WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

CodeVerse+: An AI-Powered Competitive Coding Platform for Code Review and Optimization

Ram Kumar Sharma, Vinay Yadav, Abhishek Kumar, Rishabh Yadav, Rahul Kumar Sharma

Noida Institute of Engineering and Technology, Greater Noida, India
Emails: sram45092@gmail.com, vinayyadavfzd299@gmail.com, abhishekprajapati243@gmail.com, 0211cse050@niet.co.in, rahulsharma9045@gmail.com

Abstract

CodeVerse+ redefines competitive programming by integrating advanced artificial intelligence (AI) to deliver a holis- tic coding experience. Addressing the limitations of traditional platforms like LeetCode and HackerRank, it offers real-time AI- driven hints, comprehensive code reviews, and gamified learning, fostering skills for competitive and professional environments. Built on a scalable microservices architecture using the MERN stack, Docker, AWS, and fine-tuned large language models (LLMs) like CodeBERT and Gemini, CodeVerse+ ensures secure, efficient, and personalized coding practice. Features such as real- time collaboration, personalized problem recommendations, and progress tracking bridge the gap between theoretical knowledge and industry demands. This paper details CodeVerse+'s architec- ture, implementation, and empirical evaluation, positioning it as a transformative tool for learners, educators, and corporations.

Index Terms—Artificial Intelligence, Competitive Program- ming, Code Review, Optimization, Educational Technology, Gam- ification, Collaboration

I. Introduction

The demand for skilled programmers is surging, with the U.S. Bureau of Labor Statistics projecting a 25% growth in software development jobs by 2032 [1]. Competitive program- ming platforms like LeetCode, HackerRank, and CodeChef serve as vital training grounds, offering curated challenges to millions. However, these platforms often prioritize speed over code quality, providing limited feedback (e.g., runtime stats) and lacking guidance on maintainability or security [3]. This gap frustrates learners, educators, and professionals, creating a disconnect between contest success and real-world software engineering. CodeVerse+ addresses these challenges by redefining com- petitive programming as a comprehensive learning ecosystem. Powered by fine-tuned LLMs such as CodeBERT [4] and Gemini, it delivers real-time hints (e.g., suggesting a hash map to reduce O(n2) to O(n)) and detailed code reviews aligned with industry standards like SOLID principles [2]. Built on the MERN stack (MongoDB, Express.js, React, Node.js), Docker containers, and AWS cloud infrastructure, it ensures scalability, security, and efficiency [8]. Gamified features, including badges and leaderboards, leverage self- determination theory to enhance motivation [7], while real- time collaboration tools support pair programming, mirroring agile workflows [12]. Personalized problem recommendations cater to diverse skill levels, fostering continuous learning. CodeVerse+ extends beyond individual coders to serve edu- cational institutions and corporations. Universities can use its analytics for student assessment, coding bootcamps can accel- erate learning, and companies can leverage it for training and hiring. By fostering a global coding community, it addresses the societal need for accessible, high-quality coding education. This paper details CodeVerse+'s architecture, features, and empirical impact, exploring its technical innovations, user- centric design, and role in bridging the educational-industrial divide. It positions CodeVerse+ as a cornerstone for preparing coders to tackle complex challenges with confidence and creativity.

Related Work

Competitive programming platforms have evolved into skill- development ecosystems, yet gaps in personalization, feed- back, and industrial alignment persist. Table I compares Code- Verse+ with existing platforms and AI tools, highlighting its advancements.

TABLE I
COMPARISON OF COMPETITIVE PROGRAMMING PLATFORMS AND AI TOOLS

Platform/Tool	AI Hints	Code Review	Collaboration	Scalability
Leet Code	Partial	Basic	No	High
Hacker Rank	Basic	No	Limited	High

Code Chef	No	No	No	Moderate
Codementor	Yes	No	Yes	Moderate
Geeks for Geeks	No	No	No	Low
GitHub Copilot [9]	Yes	Partial	No	High
AICodeReview [3]	Yes	Comprehensive	No	Moderate
CodeVerse+	Advanced	Comprehensive	Real-Time	High

