International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

# Code Quest: Gamified Platform to learn Programming.

Nikhil Joshi
Department of Computer Engineering
Atharva College of Engineering
Mumbai, India
joshinikhil-cpmn@atharva.coe.ac.in

Drishti Malavade
Department of Computer Engineering
Atharva College of Engineering Mumbai,
India
malavadedrishti-cmpn@atharvacoe.ac.in

Surveshkumar Chourasia Department of Computer Engineering
Atharva College of Engineering Mumbai,
India
chourasiasurveshkumar-cmpn@atharvacoe. ac.in

Suyash Mane
Department of Computer Engineering Atharva College of Engineering Mumbai, India
manesuyash-cpmn@atharva.coe.ac.in

Dr. Suvarna Pansambal
Department of Computer Engineering
Atharva College of Engineering Mumbai, India
suvarnashirke@atharva.coe.ac.in

***Abstract — Code Quest*** **is an innovative platform that blends coding education with gamification to make programming more accessible, engaging, and rewarding. Designed for beginners and students, it integrates structured learning and interactive gameplay to create a balanced environment that encourages consistent practice and growth. Participants (users) gain or lose points according to their performance, with a real-time leaderboard fostering healthy competition and motivation.**

**The platform prioritizes core programming skills such as syntax retention, problem-solving, and logical thinking, while also promoting user engagement through game-based reinforcement. By removing traditional barriers to entry and introducing coding in a fun and less intimidating manner,** ***Code Quest*** **nurtures curiosity, builds confidence, and inspires a sustained interest in technology.**

**Through its intuitive design and learner-centered approach,** ***Code Quest*** **redefines how coding is introduced to new learners, helping them develop technical fluency in an enjoyable and accessible way. It is adaptable across diverse learning styles, combining play with pedagogy which makes it very suitable for different types of learners and offers both self-paced and classroom environments.**

***Keywords:*** ***Programming Education, Gamification, eLearning, Syntax Retention, Coding Challenges, Leaderboard System, Interactive Learning, Beginner-Friendly Coding***

## I. INTRODUCTION

In today's rapidly evolving digital world, programming has become an essential skill across various domains. However, for many beginners, particularly those with no or minimal previous exposure learning to code can feel intimidating, complex, and disengaging. Traditional teaching methods often fail to hold learners' attention or offer the kind of interactive feedback required to build confidence and long-term understanding. To bridge this gap, there is a growing need for innovative educational tools that make programming more approachable, enjoyable, and effective.

**Gamification** has emerged as a promising strategy to enhance learner motivation and retention by transforming abstract concepts into engaging, hands-on experiences. By incorporating

game elements such as points, progression, and challenges, educational platforms can significantly improve **user involvement** and **consistency**. This approach not only supports the acquisition of technical skills but also fosters **critical thinking** and **problem-solving abilities** in a dynamic learning environment.

*Code Quest* was developed to address the need for more engaging and accessible coding education. By combining structured lessons, coding tasks, quizzes, and gamified reinforcement, it provides a well-rounded learning experience tailored to diverse learning styles and paces. The platform emphasizes not just syntax and logic, but also critical thinking, problem-solving, and regular practice through point systems and leaderboards.

It features two main components: a Learning section focused on foundational concepts and assessments, and a Gaming section designed to reinforce retention through interactive mini-games. This dual approach supports learners from basic to intermediate levels, building both confidence and competence without the pressure of traditional methods.

By simplifying entry into programming and encouraging active engagement, Code Quest supports inclusive, skill-driven learning and contributes to building a more digitally literate generation.

## II. REVIEW OF LITERATURE

By incorporating game-like elements into the learning process, gamified learning platforms have drastically changed professional programming education and improved learner engagement, motivation, and retention. This method uses interactive scenarios, challenges, and rewards to make difficult programming concepts more approachable and entertaining. Learners are more likely to stay consistent and overcome their apprehension of failure when success is made to depend on concrete incentives and visual feedback. Moreover, the sense of achievement gained through completing levels or earning points can boost self-confidence and encourage continuous learning.

