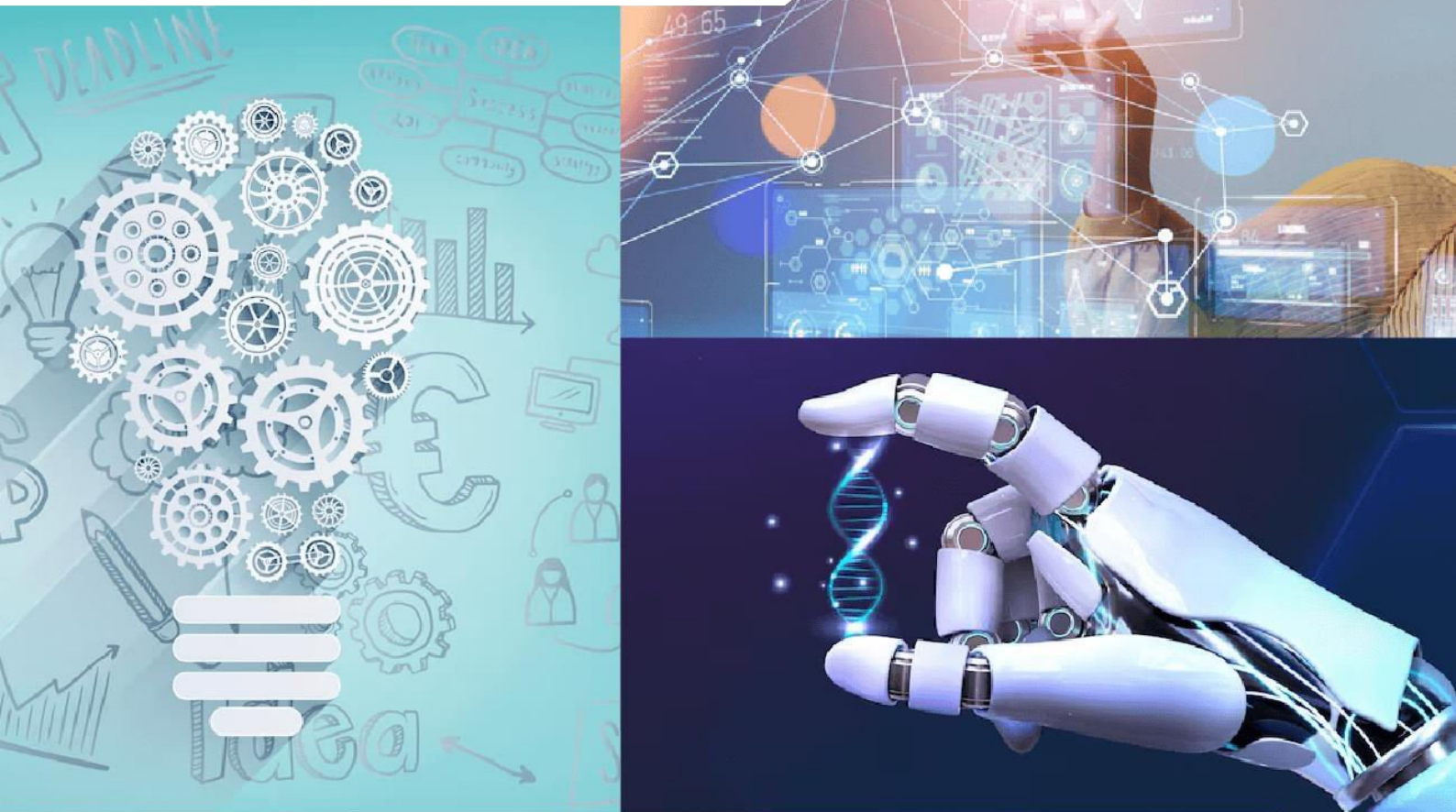




# **International Journal of Advanced Research in Education and Technology (IJARETY)**

**Volume 13, Issue 2, March - April 2026**

**Impact Factor: 8.152**



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# CodeForge: An AI-Integrated Multidimensional Gamified Arcade for Personalized Programming Education

