AI Based Gamified Educational Application

[1]Prof.S.P.Gunjal, [2]Swejal Vairagade, [3]Dnyaneshwari Thopate, [4]Aryan Bhagat
[1]Associate Professor, Department of Computer Engineering, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra
[2,3,4]U.G. Student, Department of Computer Engineering, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra

Abstract: Traditional programming education often poses engagement challenges, leading to learner fatigue, uneven progress, and reduced conceptual retention among students. This study introduces CodeVerse - Learning with Playing, an AI-driven gamified learning platform designed to revolutionize programming education through interactive, adaptive, and game-based techniques. The system integrates gamification and adaptive learning models to provide an immersive environment where users learn programming concepts through quizzes, coding challenges, and level-based modules. CodeVerse's architecture employs Next.js and TypeScript for dynamic front-end rendering, Supabase for real-time data handling, and AI-powered analytics to personalize learning paths. The system's core features include a Gamification Engine that awards badges and points, and an Adaptive Engine that monitors learner performance to adjust content difficulty. This approach ensures that students remain motivated, receive personalized feedback, and progress according to their capability. The paper evaluates the platform's design, implementation, and testing results, demonstrating its potential to deliver an engaging, scalable, and effective web-based learning ecosystem for programming education.

Keywords:- Gamified Learning , Adaptive Learning Platform , Web Application, Programming Education ,Student Engagement , Educational Technology.

I. INTRODUCTION

The increasing demand for effective, engaging, and scalable programming education has emphasized the need for modernized learning platforms for beginners and students. Traditional coding instruction, such as classroom lectures, textbooks, or static coding exercises, is often rigid, non-interactive, and fails to sustain learner motivation. These challenges result in low engagement, slower skill acquisition, and limited practical problem-solving experience. Hence, there is a growing requirement for an interactive, gamified, and adaptive digital solution to teach programming concepts efficiently.

The CodeVerse - Learning with Playing addresses these challenges by introducing a web-based, gamified platform that combines coding exercises with interactive game mechanics. The system ensures that learners progress through structured coding levels, receive instant feedback, and reinforce learning via play-based challenges. Core functionalities include interactive coding tasks, real-time code execution, hint-based guidance, performance tracking, and gamified rewards to maintain motivation and engagement. Adaptive feedback mechanisms ensure learners receive personalized support according to their progress and skill level, minimizing frustration and promoting continuous improvement .Technologically, the proposed system integrates Next.js and Tailwind CSS for a responsive and intuitive frontend, while Supabase handles backend services such as user authentication, database management, and progress tracking. Embedded code editors and sandboxed execution environments provide learners with real-time feedback on their coding solutions. This architecture not only ensures a seamless learning experience but also supports scalability and accessibility across devices.

The proposed solution aligns with the vision of enhancing digital education by making programming accessible, engaging, and effective for learners. It provides interactive learning, instant feedback, and gamified motivation to improve learner autonomy and coding literacy. By integrating gamification and web-based interactivity into coding education, the CodeVerse system contributes to building an engaging, scalable, and effective learning model. The rest of this paper is organized as follows: Section II discusses related work; Section III explains system design and methodology; Section IV presents implementation and results; and Section V concludes with findings and future scope.

II. LITERATURE SURVEY

A range of research efforts has been carried out on web-based learning platforms, gamification in education, and interactive coding environments. This section reviews significant contributions, identifies existing challenges, and highlights the research gap that motivates the development of CodeVerse - Learning with Playing.

Web-based coding and educational platforms: Several studies have focused on creating web-based platforms for teaching programming and computational thinking. Platforms such as Codecademy, Code.org, and LeetCode provide structured exercises, tutorials, and challenges to learners. These systems simplify access to programming education and allow learners to practice coding online. However, most of these platforms rely on standard exercises with limited gamification elements, which can reduce sustained engagement and motivation over time.

Gamification and interactive learning models: Researchers have recently explored gamification for improving learning outcomes in computer science education. Studies by Deterding et al. and Hamari et al. highlight how game elements like points, levels, badges, and immediate feedback can increase learner motivation, engagement, and retention. While these studies demonstrate gamification's potential to enhance learning, many existing platforms implement it only partially or for specific topics, without integrating a complete, structured, and progressive coding curriculum.

