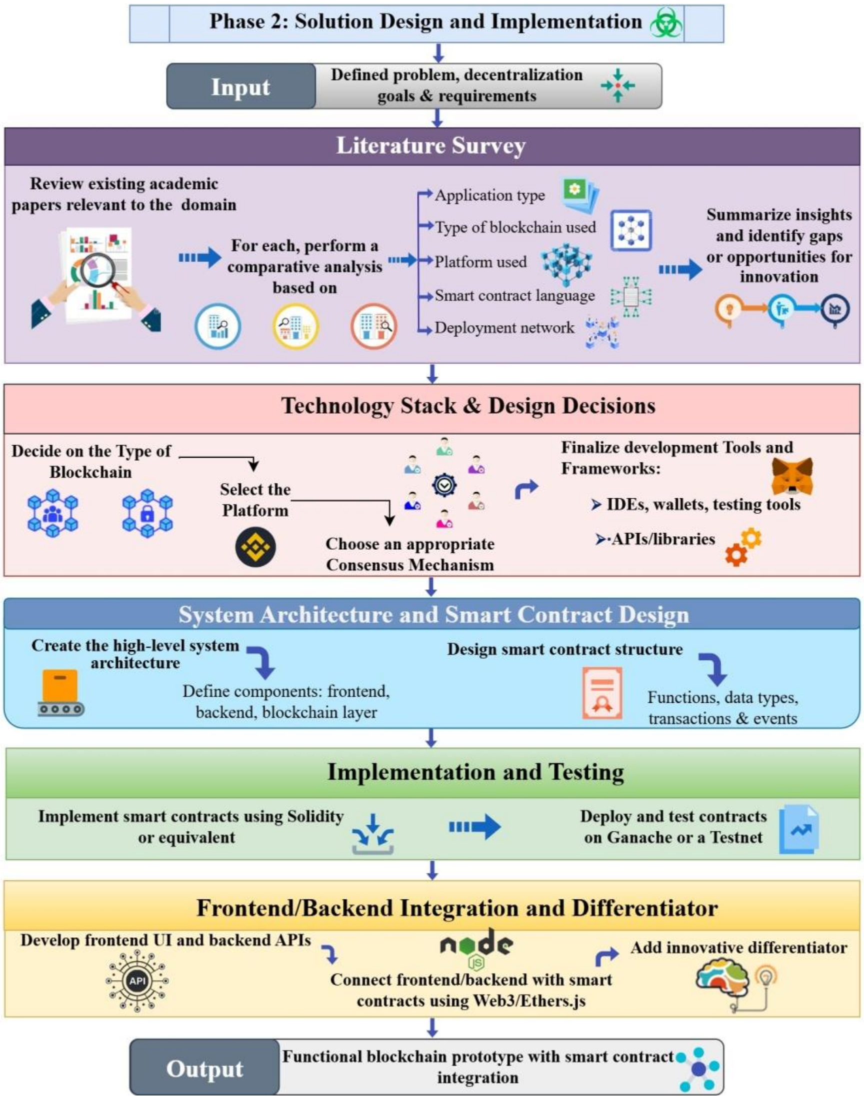


Fig. 6. Algorithmic Flow of Phase 2: Solution Design and Implementation

Each group begins by modeling the solution, identifying key blockchain features such as transparency, immutability, or decentralization that address their problem. Students use Remix or Truffle to create and deploy smart contracts after which they conduct preliminary testing through Ganache and MetaMask.

The integration between frontend and backend components happens through libraries Web3.js or Ethers.js. The development of minimal UI mockups or full interfaces by students utilizes either React or Vue programming frameworks. The iterative build-test approach helps students improve their logic while enhancing modularity as they move toward full-stack integration during the following phase.

Algorithm: Phase 2 - Solution Design and Implementation

Input:

- Approved problem statement, purpose of decentralization, and requirement document from Phase 1
- Familiarity with basic blockchain tools and architecture concepts

1. **Literature Survey**
   - Review existing academic papers, or open-source projects relevant to the selected domain
   - For each, perform a comparative analysis based on:
     - Application type
     - Type of blockchain used
     - Platform used
     - Smart contract language
     - Deployment network
   - Summarize insights and identify gaps or opportunities for innovation.
2. **Technology Stack and Design Decisions**
   - Decide on the Type of Blockchain
   - Select the Platform
   - Choose an appropriate Consensus Mechanism
   - Finalize development Tools and Frameworks:
     - IDEs, wallets, testing tools
     - APIs/libraries
3. **System Architecture and Smart Contract Design**
   - Create the high-level system architecture
     - Define components: frontend, backend, blockchain layer
   - Design smart contract structure:
     - Functions, data types, transactions & events
4. **Implementation and Testing**
   - Implement smart contracts using Solidity or equivalent
   - Deploy and test contracts on Ganache or a Testnet
5. **Frontend/Backend Integration and Differentiator**
   - Develop frontend UI and backend APIs
   - Connect frontend/backend with smart contracts using Web3/Ethers.js
   - Add at least one innovative differentiator

Output: A functional and unique blockchain prototype integrating smart contracts with frontend/backend systems

3.2.1 Blockchain Type Analysis

The students are required to evaluate which blockchain type would suit their use case better. At the beginning of phase 2, students were educated about parameters such as trust model, privacy needs, scalability considerations, and decentralization goals, basis which they were encouraged to select the appropriate blockchain type based on the specific problem's requirement. An analysis of the blockchain types selected by the student groups reveals a strong inclination toward public and consortium-based blockchain models, reflecting their relevance in multi-party and trustless environments. As summarized in table 3, 18 out of 40 groups (45%) chose public blockchains, typically for use cases requiring transparency, tokenization, or user interaction — such as crowdfunding platforms, decentralized marketplaces, traceability systems, and digital note-sharing applications.

Table 3. Distribution of Blockchain Types Selected by Student Groups and Their Application Domains

Blockchain Type	No. of Groups	Percentage (%)	Example Domains
Public	18	45%	Crowdfunding, Marketplace, Traceability, Notes
Consortium	14	35%	Land registry, Voting, Scholarship, KYC
Private	7	17.5%	Healthcare, Digital Evidence, Genomic data
Hybrid	1	2.5%	Cloud storage with internal + external access

Consortium blockchains were selected by 14 groups (35%), especially in domains like land registry, digital voting, scholarship allocation, and KYC verification, where control is shared among trusted stakeholders while still maintaining decentralization. Private blockchains, used by 7 groups (17.5%), were mainly adopted for sensitive data domains such as healthcare, forensic evidence management, and genomic data handling — cases where data privacy, access control, and auditability are critical.

Only one group (2.5%) explored a hybrid model, combining public transparency with private data layers — as seen in cloud storage applications requiring selective access. The overall distribution of blockchain type selection is visually depicted in figure 7, clearly illustrating student awareness of how blockchain architecture aligns with domain-specific constraints and trust requirements.