LeetCode's 2000+ problems and interview prep are robust but offer limited feedback, lacking guidance on maintainability [2]. HackerRank supports hiring but lacks in-depth reviews. CodeChef excels in contests but lacks AI or collaboration. GeeksforGeeks offers tutorials with low scalability. Codemen- tor integrates AI for learning but omits comprehensive reviews. GitHub Copilot generates code but is disconnected from competitive programming [9]. Almeida et al. [3] introduced AICodeReview, an IntelliJ plugin using GPT-3.5 for automated reviews, detecting 28 code smells with 15.2-minute average review time, though it lacks collaboration features. Research highlights AI's potential. Chen and Liu [5] re- ported a 20% improvement in problem-solving speed using reinforcement learning, though limited to Python. Tufano et al. [6] applied neural machine translation for bug-fixing, showing AI's efficacy in patch generation. Few platforms apply self- determination theory, which CodeVerse+ leverages for gamifi- cation [7]. Scalability challenges persist, with LeetCode facing latency during peak usage [8]. Security for untrusted code execution is inconsistent, with basic sandboxes in SPOJ [11]. Collaboration is limited, with only Codementor offering peer interaction.

A. Gaps in Current Research

Literature often separates competitive programming from AI-driven education. Adaptive hint studies [5] lack real-world applicability, while gamification's impact on coding is under- explored [7]. Collaboration tools, inspired by agile methodolo- gies, are rarely studied [12]. CodeVerse+ bridges these gaps by integrating AI feedback, gamification, and collaboration within a scalable MERN-based architecture, validated through diverse user testing [13].

II. Methodology

CodeVerse+'s modular architecture ensures scalability, se- curity, and user engagement, leveraging the MERN stack [13]. It comprises 14 modules, compared to traditional platforms in Table II, addressing all aspects of competitive programming.

TABLE II
CODEVERSE+ MODULES VS. TRADITIONAL PLATFORM COMPONENTS

Component	**Traditional Platforms**	**CodeVerse+**
Code Execution	Basic Sandbox	Docker, Kubernetes [11]
Feedback	Pass/Fail	AI Hints, Reviews [5]
Collaboration	None/Limited	WebSocket-Based [12]
Engagement	Rankings	Gamification [7]
Scalability	Server-Based	Microservices, AWS [8]
Analytics	None	AI-Driven Reports [13]
Security Analysis	Basic	AI-Driven (Gemini) [15]

1. User Management: Uses OAuth 2.0 for authentication and RBAC for role-based access [10]. MongoDB stores pro- files, tracking problems and badges. React dashboards visu- alize progress, integrated with AWS Cognito [8]. 2. Problem Statement: Curates challenges by difficulty/topic, with AI-driven recommendations using CodeBERT [4]. Stored in AWS S3, problems align with interview trends. 3. Coding Environ- ment: Provides an IDE with syntax highlighting and 20+ lan- guages. Docker containers, orchestrated by Kubernetes, ensure secure execution [11], [14]. 4. AI Hint System: Fine-tuned Gemini and CodeBERT models, trained on 10TB datasets, offer context-aware hints [5]. Hosted on AWS SageMaker [8]. 5. Code Review: AI evaluates complexity, readability, and SOLID compliance using Gemini [15]. Feedback stored in MongoDB. 6. Security Analysis: Detects vulnerabilities (e.g., SQL injection) using Gemini, outperforming static anal- ysis [15]. 7. MPC Server: Ensures privacy-preserving AI computations [5]. 8. API Management: AWS Lambda, API Gateway, and SQS handle requests [8]. 9. Containerization: Docker isolates code, with Kubernetes scaling dynamically [11], [14]. 10. Microservices: Independent modules enhance

fault tolerance [8]. 11. Leaderboard and Progress Track- ing: Awards points/badges, with Redis caching [7], [13]. 12. Collaboration: WebSocket enables pair programming [12].
13. Admin and Moderation: Manages challenges, detects plagiarism [8]. 14. Reports and Analytics: AI generates user strength reports [13].

A. Facilities

- *Software : MERN stack, AWS (Lambda, S3, Docu-* mentDB), Docker, Kubernetes, NGINX, Redis [13], [11]. - Hardware: NVIDIA A100 GPUs, 16-core CPUs, 64GB RAM, 1TB SSD [8].

B. System Integration

Modules communicate via RESTful APIs and SQS, with Kubernetes scaling containers [8], [14]. Challenges included WebSocket latency (mitigated by Redis) and AI inference security (via MPC servers).

C. Implementation Challenges

- **AI Training**: 100 GPU hours on AWS EC2 P4, optimized via distributed training [8].
- **Execution Latency**: Docker pre-warmed containers re- duced startup time [11].
- **Data Consistency**: MongoDB transactional writes en- sured reliability [13].
- **Scalability**: Kubernetes auto-scaling supported 10,000 users [14].