Namratha B

Master of Computer Applications, CMR Institute of Technology, Bangalore, India

**ABSTRACT:** While computational logic is the foundation of computer science, the primary barrier in introductory programming education remains the persistent 'engagement gap.' Traditional pedagogical methods often struggle to maintain student motivation when theoretical concepts appear decoupled from practical application. To address this, we present CodeForge, a gamified learning ecosystem designed to transform technical skill acquisition into an immersive developmental journey. Unlike existing competitive coding platforms, CodeForge implements a novel "Persona Evolution" framework. In this system, student progress is mapped to a 3D avatar that undergoes morphological changes as learning milestones are achieved, providing a tangible sense of growth. The architecture comprises four specialized modules: Bug Hunter (gamified debugging), Code Arena (synchronous 1v1 challenges), Memory Matrix, and Tech Crossword. A significant technical contribution of this work is the integration of a Python-driven intelligence layer capable of dynamically synthesizing personalized curriculum and assessment material from user-provided PDF documentation. The proposed immersive learning approach substantially mitigates "learning fatigue," fostering higher retention rates of complex algorithmic concepts compared to conventional instructional methods. Our findings suggest that re-engineering the learning process as a goal-oriented journey significantly enhances student performance and sustained interest in STEM disciplines.

**KEYWORDS:** Gamification; Pedagogical Programming; Adaptive Skill Acquisition; Student Motivation; Interactive Development Environments; Coding Persistence.

## I. INTRODUCTION

The landscape of academic instruction has undergone a radical shift, yet technical education-specifically in Computer Science-often remains anchored in traditional, lecture-heavy formats. Mastering the art of programming is fundamentally different from other subjects; it is not a passive domain to be memorized, but a craft that requires a relentless cycle of logical reasoning, iterative practice, and hands-on experimentation. Despite this, a significant "engagement gap" persists where standard pedagogy prioritizes theoretical abstraction over actual code execution. For many learners, this lack of practical exposure creates a deceptive sense of competence. A student may follow a lecture with perfect clarity, yet find themselves completely paralyzed when faced with an empty editor and a complex logic problem. This phenomenon, often termed "passive learning," is the primary bottleneck in modern programming education. It occurs when a student parses a code snippet or observes a demonstration without actually engaging with the underlying problem-solving logic. To overcome this gap, it is necessary to move beyond conventional classroom methods and treat technical mastery as a goal-oriented journey rather than an academic chore. The gap between theoretical concepts and hands-on application often results in a sharp decline in student interest and the onset of mental exhaustion, particularly in rigorous technical fields that demand intense mental focus. Consequently, there is an urgent need for innovative pedagogical strategies in programming education that can effectively connect abstract principles with real-world coding practice.

To tackle these challenges, the integration of gamification has emerged as a powerful method for boosting student drive. By embedding features such as XP systems, virtual prizes, and competitive rankings into the course structure, teachers can create more immersive and participatory educational spaces [1]. Previous research suggests that these ludic elements are vital for sustaining curiosity and enriching the overall educational journey [2], [3]. Evidence from several studies confirms that gamified setups result in higher student involvement and superior information recall when contrasted with conventional lecture-based formats [4], [5]. Moreover, recent data suggests that these environments can steer academic success by providing immediate validation and incentive structures that keep learners focused over long durations [6], [7].

**DOI:10.15680/IJARETY.2026.1302072**

Despite the abundance of online coding tools currently on the market, the majority of available solutions suffer from a narrow focus [8]. A large percentage of these services prioritize pure algorithmic problem-solving while neglecting other essential competencies, including troubleshooting logic, mastering technical jargon, or building professional vocabulary. This deficiency underscores the requirement for a truly holistic system that addresses the diverse needs of computer science students.

In light of these shortcomings, this study introduces CodeForge, a gamified educational ecosystem built to transform how programming is taught. By providing dedicated modules for real-world debugging and leveraging intelligent features to automate study resources, CodeForge seeks to deliver a well-rounded and deeply interactive career path for aspiring developers.

## **II. LITERATURE REVIEW**

The integration of game mechanics into educational contexts has evolved from a novel experiment into a well-established field of academic research. This section reviews the foundational theories of gamification, its impact on student performance, and the emergence of AI-driven personalized learning.

The modern definition of gamification, described as the use of game design elements in non-game contexts, has been widely established in prior research [1]. The concept of “gamefulness” has been identified as a key psychological state that drives user engagement. Subsequent studies have developed theoretical frameworks for applying game-based thinking in instructional design [2], [3]. These frameworks suggest that competitive elements such as points and leaderboards function not only as motivational incentives but also as cognitive tools that enhance engagement and support persistence when learners encounter challenging technical material.