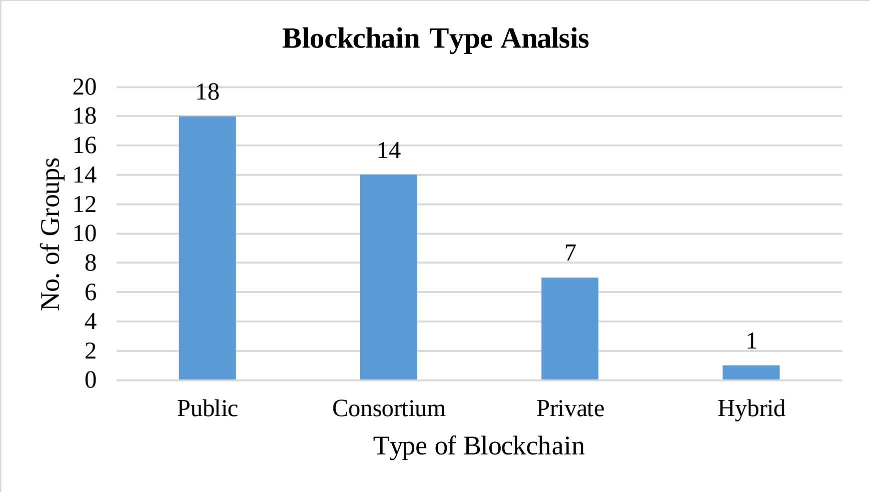


Fig. 7. Blockchain Type Selection across 40 Student Projects

3.2.2 Platform Selection Analysis

The selection of blockchain platforms by student groups reflects both the functional requirements of their use cases and the accessibility of development ecosystems. As shown in table 4, the most widely adopted platform was Ethereum, used by 21 out of 40 groups (52.5%). Its widespread adoption was driven by compatibility with smart contract languages like Solidity, robust tooling support, and suitability for use cases involving voting, identity management, crowdfunding, and fund tracking.

Table 4. Blockchain Platform Selection by Student Groups and Corresponding Use Cases

Platform	**No. of Groups**	**Percentage (%)**	**Typical Use Cases**
Ethereum	21	52.5%	Voting, Identity, Fund Tracking, Crowdfunding
Polygon	11	27.5%	Marketplace, Loyalty, Mobility, Event Tracking
Hyperledger Fabric	7	17.5%	Healthcare, Forensics, Digital Evidence, Defense
Not Specified / Mixed	1	2.5%	Theoretical project or platform-agnostic research

Polygon, chosen by 11 groups (27.5%), was preferred for projects requiring low-cost transactions and high throughput, such as marketplaces, loyalty point systems, mobility solutions, and event-tracking applications. Its Ethereum compatibility and active testnet infrastructure made it especially appealing for student prototypes.

Hyperledger Fabric was selected by 7 groups (17.5%), primarily for projects dealing with sensitive data and enterprise-grade requirements, such as healthcare, digital evidence, and defense-related systems. Fabric's permissioned architecture aligned well with privacy-focused applications.

One group (2.5%) worked on a platform-agnostic or conceptual design, without committing to a specific deployment platform. This typically occurred in exploratory or early-phase research projects.

The distribution of platform choices is also visually presented in figure 8, illustrating a clear preference for Ethereum-based ecosystems while also showcasing thoughtful alignment between platform selection and application domain.

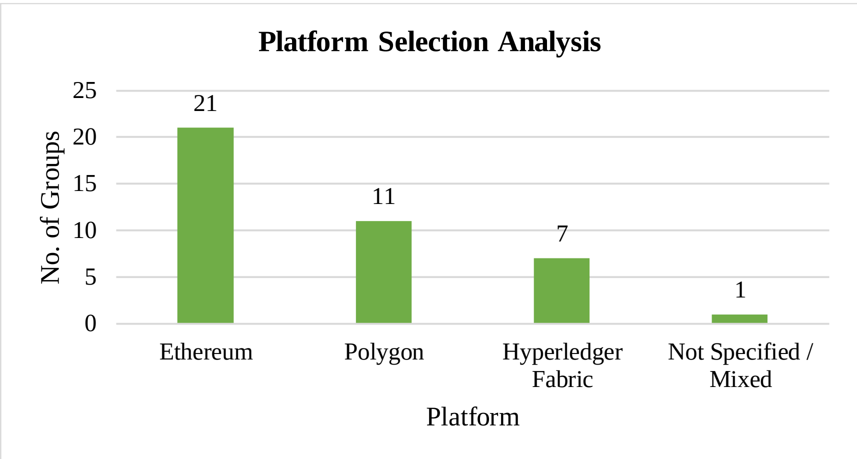


Fig. 8. Distribution of Blockchain Platforms Used in Student Projects

3.2.3 Smart Contract Language Analysis

The selection of smart contract languages across student projects closely aligned with the platforms they chose. As shown in table 5, Solidity was used by 33 out of 40 groups (82.5%), reflecting the popularity and maturity of Ethereum and Polygon ecosystems for decentralized application development. These projects typically addressed domains such as voting, crowdfunding, identity verification, and supply chain traceability.

In contrast, 7 groups (17.5%) adopted Go (Chaincode) as their contract language, corresponding with the use of Hyperledger Fabric for privacy-sensitive applications in healthcare, digital forensics, and genomic data management.

Table 5. Smart Contract Language Selection and Application Domains

Language	No. of Groups	Percentage (%)	Associated Platforms	Typical Use Cases
Solidity	33	82.5%	Ethereum, Polygon	Voting, crowdfunding, identity, traceability
Go (Chaincode)	7	17.5%	Hyperledger Fabric	Healthcare, digital forensics, genomic data

3.2.4 Consensus Mechanism Analysis

Student blockchain projects received their operational and architectural form from the consensus mechanism selection process. The analysis presented in table 6 along with figure 9 demonstrates that Proof of Stake (PoS) emerged as the consensus mechanism of choice for 21 out of 40 student projects because it provided energy-efficient solutions that worked natively on popular platforms including Ethereum and Polygon. The selected implementations focused on education, voting, crowdfunding and food traceability systems.

The Practical Byzantine Fault Tolerance (PBFT) protocol enabled 7 groups to implement their blockchain applications through Hyperledger Fabric specifically for healthcare and genomic data management and digital forensics. The Proof of Authority (PoA) mechanism received selection by 6 groups who deployed it through Ganache or private Ethereum networks because of its low latency and test environment suitability. Another 6 groups operated without implementing a specific consensus protocol, either focusing on high-level design or conducting only simulated demonstrations.

This distribution reflects a thoughtful alignment of consensus strategies with domain-specific needs and technological feasibility.