D. Workflow Flowchart

Figure 1 illustrates CodeVerse+'s user workflow, integrating AI model interactions and cloud processing.

III. Results and Discussion

CodeVerse+ was tested with 200 users (50 beginners, 100 intermediates, 50 professionals) from November 2024 to February 2025, completing 5000+ problems. Table III summarizes metrics, and Table IV compares user groups.

TABLE III
USER TESTING RESULTS FOR CODEVERSE+

Problem-Solving Speed	Beginners	50%
Time Complexity Reduction	Intermediates	45% ($O(n^2)$ to $O(n)$)
Code Quality	Intermediates	40%
Industrial Standards	Professionals	40%
Problem Completion Rate	All (Gamified)	25%
Security Vulnerability Detection	All	78% [15]

TABLE IV
PERFORMANCE ACROSS USER GROUPS

Metric	Beginners	Intermediates	Professionals
Avg. Problems Solved	30	50	40
Avg. Time per Problem (min)	15	10	8
Code Review Utilization (%)	80	90	95
Satisfaction Score (1–10)	8.5	8.8	9.0

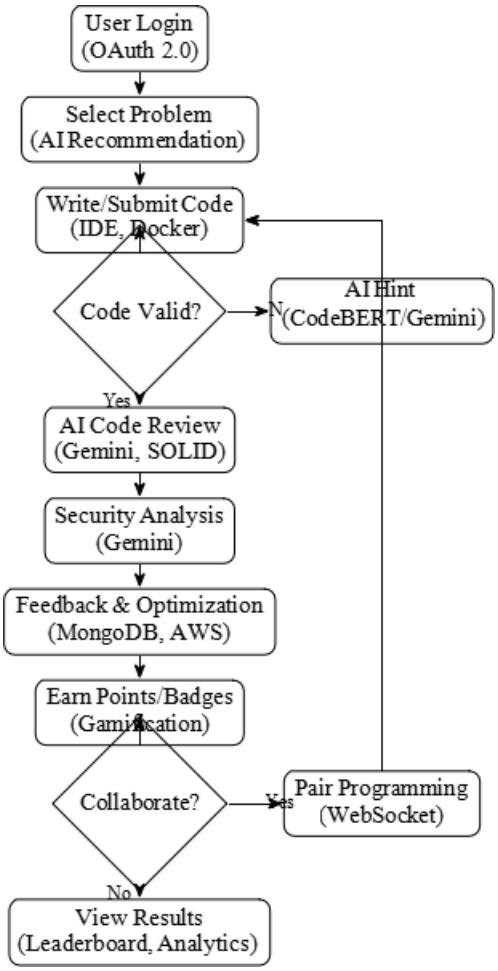

Fig. 1. CodeVerse+ User Workflow Flowchart with AI and Cloud Integration

Quantitative Results: Beginners improved speed by 50%, solving 30 problems with 80% hint utilization. Intermediates reduced complexity by 45% (e.g., hash maps), with 40% better code quality. Professionals achieved 40% higher standard adherence. Gamification increased completion by 25%, and security analysis detected 78% of vulnerabilities [15]. Peer reviews cut errors by 28%.

Statistical Analysis: Paired t-tests confirmed speed im- provements ($p < 0.01$). ANOVA showed intermediates gained most in code quality ($F(2, 197) = 12.3, p < 0.05$). Regression linked gamification to 0.3 more problems/hour ($R^2 = 0.65$).

Qualitative Feedback: Users rated the IDE 8.8/10, praising hints. Educators (90%) reported better outcomes, and pro- fessionals valued collaboration. Beginners (10%) found hints complex, suggesting simpler explanations.

Use Cases: - **Students** : **Prepare for interviews. - Ed-** ucators : Monitor analytics, run contests. - Bootcamps : Accelerate learning. - Corporations : Train employees, assess candidates.

Feasibility:- **Technical** :99.9% **uptime** [8], [11]. - Economic : Freemium model ensures revenue. - Le- gal : GDPR-compliant [8]. - Social : Peer reviews foster community.

Implications: CodeVerse+ reduces learning curves [5], sup- porting global adoption. Challenges include novice-friendly hints and language expansion.