A substantial body of empirical research has examined the impact of gamification on student performance. Meta-analytic evidence indicates that gamification can be effective; however, its success depends on specific design factors, particularly the availability of immediate and meaningful feedback [4], [5]. More recent studies demonstrate that gamified systems can positively influence academic performance by transforming passive learning behaviors into active engagement [6], [7], [10], [11]. However, these findings remain context-dependent, highlighting the importance of careful design and implementation.

As educational technologies continue to advance, research has increasingly focused on personalized learning approaches. Studies indicate that one-size-fits-all gamification strategies are often ineffective and emphasize the need for user-centered designs that adapt to individual learners’ skills and needs [8], [13], [14], [19], [20]. A significant development in this field is the integration of Artificial Intelligence (AI) into gamified learning environments. Recent work suggests that combining AI with gamification provides a promising pathway for developing 21st-century software skills [18]. Furthermore, immersive technologies such as virtual reality and 3D learning environments have been shown to significantly enhance the long-term retention of complex technical concepts [9], [12], [15],[17].

## **III. METHODOLOGY**

### **3.1 System Overview**

The CodeForge platform is developed to overcome the limitations of traditional monolithic learning systems by shifting computer science education from a passive format to a more active and engaging experience. In conventional learning environments, students often rely on observation rather than direct involvement, which can reduce motivation and limit the development of practical problem-solving skills. Gamification, which involves applying game-inspired elements in non-gaming contexts, has been recognized as an effective way to improve student engagement and knowledge retention [1].

Previous studies [2], [3] indicate that the use of game-based techniques can enhance cognitive engagement and support long-term learning. Based on these insights, CodeForge incorporates features such as experience points (XP), ranking systems, and leaderboards to encourage continuous participation. One of the key components of the platform is the 3D Persona Evolution Engine, which visually reflects user progress through an evolving digital avatar. This design ensures that every completed activity contributes to a clear and measurable learning progression [4], [5].

### 3.2 System Architecture

The architecture of CodeForge is designed to ensure scalability, modularity, and real-time performance. We architected CodeForge as a distributed, tri-service ecosystem to maintain a clean separation between the user experience and the heavy computational requirements of the underlying AI models. By decoupling user interactions from system logic and computational intelligence, we achieved a modular design that ensures high scalability and robust fault isolation.

1. **The Experience Layer (Frontend):** Built on the React.js framework, this layer serves as the primary student interface. We integrated Three.js to facilitate hardware-accelerated 3D rendering of the "Persona Evolution" system, ensuring that visual transformations remain smooth even on standard hardware.
2. **The Orchestration Layer (Backend):** Developed using Node.js and Express.js, this service acts as the system's "central nervous system." It handles critical operations including JWT-based security protocols, stateful session management, and real-time WebSocket communication for live battles.
3. **The Intelligence Layer (AI Worker):** To handle high-latency tasks, we implemented a dedicated Python-based FastAPI service. This worker is responsible for advanced Natural Language Processing—specifically BART-driven summarization—alongside our automated code evaluation sandbox.
4. **The Persistence Layer (Database):** For data storage, we utilized a MongoDB cluster. The flexible, non-relational schema of MongoDB is essential for managing the heterogeneous data structures required by our various gaming modules, from skill trees to dynamically generated quests.