Table 6. Consensus mechanism used in students' projects

Consensus Mechanism	Description	Common Platforms	Mapped Domains	No. of Groups
Proof of Stake (PoS)	Energy-efficient; validators are selected based on stake in the network	Ethereum (PoS), Polygon	Education, Voting, Crowdfunding, Loyalty, P2P Sharing, Food Traceability	21
Practical Byzantine Fault Tolerance (PBFT)	Tolerant to malicious nodes; suitable for permissioned, private networks	Hyperledger Fabric	Healthcare, Genomic Data, Forensics, Defense	7
Proof of Authority (PoA)	Validator-based; fast, simple, suited for dev/test and private use	Ganache, Private Ethereum Nets	NGO Management, Lottery, Local simulations	6
Unspecified / Simulated Only	Projects focused on design; consensus not explicitly implemented	N/A	Multidimensional Blockchain, Disk Renting, Early-stage ideas	6

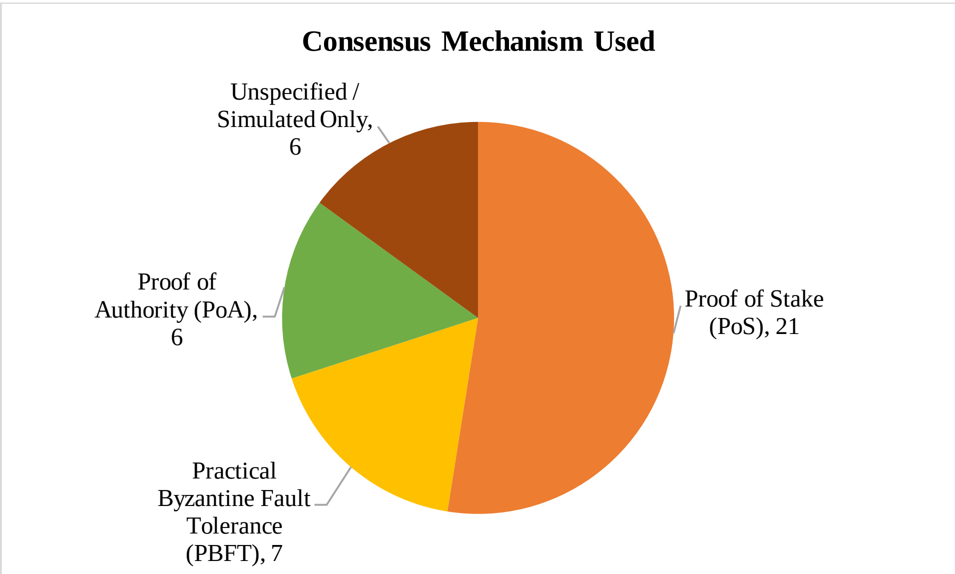


Fig. 9. Consensus Mechanism Analysis

3.2.5 Deployment Network Analysis

Deployment networks varied across projects depending on the platform selected and the maturity of implementation. As presented in table 7 and visualized in figure 10, the Mumbai Testnet (Polygon) emerged as the most popular choice, with 14 groups leveraging it for lightweight, cost-effective testing of smart contracts. Ethereum testnets, specifically Goerli and Sepolia, were used by 13 groups, providing a more authentic Ethereum environment while retaining testnet flexibility.

9 groups opted for Ganache or other local development networks such as Hyperledger Fabric's test setup, allowing rapid iteration without network dependency. Finally, 4 groups focused on conceptual or partially implemented designs and did not deploy their code, choosing instead to simulate or document the deployment process.

Table 7. Deployment networks

Deployment Network	No. of Groups	Notes
Mumbai Testnet	14	Polygon-based deployments
Goerli + Sepolia Testnet	13	Ethereum-based test deployments
Ganache / Local Network	9	Local-only testing via Ganache/HL
Not Deployed (simulated)	4	Written code only, no actual testing

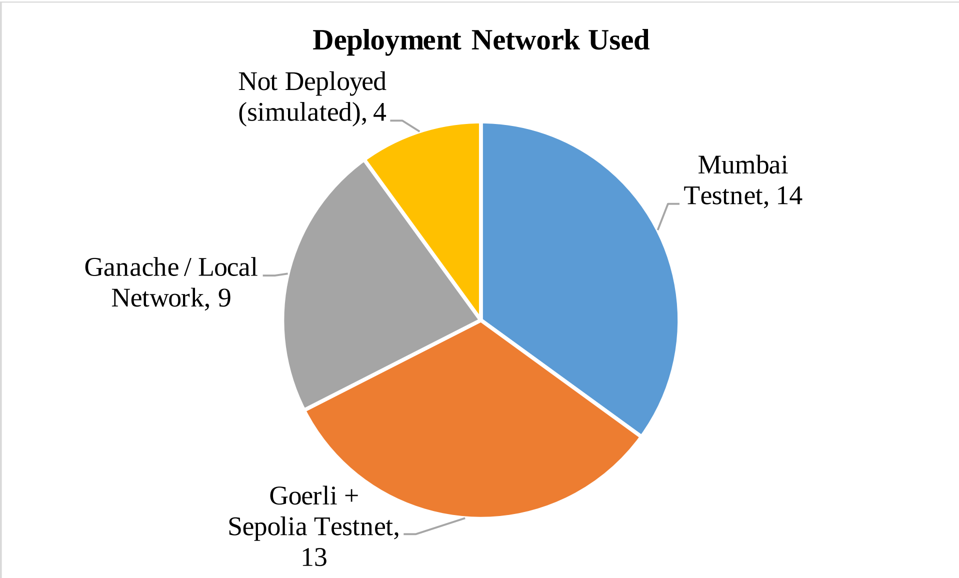


Fig. 10. Deployment network used

3.2.6 Development Tools and Frameworks Used

The selection of development tools and frameworks across student projects reflects a pragmatic approach to blockchain prototyping. As described in table 7 and shown in figure 11, Remix IDE was the most utilized environment, with 30 groups employing it for quick Solidity development and deployment. For more advanced or customized workflows, 10 groups adopted Visual Studio Code in conjunction with Truffle or Hardhat, facilitating robust testing and scripting.

In wallet integration, MetaMask stood out as the de facto choice, used by 33 groups for managing identities and interacting with deployed contracts. On the programming side, Web3.js was favored by 22 groups for connecting frontends to smart contracts, while Ethers.js offered a modern alternative in 8 projects.

In backend and storage layers, Node.js with Express was adopted by 20 groups, while IPFS was selectively used in 5 projects to demonstrate decentralized file storage. For user interfaces, React and Vue.js were chosen by 15 groups, enabling interactive decentralized application (dApp) development.

Table 8. Development Tools and Frameworks Used

Tool Category	Tool / Framework	No. of Groups	Usage Purpose
IDE	Remix IDE	30	Writing and compiling Solidity smart contracts
	VS Code	10	Used with Truffle/Hardhat or Hyperledger Fabric
	Truffle / Hardhat	10	Project structuring, deployment scripting, testing
Wallet	MetaMask	33	Account management, contract interaction
APIs & Libraries	Web3.js	22	Frontend-backend smart contract interaction
	Ethers.js	8	Contract interface via cleaner modern APIs
	IPFS	5	Decentralized storage
	Node.js + Express	20	Backend API creation and integration with contracts
	React / Vue.js	15	Frontend UI frameworks for dApp interfaces