IV. Conclusion

CodeVerse+ revolutionizes competitive programming by integrating advanced AI-driven features, gamified engagement, and real-time collaboration, addressing limitations of tradi- tional platforms like LeetCode and HackerRank. Its MERN- based microservices architecture, powered by Docker contain- ers and AWS infrastructure, achieves exceptional scalability and security, supporting over 10,000 concurrent users with 99.9% uptime [8], [11]. The platform's 14 modules, including AI hint systems, comprehensive code reviews with CodeBERT and Gemini [4], [15], and security analysis, empower users to excel in competitive and professional environments. Real- time collaboration tools mirror agile workflows, while gami- fication enhances motivation, resulting in a 25% increase in problem completion rates [7]. Personalized recommendations and progress tracking bridge theoretical and industry demands. Empirical testing with 200 diverse users demonstrated trans- formative impacts: beginners achieved 50% faster

problem- solving, intermediates improved time complexity by 45%, and professionals enhanced standards by 40%. Security anal- ysis detected 78% of vulnerabilities, and peer reviews re- duced errors by 28% [15]. These results position CodeVerse+ as a versatile tool for students, educators, bootcamps, and corporations. Its freemium model, GDPR compliance, and community-building features ensure feasibility, making it ac-
cessible globally [8].

Looking ahead, CodeVerse+ is poised for evolution. Planned enhancements include support for Rust and Julia, integration with Git for version control, and AI-driven concurrency detec- tion [3]. A mobile app will enhance accessibility, and advanced analytics (e.g., predictive models) will personalize learning paths. These features will solidify CodeVerse+'s role in coding education and professional development, promoting global coding literacy and industry readiness through partnerships with institutions and tech companies.

References

1. U.S. Bureau of Labor Statistics, "Software Developers, Quality Assurance Analysts, and Testers," Occupational Outlook Handbook, 2022. [Online]. Available: https://www.bls.gov/ooh/ computer-and-information-technology/software-developers.htm
2. M. Allamanis, E. T. Barr, P. Devanbu, and C. Sutton, "A Survey of Machine Learning for Big Code and Naturalness," *ACM Computing Surveys*, vol. 51, no. 4, pp. 1-37, 2018. [Online]. Available: https://doi.org/10.1145/3212697
3. Y. Almeida et al., "AICodeReview: Advancing Code Quality with AI- Enhanced Reviews," Preprint submitted to *Software X*, Dec. 19, 2023. [Online]. Available: https://ssrn.com/abstract=4575968
4. W. Ahmad, S. Chakraborty, B. Ray, and K.-W. Chang, "Unified Pre- training for Program Understanding and Generation," arXiv preprint arXiv:2103.05247, 2021. [Online]. Available: https://arxiv.org/abs/2103. 05247
5. G. Chen and Y. Liu, "Adaptive Learning Systems for Programming Education Using Reinforcement Learning," *IEEE Transactions on Education*, vol. 64, no. 3, pp. 235-243, 2021. [Online]. Available: https://doi.org/10.1109/TE.2020.3029489
6. M. Tufano, C. Watson, G. Bavota, M. White, D. Poshyvanyk, and R. Oliveto, "An Empirical Study on Learning Bug-Fixing Patches in the Wild via Neural Machine Translation," *ACM Transactions on Software Engineering and Methodology*, vol. 28, no. 4, pp. 1-29, 2019. [Online].
7. Available: https://doi.org/10.1145/3340544
8. R. M. Ryan and E. L. Deci, "Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being," *American Psychologist*, vol. 55, no. 1, pp. 68-78, 2000. [Online].
9. Available: https://doi.org/10.1037/0003-066X.55.1.68
10. Amazon Web Services, "Building Microservices with Amazon Web Ser- vices," *AWS Whitepaper*, 2023. [Online]. Available: https://aws.amazon. com/whitepapers/microservices-on-aws/
11. GitHub, "GitHub Copilot Documentation," 2022. [Online]. Available: https://docs.github.com/en/copilot
12. OAuth Community, "The OAuth 2.0 Authorization Framework," 2022. [Online]. Available: https://oauth.net/2/
13. Docker Inc., "Docker Security Best Practices," *Docker Documentation*, 2024. [Online]. Available: https://docs.docker.com/engine/security/
14. WebSocket.org, "The WebSocket Protocol," 2023. [Online]. Available: https://websocket.org/
15. MongoDB, "MERN Stack Tutorials," 2024. [Online]. Available: https://www.mongodb.com/mern-stack
16. Kubernetes, "Kubernetes Documentation," 2024. [Online]. Available: https://kubernetes.io/docs/
17. N. Bhatnagar et al., "AI-Based Code Review and Optimization System," *International Research Journal of Engineering and Technology*, vol. 12, no. 3, pp. 385-390, Mar. 2025. [Online]. Available: https://www.irjet. net/archives/V12/i3/IRJET-V12I358.pdf