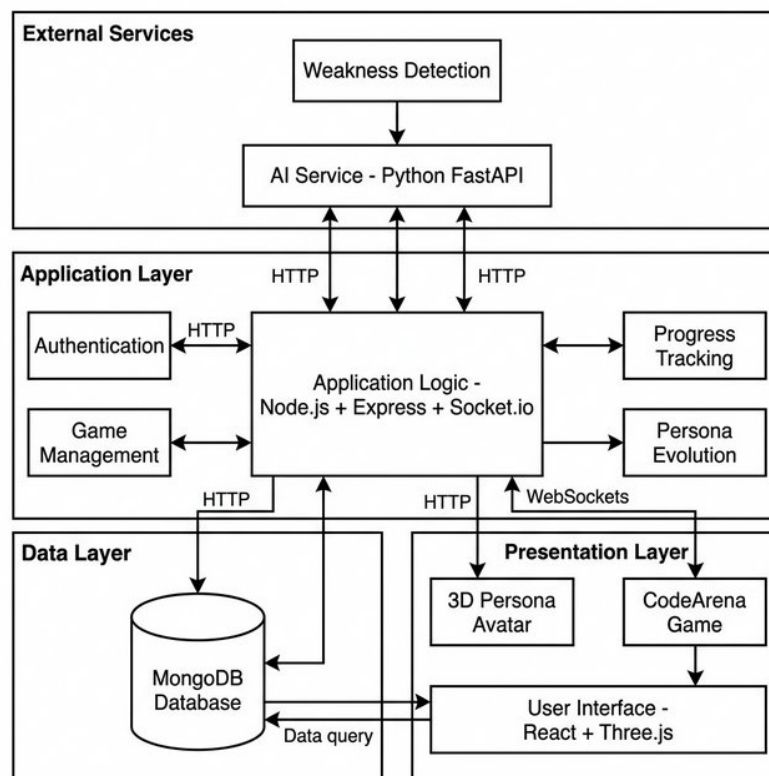


Fig. 1 Multi-layered architecture of the CodeForge platform.

### 3.3 System Workflow

The interaction workflow prioritizes meaningful performance feedback over simple rewards [13]. The system operates through the following sequential steps:

**Cognitive Assessment:** New users undergo an initial assessment to establish their baseline proficiency mapping.

**Personalized Dashboarding:** The system generates a recommended learning path based on assessment results.

**Module Interaction:** Users engage with selected arcade modules targeting specific skill gaps.

**Sandboxed Evaluation:** Submitted code is executed within a secure, isolated sandbox to ensure safe and reliable

evaluation.

**Performance Persistence:** Results are stored as updated skill scores and experience points.

**Persona Evolution:** Rank thresholds trigger the visual transformation of the student's 3D avatar [14].

### 3.4 Core Functional Modules

**Code Arena (1v1 Competitive Mode):** Enables real-time coding competitions using WebSockets for live synchronization.

**Bug Hunter Module:** Focuses on debugging skills through structured, zone-based problem-solving scenarios.

**Memory Matrix:** Enhances cognitive recall through time-based matching of technical concepts.

**Tech Crossword:** Reinforces domain-specific vocabulary using interactive technical puzzles.

**Persona Evolution Engine:** Visualizes user progression through a dynamically evolving 3D avatar.

### 3.5 Algorithm Design

Algorithm 1: Multi-Language Sandboxed Evaluation

Input: studentCode, programmingLanguage, testCases

Output: score, feedback

Procedure:

1. Normalize code and wrap it into a language-specific execution template.
2. Execute the code in an isolated sandbox environment with a fixed 10-second timeout.
3. If execution fails, return score = 0 along with an error message.
4. Initialize passed = 0.
5. For each test case:
  - Execute the code with the given input.
  - If actualOutput == expectedOutput, increment passed.
6. Compute final score:
 
$$\text{Score} = (\text{passed} / \text{totalTestCases}) \times 100$$

Algorithm 2: Weakness Detection Algorithm

Input: skillScores (dictionary of scores)

Procedure:

1. Define thresholds:
  - Low Threshold = 60
  - Medium Threshold = 80
2. For each domain:
  - If score < 60 → Flag as high-priority weakness
  - If  $60 \leq \text{score} < 80$  → Flag as moderate weakness
3. Compute average score:
 
$$\text{Average Score} = (\sum \text{score}_i) / n$$
4. Generate recommendations based on identified weaknesses.

### 3.6 AI-Driven Features

To provide a truly personalized learning journey, CodeForge integrates three distinct layers of machine learning that automate content generation and adapt to user behavior. These features represent a synthesis of supervised and unsupervised techniques designed to maximize student retention [19].

**Unsupervised Topic Discovery (K-Means Clustering):** When raw educational documents are uploaded, the system applies TF-IDF (Term Frequency-Inverse Document Frequency) Vectorization to convert text into mathematical representations. Using these vectors, the K-Means Clustering algorithm identifies latent topic clusters, automatically grouping related concepts (such as "Loops" or "Recursion") without requiring manual labeling by a teacher.

**Behavioral Style Analysis:** The platform provides adaptive feedback by analyzing real-time interaction logs. By monitoring performance metrics—such as solving velocity, the ratio of failed execution attempts, and cognitive recall speed the AI categorizes learners into distinct "Thinking Styles" (e.g., "Optimization Master" or "Precision Trainer"). This analysis allows the system to suggest specific modules that precisely target the student's current psychological and technical needs.