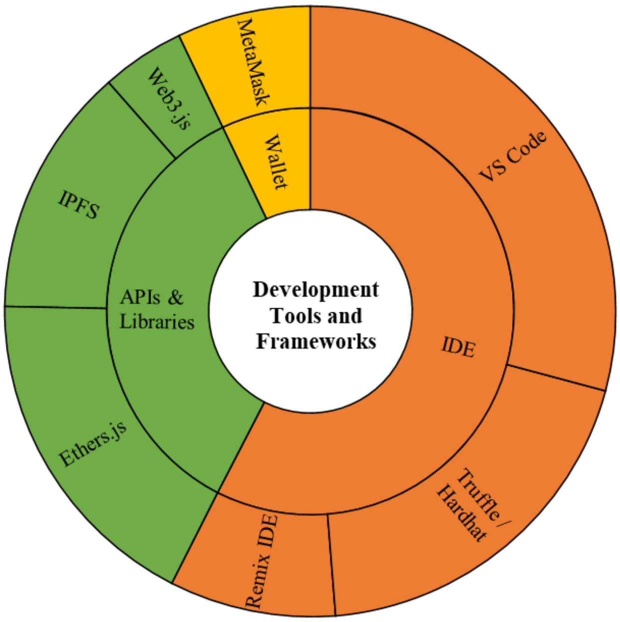
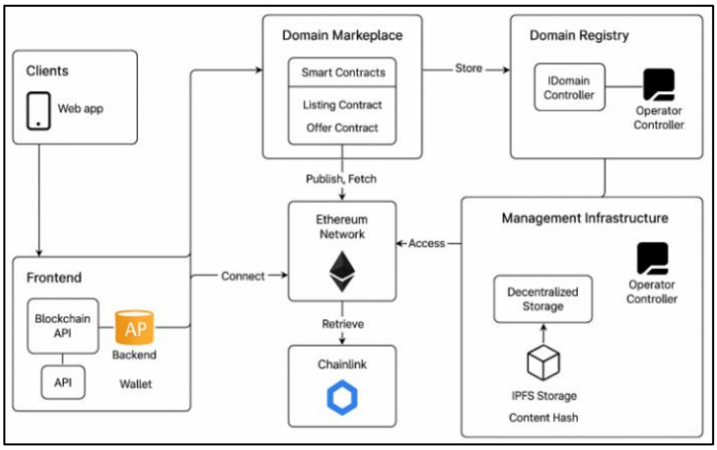


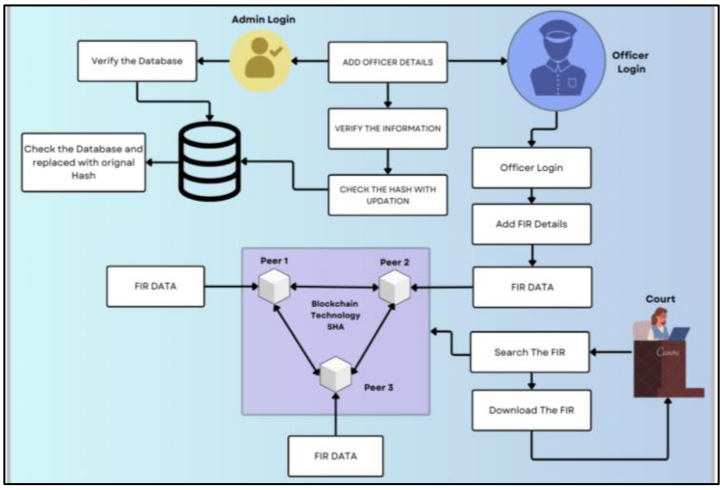
Fig. 11. Development Tools and Frameworks Used

3.2.7 Architecture Diagrams

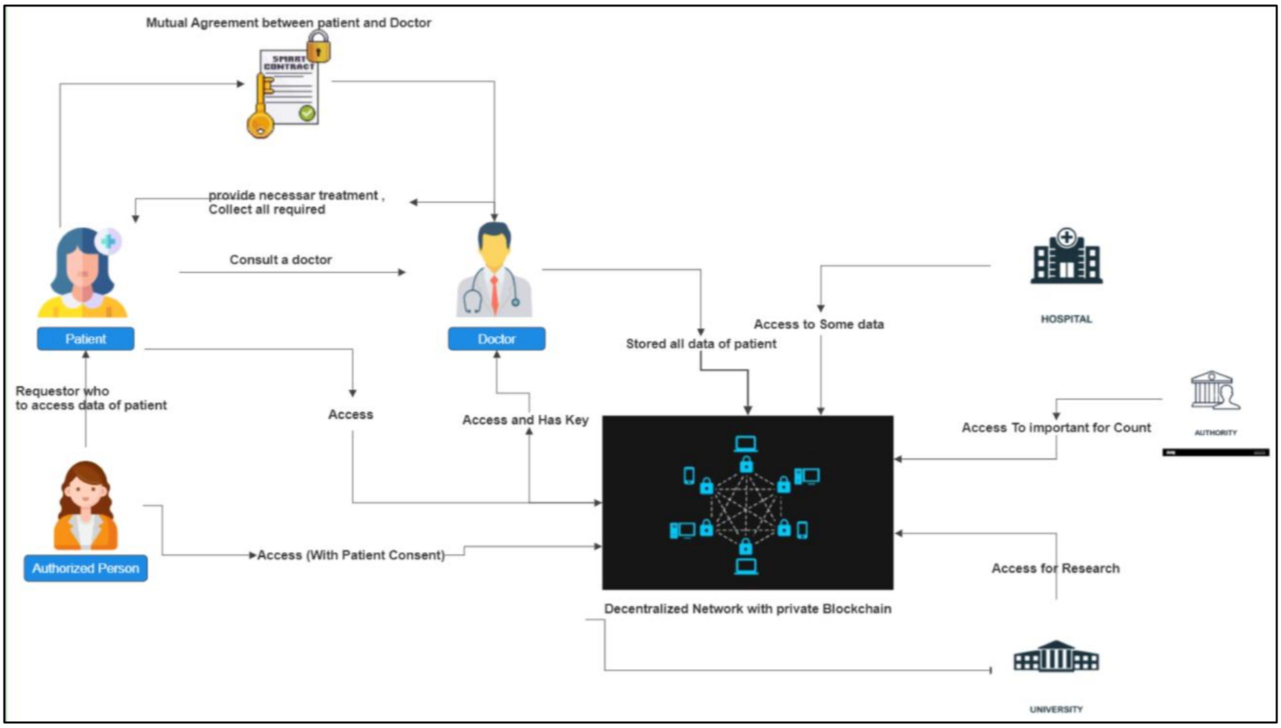
In Phase 2, each student group was required to design a comprehensive system architecture as a blueprint for their blockchain-based application. The objective of this exercise was to encourage students to think beyond isolated smart contracts and consider the broader interaction between users, blockchain components, off-chain systems, and interface layers.



(a)



(b)



(c)

Fig. 12. (a). Domyn – Decentralized Domain Trading Platform, (b). Decentralized FIR Filing and Monitoring System, (c). Blockchain-Based Genomic Data Management for Secure DNA Sequencing

These architecture designs played a central role in aligning the problem statement, technology decisions, and implementation. They helped define system boundaries, identify trust models, and specify the interaction between on-chain and off-chain logic. Students were expected to incorporate platform-specific components, integration tools, and deployment environments into their models.

To illustrate the diversity and depth of student work, three representative architecture diagrams have been included: Domyn – Decentralized Domain Trading Platform figure 12(a), Decentralized FIR Filing and Monitoring System figure 12(b), and Blockchain-Based Genomic Data Management for Secure DNA Sequencing figure 12(c). These examples reflect different blockchain types, domains, and system complexities, highlighting the students' ability to model secure, decentralized systems across real-world contexts.

3.2.8 Challenges Faced in Adopting Blockchain Tools and Platforms

While the structured project-based methodology was effective in guiding students through the development process, the flexibility of of projects offered presented its own challenges with each student group working on different stacks. Most students were new to blockchain environments, and even the introductory tools presented a significant learning curve. For instance, managing transactions in MetaMask, working with Ganache's local blockchain simulator, and writing bug-free smart contracts in Solidity often led to confusion, especially when deployment failed without clear errors.