Adaptive learning and feedback mechanisms: Some works implement adaptive feedback and interactive coding editors to personalize learning experiences and guide learners through challenges. These systems improve learning efficiency by providing hints, error detection, and dynamic difficulty adjustments. However, deployment often requires advanced technical infrastructure or is limited to desktop-based solutions, making accessibility on mobile devices and seamless user experience challenging.

Comparative observations and gaps: Most existing coding platforms are either static, lack structured gamification, or do not provide fully interactive, adaptive learning experiences. While gamified approaches improve engagement, few integrate a progressive curriculum with real-time code execution, immediate feedback, and reward systems in a single, web-accessible interface.

Positioning of the present work: The proposed CodeVerse – Learning with Playing bridges these gaps by integrating gamification, interactive coding challenges, real-time feedback, and a web-based interface accessible across devices. It ensures a progressive and engaging learning experience, helping learners develop coding skills efficiently while maintaining motivation. Compared to earlier platforms, it offers enhanced interactivity, structured progression, and adaptive feedback mechanisms, supporting the vision of accessible, effective, and engaging digital education for beginners and students.

III. METHODOLOGY

The proposed system is developed using an Agile development approach to ensure flexibility, iterative improvement, and continuous incorporation of user feedback. The research begins with an analysis of the existing challenges in programming education, such as low learner engagement, difficulty in understanding abstract concepts, lack of instant feedback, and limited motivation in traditional learning methods.

A web-based gamified framework is adopted to enhance engagement, interactivity, and adaptive learning. The system is implemented using technologies such as Next.js and Tailwind CSS for frontend development, Supabase for backend services including authentication

and database management, and embedded code editors for real-time code execution and assessment.

Gamification mechanics are designed to structure the learning experience, including progressive levels, points, badges, and adaptive hints. Learner actions, such as solving coding challenges or completing levels, are tracked and used to provide personalized feedback and progression recommendations. All user interactions, progress data, and achievements are securely stored in the backend database to ensure consistency and persistence.

The initial “analysis of existing processes” involved a qualitative assessment of beginner learning workflows through surveys, literature review, and user testing of standard coding platforms. We mapped typical learner pain points, including delayed error feedback, lack of structured progression, and insufficient motivation. Based on this analysis, distinct user roles were defined: “Learner” and “Administrator,” each with specific functional requirements such as challenge progression, performance tracking, and content management.

Following the Agile methodology, the system’s development was structured into distinct sprints, each focusing on core functional modules. The foundational sprints targeted user authentication, learner dashboard, and challenge modules to establish a seamless, interactive interface. Subsequent sprints iteratively built the gamification engine, real-time code execution module, hints and feedback system, and progress tracking components. This approach allowed the development team to refine each module based on user feedback while ensuring seamless integration between the frontend interface and backend services, resulting in an adaptive and engaging learning platform.

IV. SYSTEM DESIGN

The system architecture integrates the Presentation, Application, and Gamification layers to establish a seamless, interactive, and adaptive coding education ecosystem. The system is designed around a learner-centric model, focusing on engagement, instant feedback, and structured progression. While the user flow is role-based—distinguishing between Learners and Administrators—the underlying workflow ensures real-time execution, assessment, and progression tracking, reflecting the project’s core philosophy of interactive and gamified learning. This architecture facilitates two primary, parallel workflows:

Learner Challenge Flow: This workflow provides an interactive and structured learning process. The learner initiates this flow by interacting with the Next.js-based Presentation Layer to select and attempt coding challenges. Their inputs are captured by the Application Layer, which validates the code, executes it in the embedded sandbox environment, and provides instant feedback on correctness, efficiency, and errors. Successful completion updates the learner’s progress, awards points or badges, and unlocks the next level of challenges.