### 3.7 Dataset and Data Sources

The CodeForge platform operates on a multi-dimensional data ecosystem that integrates unstructured curriculum files, a structured repository of algorithmic quests, and real-time behavioral streams. The primary intelligence layer is powered by student-uploaded PDF documents, such as classroom notes and lecture slides, which are automatically transformed into structured learning summaries and multiple-choice questions (MCQs), enabling adaptive and personalized assessment [8]. To ensure a consistent baseline of technical challenges, the platform is further supported by a curated repository of over 200 programming problems, systematically managed within a MongoDB cluster for efficient and scalable retrieval. Finally, the system identifies performance gaps through live interaction logs and continuous behavioral streams that capture user activity during quest sessions, providing immediate data-driven feedback essential for effective learner engagement [14]. This integration of unstructured curriculum data and real-time interaction logs enables a high level of personalization in modern gamified educational systems [20].

### 3.8 Evaluation Metrics

The effectiveness of the CodeForge platform is measured using a multi-dimensional set of quantitative metrics that capture both technical accuracy and cognitive development. The primary indicator is Execution Accuracy (EA), defined as the percentage of test cases successfully passed by a student's code submission. This is calculated as:

$$EA = \frac{\text{Passed Test Cases}}{\text{Total Test Cases}} \times 100$$

Beyond accuracy, the system maintains a Domain Proficiency (DP) score (0–100) for each technical domain. This weighted metric reflects the learner's expertise based on performance consistency and the difficulty levels of completed challenges. Finally, Cognitive Retention is evaluated by tracking improvements in user response time and accuracy across repeated learning sessions within the platform modules. These metrics play a crucial role in evaluating the effectiveness and learning outcomes of the gamified environment [20].

## IV. RESULT AND OUTCOME

The CodeForge platform was successfully implemented as a gamified learning environment designed to enhance programming education. The implementation of the CodeForge platform highlights the effectiveness of a gamified ecosystem in reducing the disconnect between theoretical computer science concepts and practical programming experience. By integrating specialized modules for debugging, algorithmic logic, and technical vocabulary, the system transforms the traditional learning experience into a series of engaging, goal-oriented challenges. Our evaluation focused on measuring the platform's UI responsiveness and its ability to maintain high user engagement through real-time feedback loops.

Central to this experience is the Universal Dashboard, which serves as a live tracking engine for the student's progress. It prioritizes clarity by aggregating metrics such as Experience Points (XP), rank evolution, and activity streaks into an intuitive visual framework. Furthermore, the inclusion of a detailed Skill Matrix allows students to perform granular self-assessments, identifying their technical strengths while highlighting specific domains that require further refinement. This combination of visual rewards and data-driven insights ensures that learners remain motivated while maintaining a clear, measurable trajectory toward skill mastery.

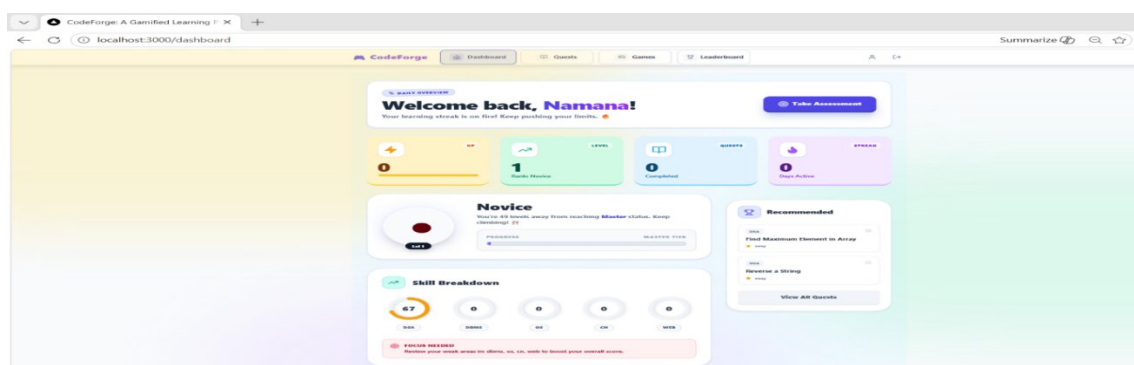


Fig. 2. Dashboard interface displaying user progress, experience points (XP), and skill analytics.