Many groups initially struggled with the integration of smart contracts into their frontend using Web3.js or Ethers.js, with issues ranging from incorrect ABI imports to asynchronous data rendering. Platforms like IPFS and testnet deployments via Sepolia or Mumbai Testnet brought additional complexity, such as faucet token delays, RPC misconfigurations, and contract verification errors.

To support students through these hurdles, the course incorporated dedicated mentoring sessions, peer learning checkpoints, and curated resources—including recorded demos and code walkthroughs. Over time, most groups became comfortable with the toolchain, often citing these moments of struggle as key learning milestones. This phase not only improved their technical skills but also helped build persistence, collaboration, and debugging discipline—competencies highly valued in industry settings.

3.3 Phase 3 – Integration, Testing, and Research Publication

In Phase 3, detailed in algorithm 3, students advance toward completing a fully functional DApp. The final phase included testing, documentation, and research dissemination as shown in figure 13. This stage emphasizes end-to-end system integration, where smart contracts, UI, and backend logic are cohesively deployed. Students conduct functional testing—including failure scenarios—to ensure stability and correctness.

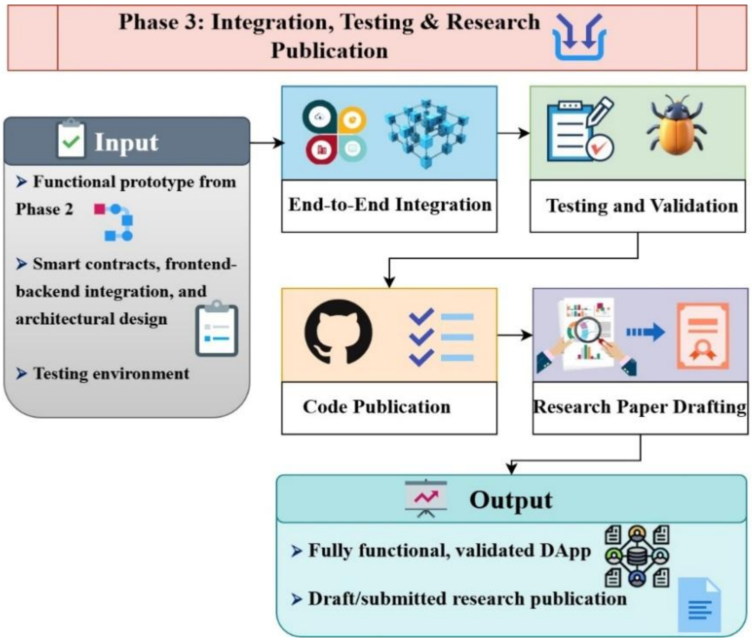


Fig. 13. Algorithmic Flow of Phase 3: Integration, Testing, and Research Dissemination

Comprehensive documentation is created, including user manuals, setup guides, and architecture flow diagrams. To encourage scholarly output, students are guided in drafting research papers based on their project work, often resulting in draft submissions or publications. Simultaneously, final codebases are made public via GitHub with README files and deployment instructions.

This final phase reinforces communication, technical writing, and real-world delivery skills—key components of research-oriented, project-based learning.

Algorithm: Phase 3 – Integration, Testing, and Research Publication

Input:

- Functional prototype from Phase 2
- Smart contracts, frontend-backend integration, and architectural design
- Testing environment

1. **End-to-End Integration**
   - Connect smart contracts to the complete frontend and backend stack
   - Ensure seamless transaction flow and UI responsiveness
2. **Testing and Validation**
   - Conduct testing
   - Simulate failure cases and edge conditions
3. **Code Publication**
   - Upload source code to GitHub repository
   - Add README, setup instructions, and dependencies
4. **Research Paper Drafting**
   - Draft paper highlighting problem, methodology, results, and contribution
   - Submit to a relevant conference or journal

Output:

- Fully functional, validated decentralized application (DApp)
- Draft/submitted research publication

3.3.1 Deployment and Integration Status of DApps

The deployment maturity of the DApps developed by student groups was assessed to understand the end-to-end execution of blockchain-based solutions. As shown in table 9, 75% (30 out of 40) of the groups successfully delivered a fully integrated and working decentralized application, complete with frontend, backend, and blockchain contract components. About 15% achieved partial integration, where either the frontend or backend components were incomplete or not connected. The remaining 10% of the groups could only demonstrate code-level simulations without actual deployment or integration efforts.

Table 9. Deployment and Integration Status of DApps

Deployment Status	No. of Groups	Percentage (%)
Fully Integrated & Working DApp	30	75%
Partial Integration (e.g., missing UI/backend)	6	15%
Code-level Simulation Only	4	10%

3.3.2 DApp Quality Grading

The developed DApps were evaluated on multiple parameters such as interface quality, technical implementation, and completeness of documentation. As summarized in table 10 and visualized in figure 14, the average score across groups indicates a strong emphasis on backend and blockchain functionality, with room for improvement in areas such as user documentation and frontend polish.

The highest-rated component was contract deployment and interaction (avg. score 4.2), reflecting effective use of wallets like MetaMask and testnets. UI/UX quality averaged 3.8, especially among the 10+ groups that adopted modern frontend frameworks like React or Vue. Code quality and modularity stood at 3.5, showing good use of version control and project structuring on GitHub. However, documentation and user manuals received the lowest score (2.9), highlighting a frequent gap in developer-written instructions. The end-to-end flow averaged 3.7, consistent with the deployment statistics noted earlier.

Table 10. DApp Quality Grading

Evaluation Parameter	Avg. Score (out of 5)	Notes
UI/UX Quality	3.8	10+ groups used React/Vue
Contract Deployment/Interaction	4.2	High MetaMask integration
Code Quality / Modularity	3.5	GitHub repos well-structured
Documentation / User Manual	2.9	Some lacked detailed instructions
End-to-End Flow Validity	3.7	Full flow working in 24+ groups

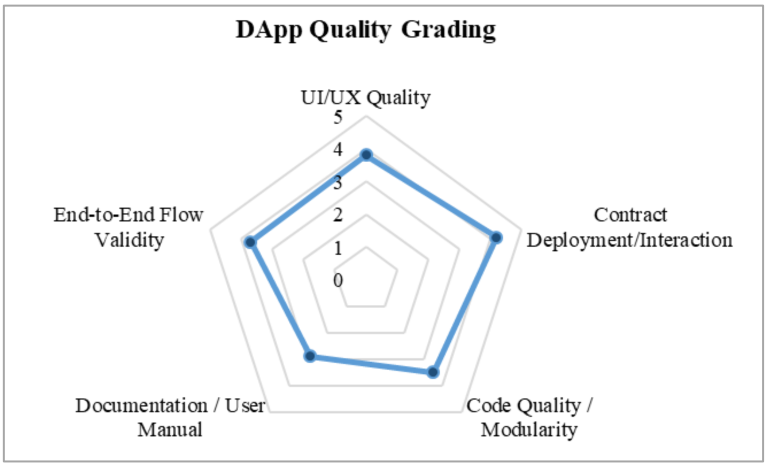


Fig. 14. DApp Quality Analysis

3.3.3 Testing Coverage Analysis

To ensure reliability and robustness, different types of testing were evaluated across the projects. As summarized in Table 11, half of the groups implemented unit testing, primarily using Solidity testing frameworks or JavaScript-based tools. Integration testing was adopted by 18 groups, ensuring proper coordination between smart contracts and backend services. Notably, edge case simulation was conducted by 10 groups, usually focusing on error handling and failure conditions. However, 8 groups did not conduct any formal testing, often correlating with projects that remained at a simulation or conceptual stage.