Administrative/Content Management Flow: In parallel, this workflow manages content creation and performance monitoring. Administrators, using their role-based credentials via the same web portal, create and update challenges, monitor learner progress, and configure gamification elements such as points, badges, and adaptive hints. The Application Layer processes these actions, ensuring that changes are reflected in the learner dashboards in real time.

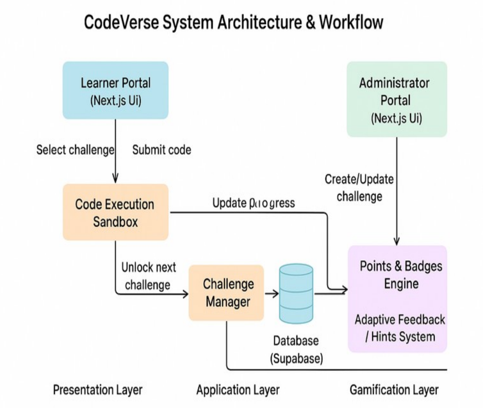

Fig.1-System Architecture Diagram

This dual-flow architecture culminates in an interactive, adaptive, and gamified learning experience, where learner actions are immediately assessed and rewarded, while administrators maintain full control over content, difficulty progression, and performance analytics. The design ensures scalability, responsiveness, and engagement, providing a robust framework for effective coding education.

V. PROPOSED SYSTEM AND RESULTS

The proposed CodeVerse - Learning with Playing system is a secure, web-based platform designed to gamify and digitize the learning process for programming and software concepts. It addresses the common issues of low engagement, lack of interactivity, and difficulty in conceptual understanding found in traditional text-based or lecture-based learning methods by leveraging a three-tier architecture. This architecture features a React-based Presentation Layer for user interaction, a Node.js Application Layer for game and challenge logic, and a MongoDB Data Layer for secure storage of user progress and analytics. The system integrates gamification elements such as points, badges, and levels to motivate learners and track achievements, ensuring that every user action is recorded and progress is persistently stored.

Upon successful development and testing, the system validated all primary functional objectives. The end-to-end flow for learning modules was confirmed: a learner's challenge attempt, upon evaluation by the system, automatically triggered reward points and badge assignments, which were securely recorded in the database. Similarly, the interactive feedback module proved to be fully effective, with each user action producing immediate visual and textual feedback, making the learning process engaging and transparent. The successful implementation of these modules demonstrates that the system effectively achieves its core goals of enhanced engagement, gamified learning, and progressive knowledge acquisition, proving the viability of using interactive web-based platforms for improving programming education.

VI. REFERENCES

[1] D. A. Norman, The Design of Everyday Things, Basic Books, 2013.

[2] K. Squire, “Video Games and Learning: Teaching and Participatory Culture in the Digital Age,” Teachers College Record, vol. 110, no. 5, pp. 200–205, 2008.

[3] C. Deterding, D. Dixon, R. Khaled, and L. Nacke, “From Game Design Elements to Gamefulness: Defining Gamification,” Proceedings of the 15th International Academic MindTrek Conference, pp. 9–15, 2011.

[4] J. Hamari, J. Koivisto, and H. Sarsa, “Does Gamification Work? – A Literature Review of Empirical Studies on Gamification,” 2014 47th Hawaii International Conference on System Sciences, pp. 3025–3034, 2014.

[5] R. Mayer, Multimedia Learning, 2nd edition, Cambridge University Press, 2009.

[6] B. Fitzgerald, “Gamified Learning Platforms for Programming Education: A Review,” International Journal of Educational Technology in Higher Education, vol. 17, no. 1, pp. 1–23, 2020.

[7] M. Resnick et al., “Scratch: Programming for All,” Communications of the ACM, vol. 52, no. 11, pp. 60–67, 2009.

[8] S. Papastergiou, “Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation,” Computers & Education, vol. 52, no. 1, pp. 1–12, 2009.

[9] Codeverse Studio, “How Coding Encourages Students' Creativity and Independence,” Blog, 2023.

[10] Codeverse Studio, “Learning through Play: Mission-Based Gamified Coding,” Blog, 2022. Available at: https://www.codeverse.com/blog-post/mission-based-gamified-coding