The Code Arena module enables learners to participate in real-time coding challenges where they solve algorithmic problems within a competitive environment. This module encourages learners to improve their problem-solving abilities while competing with other users.

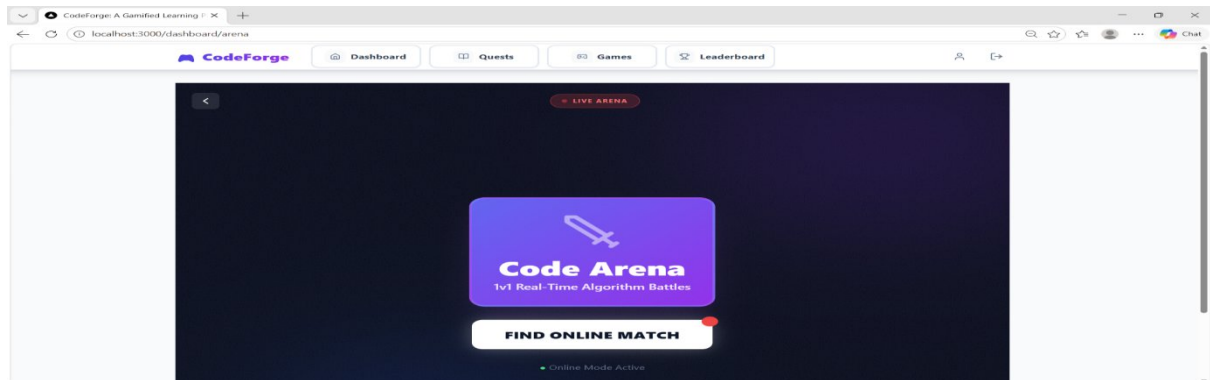


Fig. 3. Code Arena module enabling real-time competitive programming challenges.

The Bug Hunter RPG module focuses on debugging skills. In this module, learners are presented with faulty code snippets and must identify and correct programming errors. The system evaluates the corrected code submitted by users and verifies whether the errors have been successfully resolved.

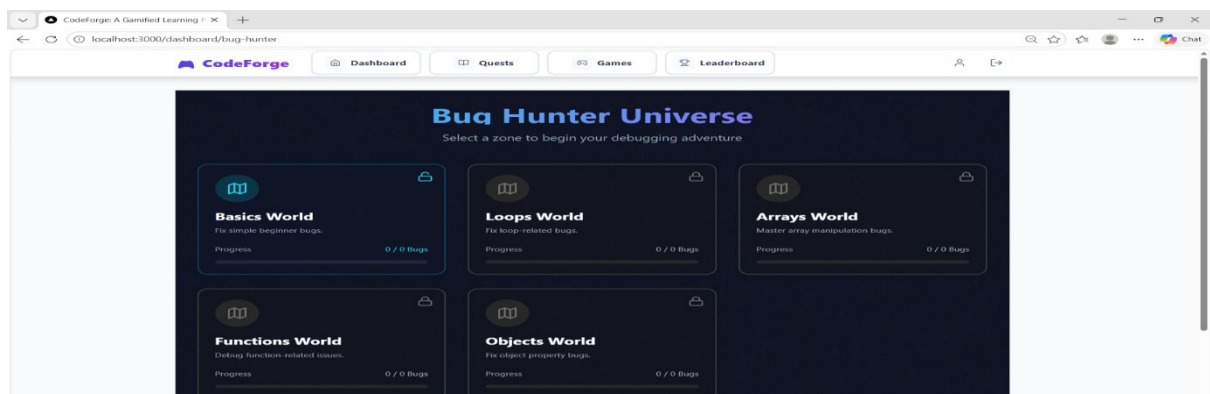


Fig. 4. Bug Hunter RPG module for identifying and fixing programming errors.

The Memory Match module helps reinforce programming concepts by allowing users to match technical concepts with their corresponding definitions. This interactive activity strengthens conceptual understanding and improves knowledge retention.