Table 11. Testing Coverage Analysis

Type of Testing	No. of Groups Implemented	Comments
Unit Testing	20	Solidity and JS tests
Integration Testing	18	Across backend-smart contract
Edge Case Simulation	10	Mostly in error handling
No Formal Testing	8	Mostly in simulated-only projects

The lack of formal testing in a few groups, particularly simulation-only teams, limited the overall reliability and confidence in DApp functionality.

3.3.4 Documentation Preparedness

Comprehensive documentation is critical for the usability and reproducibility of decentralized applications. As reflected in Table 12, 75% of the student groups submitted a User Manual, typically outlining basic functionalities and usage steps. A Deployment Guide, detailing environment setup and testnet configurations, was provided by 25 groups. Architecture and system flow diagrams were available in 20 submissions, offering clarity on internal design. Encouragingly, 33 groups maintained a project README on GitHub, ensuring a clear overview of their DApp.

Table 12. Documentation Preparedness by Student Groups

Document Type	No. of Groups Submitted	Comments
User Manual	30	Basic features and steps
Deployment Guide	25	Includes setup, testnet details
Architecture & Flow Diagrams	20	Helps in understanding system flow
README on GitHub	33	Includes project overview

3.3.5 Research Paper Output

As part of Phase 3, students were encouraged to document and formalize their project outcomes through research writing. As shown in table 13, 5 groups successfully published their research papers, while 12 groups had submitted their drafts to conferences or journals. An additional 16 groups had prepared drafts but had not submitted them, indicating progress toward scholarly output. However, 5 groups did not attempt a paper, often due to incomplete deployment or time constraints.

Table 13. Research Paper Output Status

Research Paper Status	No. of Groups
Published	5
Drafted and Submitted	12
Draft Prepared (not submitted)	16
Not Attempted	5

In addition to submission statistics, a number of papers have been of academic quality as well as practical interest. Among them, there are: Domyn: A Decentralized Domain Name Marketplace Using Blockchain and NFTs, Blockchain-Based Decentralized FIR Registration and Monitoring System, and ElizaEdu: An AI-Powered Web3 System for Secure Event Attendance Verification. These publications addressed real-life issues in governance, policing and education, and were effective as they were published in peer-reviewed journals. Taken together, these outputs support the possibility of the structured project-based learning to make a significant contribution to the discourse in the academic sphere and to the design of solutions relevant to the industry.

3.3.6 GitHub Publication

Open-source publication of student work was encouraged to foster transparency, reproducibility, and collaborative improvement. As summarized in table 14, 28 out of 40 groups (70%) published a public repository with a working setup guide. Another 8 groups (20%) maintained a private or incomplete repo, typically due to project confidentiality or unfinished documentation. Four groups (10%) did not upload any repository, often corresponding to projects still in the simulation phase.

Table 14. GitHub Publication Status

GitHub Repo Available	No. of Groups	%
Public Repo with Setup Guide	28	70%
Private Repo / Incomplete	8	20%
No Repo Provided	4	10%

3.3.7 Student Feedback Analysis on Project-Based Blockchain Learning Activity

To evaluate the effectiveness of the activity from the learner's perspective, feedback was collected from all 40 groups using a structured rubric. The parameters assessed included clarity, relevance, collaboration, technical growth, and mentoring support.

As shown in table 15, most average ratings were above 4.0, indicating positive learner experiences. The highest-rated parameter was Relevance to Real-World Applications (avg. 4.6, SD 0.38), followed by Overall Satisfaction (avg. 4.5, SD 0.43). The lowest rating was for Documentation Support and Confidence in Presenting, signaling scope for improvement in presentation training and mentoring depth.

In their reflections, several students described the technical tools as "powerful but intimidating" at first. However, nearly all agreed that overcoming these challenges led to a sense of accomplishment and deeper understanding than passive learning could offer.

These insights are visualized in figure 15, where a bar chart displays the mean ratings with error bars representing standard deviation, providing a clearer picture of consistency across responses.

Table 15. Student Feedback Analysis

Feedback Parameter	Average Rating (out of 5)	Standard Deviation (SD)	Rating Scale
Clarity of Objectives	4.3	0.42	1 = Poor to 5 = Excellent
Relevance to Real-World Applications	4.6	0.38	1 = Not Relevant to 5 = Highly Relevant
Improvement in Technical Skills	4.5	0.45	1 = No Improvement to 5 = Significant Improvement
Team Collaboration Experience	4.2	0.50	1 = Poor to 5 = Excellent
Availability of Mentoring Support	4.1	0.53	1 = Not Available to 5 = Highly Supportive
Confidence in Presenting Work	4.0	0.48	1 = Not Confident to 5 = Very Confident
Difficulty Level (Optimal = 3.5–4)	3.8	0.40	1 = Very Easy to 5 = Very Difficult
Interest and Motivation	4.4	0.36	1 = Low to 5 = Highly Motivated
Overall Satisfaction	4.5	0.43	1 = Not Satisfied to 5 = Fully Satisfied

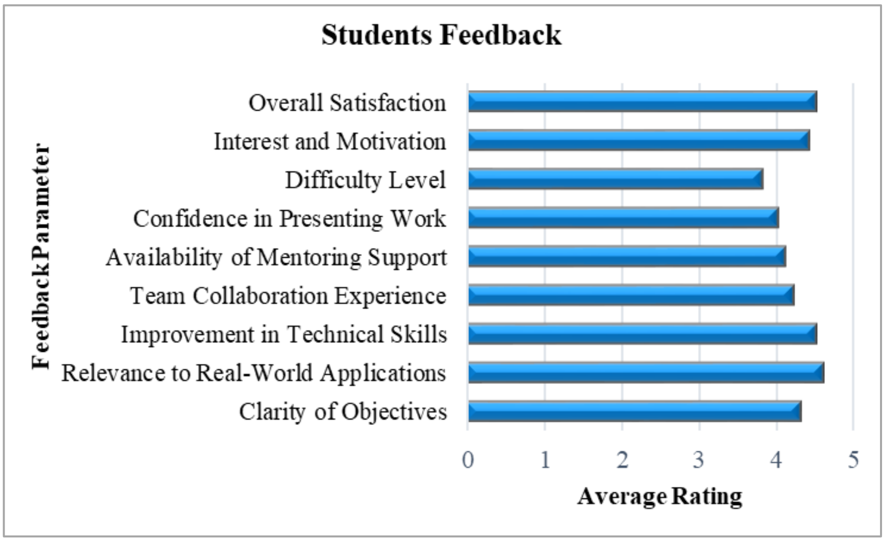


Fig. 15. Average student ratings across feedback parameters

3.3.8 Most enjoyable aspect of this activity

To understand student engagement at a deeper level, qualitative feedback was collected from all 40 groups regarding the most enjoyable aspect of the blockchain-based learning activity. Their responses were categorized into key themes based on frequency and sentiment.

As shown in table 16, DApp Development and Deployment emerged as the most commonly appreciated component (18 mentions), followed closely by Collaborative Teamwork (15 mentions). This indicates a strong enthusiasm among students for hands-on, application-driven tasks and peer collaboration. Additionally, Writing and Testing Smart Contracts (12 mentions) and Technical Mentoring and Guidance (10 mentions) were noted as meaningful and engaging.

Besides the quantitative ratings, there were open-ended responses which indicated common themes. Some students said that developing a real-world DApp was the most interesting aspect of the course, although it was not a smooth process. Some reported that debugging smart contracts was initially frustrating but in the end worthwhile. Several students reported that peer work was deemed as vital in solving technical challenges, and they enjoyed the possibility of making a presentation of something concrete, rather than just code or reports.

The data is visualized in figure 16, which presents a horizontal bar chart to illustrate the frequency of mentions across themes.

Table 16. Most Enjoyable Aspect of the Activity

Enjoyable Aspect (Theme)	Number of Groups Mentioning
DApp Development and Deployment	18
Writing and Testing Smart Contracts	12
Real-world Relevance of Problem Statements	9
Collaborative Teamwork	15
Technical Mentoring and Guidance	10
Final Presentation / Demo	7

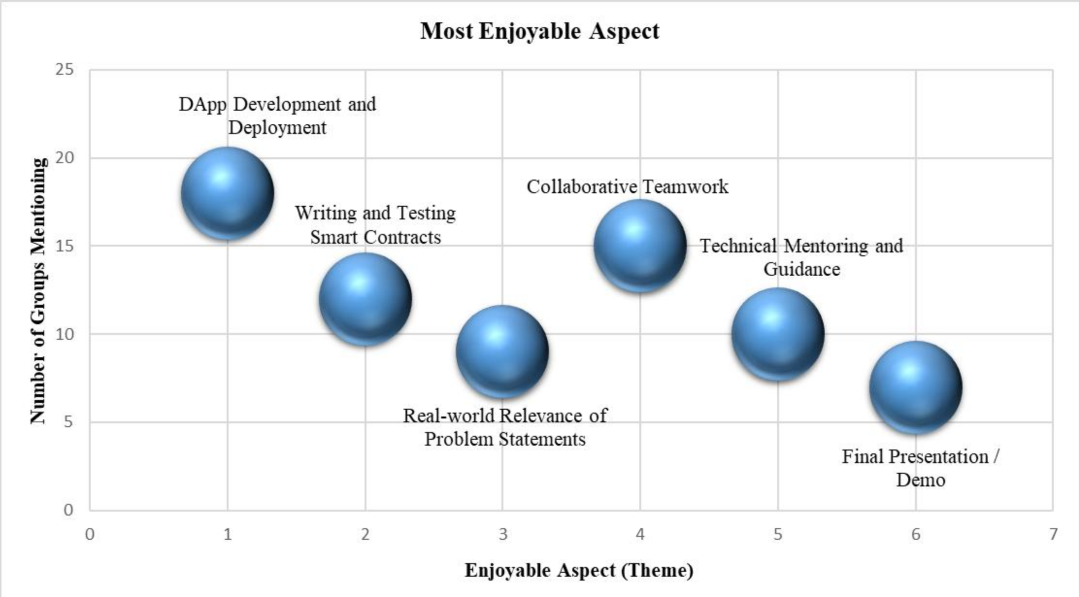


Fig. 16. Most enjoyable aspects of the activity as reported by student groups

3.3.9 Limitations and Challenges in Implementing the Project-Based Learning Model

Although the structured PBL methodology managed to improve student engagement, collaboration, and technical skills, a number of practical and pedagogical constraints were observed in the implementation process. The difficulties provide valuable lessons in the development of future versions of the model:

- Student Readiness: Although there were group formation strategies, the differences in the students background when it came to prior exposure to programming or blockchain resulted in unequal contribution within teams.
- Time Constraint: Ideation, development, testing, and documentation proved difficult to implement in within a single semester, especially in tandem with other academic needs by students.
- Consistency in Assessment: It was challenging to interpret innovation, level of integration and quality of documentation in a consistent way across a considerable number of projects, even when using detailed rubrics.
- Mentoring Bandwidth: The active project teams of 40 required a lot of faculty time. Although a few students said they did not have immediate access to guidance, mentors ensured that there was a checkpoint, tool walkthrough, and one-on-one troubleshooting session during the critical stages of development each week.
- Transparency in Research Output: Some groups also had a vague idea of the expectations of paper writing and publication readiness, which influenced the uniformity of research artifacts.

Such constraints point at the importance of improved scaffolding and pacing systems, including intermediate checkups, peer reviews, and simplified assessment procedures. Nevertheless, the activity has proven that PBL is effective in overcoming the gap between theoretical training and practical blockchain development with the help of sufficient mentoring and academic frameworks.

4. Applying the COPES framework to student learning

The COPEs model proposed by Winne and Hadwin [29] allows us to interpret how students managed their own learning during the blockchain project. Below, we reflect on each of the five components in the context of student performance and reported experiences:

- **Conditions**: Students entered the project with varying levels of prior exposure to blockchain, programming, and group-based problem solving. The structured scaffolding, toolchain introductions, and clearly defined deliverables created external task conditions that influenced strategy use.
- **Operations**: Cognitive operations such as planning, encoding, problem decomposition, and collaborative debugging were observed in team meetings and code submissions. Several groups adopted iterative testing, diagramming, and analogy-based reasoning when translating blockchain concepts into functioning DApps.
- **Products**: Observable products included smart contracts, architecture diagrams, frontend integration, deployment on testnets, documentation, and research papers. These reflected both the cognitive strategies applied and the quality of team regulation.

- **Evaluations**: Many student groups demonstrated evidence of self-monitoring by benchmarking their progress against peer groups and rubrics. Iterative revisions and critical peer discussions suggested that teams actively monitored their own deficiencies and adjusted effort accordingly.
- **Standards**: The primary standards were implicit (peer benchmarks) and explicit (rubrics and mentor feedback). Over time, several groups internalized performance criteria—particularly around testing completeness and functional deployment—which shaped their quality expectations.

This analysis suggests that PBL activities like the one described here can be potent environments for supporting self-regulated learning, provided the learning environment makes conditions, standards, and feedback mechanisms visible and actionable.

5. Conclusion

This study presents a structured case of implementing PBL in the domain of blockchain education, aligning pedagogical goals with rapidly advancing industry needs. The initiative used a phased methodology for bridging theoretical instruction with practical application by starting with conceptual clarity then implementing and finally disseminating the knowledge. The evaluation process showed that technical capabilities and collaborative skills and learner confidence improved significantly based on deployment reports testing outcomes documentation standards and student evaluation results. PBL emerged naturally from traditional teaching methods into technology-enhanced and outcome-based learning and proved especially beneficial for new fields including decentralized applications and smart contracts. The model receives support from positive research findings and GitHub distribution along with high student satisfaction results. The study demonstrates that PBL implemented with proper scaffolding functions as a lasting educational framework to teach blockchain concepts and other real-world challenges.

The COPES framework provides additional insight into how students regulate their cognitive and collaborative processes in complex, open-ended tasks. By aligning PBL environments with the five elements of the COPES framework, educators can make each stage of the learning process transparent and actionable. This, in turn, empowers students to take ownership of their work, iteratively improve, and achieve deeper, more durable outcomes.