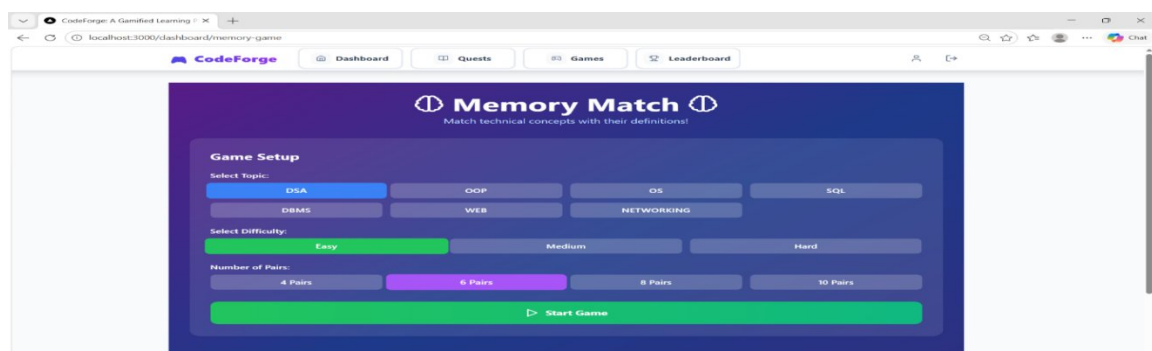


Fig. 5. Memory Match game used to reinforce programming concepts through concept-definition matching activities.

The Technical Crossword module enhances learning by engaging users in puzzle-based problem-solving tasks involving programming terminology. This module helps improve familiarity with technical vocabulary used in computer science.

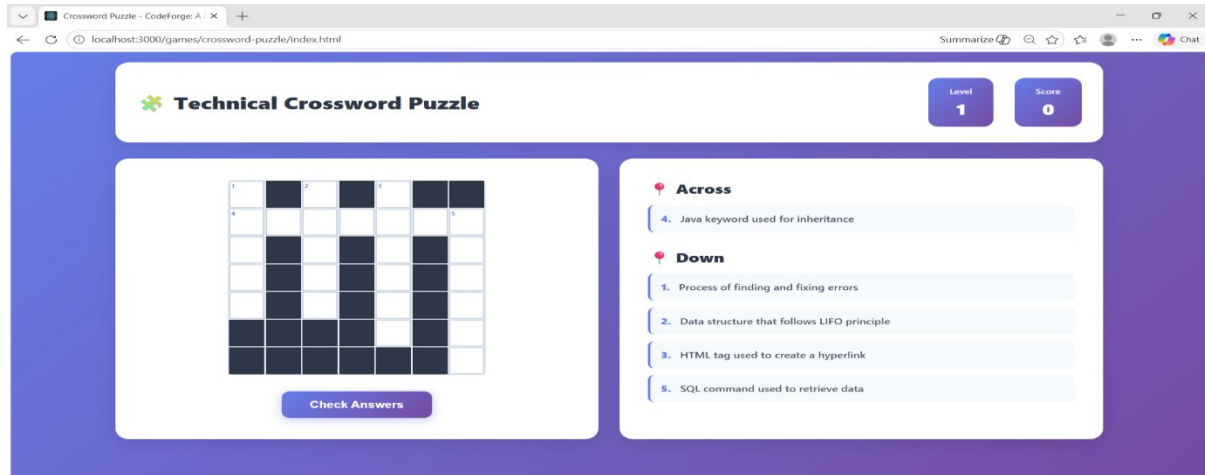


Fig. 6. Technical crossword puzzle module designed to strengthen the understanding of programming terminology.

Overall, the experimental results indicate that the CodeForge platform effectively integrates gamification techniques with programming education. The combination of coding challenges, debugging exercises, memory-based learning activities, and vocabulary reinforcement creates an engaging environment that motivates learners to practice programming concepts continuously.

## V. CONCLUSION AND FUTURE WORK

The development of CodeForge, a gamified learning platform, addresses a critical gap in modern computer science education by shifting learners from passive observation to active, performance-driven engagement. By integrating advanced game mechanics with a robust tri-service architecture, this work demonstrates that technical learning can be both immersive and highly personalized. The platform transforms abstract programming concepts into tangible learning experiences, where progress is visualized through 3D persona evolution and real-time competitive challenges.

Technically, the system integrates BART-based natural language processing to enable automatic course generation, along with a sandboxed evaluation engine designed to support scalable multi-language coding environments. This multidimensional approach-encompassing conceptual understanding, debugging skills, and competitive problem-solving-provides a more comprehensive learning environment than traditional static repositories [18], [19]. Furthermore, the inclusion of measurable indicators such as Execution Accuracy and Domain Proficiency enables a transparent and data-driven representation of learner progress.