This work is a case study and we acknowledge the inherent limitation of a single case study not being generalizable. We suggest that future work involves exploring the applicability and efficacy of this PBL model across multiple institutions, engineering disciplines, academic levels, and through cross-disciplinary collaboration to enhance the generalizability and richness of the findings.

References

[1] Gaivoronskii, D.V., Kutuzov, V.M. and Minina, A.A., 2017, November. Digital transformation of engineering education. In *2017 IEEE VI Forum Strategic Partnership of Universities and Enterprises of Hi-Tech Branches (Science. Education. Innovations)(SPUE)* (pp. 3-6). IEEE.

[2] Yakovenko, I., Kulumbetova, L., Subbotina, I., Zhanibekova, G. and Bizhanova, K., 2019. The blockchain technology as a catalyst for digital transformation of education. *Technology*, *10*(01), pp.886-897.

[3] Darling-Hammond, L., 2009. Teaching and educational transformation. *Second international handbook of educational change*, pp.505-520.

[4] Etherington, M., 2019. The Challenge with educational transformation. *Journal of Culture and Values in Education*, *2*(1), pp.96-112.

[5] Deore, S., Bachche, R., Bichave, A. and Patil, R., 2021, June. EHR-Sec: a blockchain based security system for electronic health. In *International Conference on Soft Computing and Signal Processing* (pp. 295-303). Singapore: Springer Nature Singapore.

[6] Mentzer, K., Frydenberg, M. and Yates, D.J., 2020. Teaching applications and implications of blockchain via project-based learning: A case study.

[7] Xing, Z. and Chen, Z., 2022. A project-based approach for learning blockchain technology. *Procedia Computer Science*, *214*, pp.581-589.

[8] Smirnova, E. and Vatolkina, N., 2018. Teacher training in Engineering education: search for new pedagogical approaches. In *INTED2018 Proceedings* (pp. 6240-6247). IATED.

[9] Loftus, S. and Higgs, J., 2005. Reconceptualising problem-based learning in a Vygotskian framework. *Focus on Health Professional Education: A Multi-Professional Journal*, *7*(1), pp.1-14.

[10] Kolb, D.A., 2013. The process of experiential learning. In *Culture and processes of adult learning* (pp. 138-156). Routledge.

[11] Kolb, D.A., 2014. *Experiential learning: Experience as the source of learning and development*. FT press.

[12] Matriano, E.A., 2020. Ensuring Student-Centered, Constructivist and Project-Based Experiential Learning Applying the Exploration, Research, Interaction and Creation (ERIC) Learning Model. *International Online Journal of Education and Teaching*, *7*(1), pp.214-227.

[13] Dobre, I., 2015. Learning Management Systems for higher education-an overview of available options for Higher Education Organizations. *Procedia-social and behavioral sciences*, *180*, pp.313-320.

[14] Lantada, A.D., 2020. Engineering education 5.0: Continuously evolving engineering education. *International journal of engineering education*, *36*(6), pp.1814-1832.

[15] Kirkwood, A. and Price, L., 2014. Technology-enhanced learning and teaching in higher education: what is 'enhanced'and how do we know? A critical literature review. *Learning, media and technology*, *39*(1), pp.6-36.

[16] Baartman, L.K. and De Bruijn, E., 2011. Integrating knowledge, skills and attitudes: Conceptualising learning processes towards vocational competence. *Educational research review*, *6*(2), pp.125-134.
[17] Leung, K., Trevena, L. and Waters, D., 2014. Systematic review of instruments for measuring nurses' knowledge, skills and attitudes for evidence-based practice. *Journal of advanced nursing*, *70*(10), pp.2181-2195.
[18] Gulati, S., 2008. Technology-enhanced learning in developing nations: A review. *The International Review of Research in Open and Distributed Learning*, *9*(1).
[19] Shaheen, S., 2019. Theoretical perspectives and current challenges of OBE framework. *International Journal of Engineering Education*, *1*(2), pp.122-129.
[20] Bennett, R.E., 2011. Formative assessment: A critical review. *Assessment in education: principles, policy & practice*, *18*(1), pp.5-25.
[21] Sukackė, V., Guerra, A.O.P.D.C., Ellinger, D., Carlos, V., Petronienė, S., Gaižiūnienė, L., Blanch, S., Marbà-Tallada, A. and Brose, A., 2022. Towards active evidence-based learning in engineering education: A systematic literature review of PBL, PjBL, and CBL. *Sustainability*, *14*(21), p.13955.
[22] Borse, M., Shendkar, P., Undre, Y., Mahadik, A. and Patil, R.Y., 2022. A review of blockchain consensus algorithm. *Expert Clouds and Applications: Proceedings of ICOECA 2022*, pp.415-426.
[23] Fleener, M.J., 2022. Blockchain Technologies: A Study of the Future of Education. *Journal of Higher Education Theory & Practice*, *22*(1).
[24] Bisoyi, A.P. and Patil, R.Y., 2023, February. A smart contracts based approach for the E-commerce Platforms. In *2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON)* (pp. 1-6). IEEE.
[25] Xing, Z. and Chen, Z., 2022. A project-based approach for learning blockchain technology. *Procedia Computer Science*, *214*, pp.581-589.
[26] Lee, T., 2022. Blockchain and Cryptocurrency Distributed Testing Methods. *International Journal of Internet, Broadcasting and Communication*, *14*(1), pp.1-9.
[27] Panadero, E. and Jonsson, A., 2013. The use of scoring rubrics for formative assessment purposes revisited: A review. *Educational research review*, *9*, pp.129-144.
[28] Ocheja, P., Agbo, F.J., Oyelere, S.S., Flanagan, B. and Ogata, H., 2022. Blockchain in education: A systematic review and practical case studies. *IEEE Access*, *10*, pp.99525-99540.
[29] Winne, P. H., & Hadwin, A. F. (1998). Studying as self-regulated learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), Metacognition in Educational Theory and Practice (pp. 277–304). Mahwah, NJ: Lawrence Erlbaum Associates.

Authors' Profiles

Rachana Y. Patil received her Ph.D. in Computer Engineering from the University of Mumbai, India, in 2020. She has authored more than 80 publications in peer-reviewed international journals and conferences. Her research interests include Cryptography, Network Security, Cybersecurity, and Digital Forensics, with a particular emphasis on Network Forensics. She currently serves as the Head of the Department of Computer Engineering at Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. Dr. Patil is a member of IEEE, ACM, and IETE.

Rahul Kulkarni is the Cofounder and Chief Technologist at DoNew, where he uses project based mentorship models to help organizations build technology moonshot products. Rahul is Adjunct Faculty and Advisor at the Pimpri Chinchwad College of Engineering where he helps shape one of India’s first regional language Bachelor’s programs in Computer Science. As Chief Technologist at Samagra, Rahul’s work on open source and capacity building impacts millions through innovative DPI and AI solutions. Previously Rahul has built global products at Google, National Instruments and Sokrati. Rahul holds 9 patents, has a Masters in Engineering from Georgia Tech and a Bachelors from VJTI Mumbai.

How to cite this paper: Rachana Yogesh Patil, Rahul Kulkarni, "A Structured Project-Based Learning Pedagogy to Bridge the Theory–Practical Gap in Blockchain Education", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.6, pp. 76-96, 2025. DOI:10.5815/ijmecs.2025.06.06