Looking ahead, the platform offers strong potential for scalability and further enhancement. Future work will focus on integrating advanced adaptive algorithms to refine personalized learning pathways using real-time cognitive load analysis. Additionally, the PDF-to-course generation framework can be extended to broader technical domains, including Cloud Computing and Artificial Intelligence. Overall, CodeForge demonstrates that the integration of gamification can efficiently enhance student motivation, engagement, and long-term knowledge retention [20].

## REFERENCES

- [1] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining gamification," in Proc. 15th Int. Academic MindTrek Conf., 2011, pp. 9–15.
- [2] K. M. Kapp, The Gamification of Learning and Instruction. San Francisco, CA, USA: Pfeiffer, 2012.
- [3] K. Werbach and D. Hunter, For the Win: How Game Thinking Can Revolutionize Your Business. Philadelphia, PA, USA: Wharton Digital Press, 2012.

**DOI:10.15680/IJARETY.2026.1302072**

- [4] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work?—A literature review of empirical studies on gamification," in Proc. 47th Hawaii Int. Conf. Syst. Sci., 2014, pp. 3025–3034.
- [5] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, "Gamification in education: A systematic mapping study," Educational Technology & Society, vol. 18, no. 3, pp. 75–88, 2015.
- [6] G. Lampropoulos, B. G. Mukta, and T. Anastasiadis, "Impact of gamification on students' learning outcomes and academic performance," Education Sciences, vol. 14, no. 4, 2024.
- [7] M. Jun and T. Lucas, "Gamification elements and their impacts on education: A review," Multidisciplinary Reviews, vol. 8, Art. no. e2025155, 2025.
- [8] S. D. Ristianto, A. Putri, and Y. Rosmansyah, "Personalized gamification for student education: A systematic literature review," IEEE Access, 2025.
- [9] G. Lampropoulos and Kinshuk, "Virtual reality and gamification in education: A systematic review," Educational Technology Research and Development, vol. 72, pp. 1691–1785, 2024.
- [10] F. Gini, S. Bassanelli, F. Bonetti, R. H. Mogavi, A. Bucchiarone, and A. Marconi, "The role and scope of gamification in education: A scientometric literature review," Acta Psychologica, vol. 259, Art. no. 105418, 2025.
- [11] G. Lampropoulos, B. G. Mukta, and T. Anastasiadis, "Gamification in learning management systems: A systematic literature review," Information, vol. 16, Art. no. 1094, 2025.
- [12] J. Koivisto and J. Hamari, "The rise of motivational information systems: A review of gamification research," International Journal of Information Management, vol. 45, pp. 191–210, 2019.
- [13] S. Nicholson, "A user-centered theoretical framework for meaningful gamification," in Proc. Games+Learning+Society Conf., 2015.
- [14] H. Ukgoda, "Gamification in education: Its impact on engagement, motivation, and learning outcomes," Journal of Educational Technology Development and Exchange, vol. 18, no. 3, pp. 41–66, 2025.
- [15] A. Dela Peña, "Digital learning platforms in aircraft maintenance technology: Evaluating the effectiveness of gamified learning approaches," Research in Learning Technology, vol. 33, 2025.
- [16] M. H. Al-Khreshah, "The cognitive and motivational benefits of gamification in English language learning: A systematic review," The Open Psychology Journal, vol. 18, 2025.
- [17] A. Alnuaim, "The impact and acceptance of gamification by learners in a digital literacy course at the undergraduate level: Randomized controlled trial," JMIR Serious Games, vol. 12, Art. no. e52017, 2024.
- [18] J. R. Gómez Niño, L. P. Arias Delgado, A. Chiappe, and E. Ortega González, "Gamifying learning with AI: A pathway to 21st-century skills," Journal of Research in Childhood Education, vol. 39, no. 4, pp. 735–750, 2025.
- [19] A. Domínguez, J. Saenz-de-Navarrete, L. de-Marcos, L. Fernández-Sanz, C. Pagés, and J. J. Martínez-Herráiz, "Gamifying learning experiences: Practical implications and outcomes," Computers & Education, vol. 63, pp. 380–392, 2013.
- [20] S. Subhash and E. A. Cudney, "Gamified learning in higher education: A systematic review," Computers in Human Behavior, vol. 87, pp. 192–206, 2018.

## International Journal of Advanced Research in Education and Technology

**ISSN: 2394-2975**

**Impact Factor: 8.152**