### *A. Gamification in Education: What, How, Why Bother?*

In their 2017 study, Dichev and Dicheva examined the application of gamification in education, answering the basic queries of what gamification is, how it can be used, and why it is used. They warn against a superficial approach that might not produce the intended educational benefits and emphasize that careful design is necessary for effective gamification in order to match game elements with learning objectives.

### *B. A Gamified Learning Environment to Enhance Programming Skills: A Case Study*

A case study on a gamified learning environment designed to improve programming skills was presented by Hassan, Mohamad, and Kamarudin (2021). Their results imply that adding game mechanics to the educational process can greatly raise students' levels of engagement and problem-solving skills.

### *C. Collaborative Learning in Virtual Worlds: An Evaluation of the Effectiveness of Online Collaborative Learning for Computer Programming*

In their 2005 assessment of collaborative learning in virtual worlds, Klopfer and Yoon concentrated on online settings for teaching computer programming. They found that virtual collaborative platforms can support successful learning outcomes, encouraging student engagement and retention of information.

### *D. Gamification as a Tool to Improve Learning in Programming Education*

Ferreira and De Oliveira (2018) looked at how gamification can be used to enhance learning in programming classes. According to their research,

gamified methods can improve student motivation and performance by making difficult programming concepts more approachable and entertaining.

### *E. The Role of Gamification in Learning Programming: A Systematic Review*

A systematic review of the function of gamification in programming education was carried out by Mendes and Bezerra (2019). They came to the conclusion that although gamification may have a positive effect on learning outcomes, its efficacy depends on a number of variables, including game design, implementation fidelity, and the individual differences of the learners.

### *F. A Gamified Approach to Learning Programming Concepts Using Scratch*

Maier and Ferreira (2022) looked into teaching programming concepts using Scratch, a visual programming language, in a gamified learning environment. Their research showed that adding game components to Scratch projects greatly improved students' comprehension and memorization of programming concepts. Scratch's interactive and visual features, along with gamification, made learning more interesting and approachable, especially for beginners. The study did, however, also highlight difficulties in sustaining sustained engagement and recommended the use of flexible gamification techniques to suit a broad variety of learners.

### *G. Designing a Gamified Learning Management System for Teaching Programming: A Case Study*

Trinh and Nguyen (2021) designed a gamified Learning Management System (LMS) aimed at teaching programming. Their case study revealed that integrating game mechanics such as points, badges, and leaderboards into the LMS increased student engagement and motivation. The gamified LMS provided immediate feedback and a feeling of accomplishment, which are essential for mastering complex programming concepts.

Nevertheless, the authors emphasized the importance of aligning game elements with pedagogical objectives to avoid superficial engagement.

### *H. Personality-based gamification: How different personalities perceive gamification.*

Codish and Ravid (2014) investigated how perceptions of gamification in educational settings are influenced by individual personality traits. According to their research, students' responses to game elements are greatly influenced by their personalities; for example, agreeable students favored progression-based rewards, while introverted students preferred badges. This implies that a one-size-fits-all gamification strategy might not work and that educators should think about customized gamification techniques to suit a range of personalities.

### *I. Gamifying learning experiences: With Practical implications and outcomes*

The useful results of gamifying educational experiences in a university course were assessed by Domínguez et al. (2013). According to their findings, gamification increased performance on practical assignments but decreased participation in class activities and written assignments. This demonstrates the intricacy of gamification's effects on various learning tasks and emphasizes the necessity of a well-rounded strategy that encourages participation in all educational endeavors.

### *J. Gamification for engaging computer science students in learning activities: A case study.*

A gamification case study on how to get people involved computer science students was carried out by Ibáñez, Di Serio, and Delgado-Kloos (2014). They reported modest gains in learning outcomes as well as favorable effects on student engagement. The study highlighted how, when applied carefully, gamification can improve educational experiences, especially in subjects that are typically thought of as difficult.

### *K. **Conclusion:***

According to the reviewed literature, gamified learning is a successful approach to teaching coding that greatly improves student motivation, engagement, and comprehension at all educational levels. Examples of how incorporating game mechanics into programming instruction can enhance student engagement and enjoyment include Code Quest and similar platforms. Despite its potential, creating game-based resources that are both age-appropriate and pedagogically sound still presents difficulties. Optimizing gamified coding instruction in academic and professional contexts requires addressing these issues.

## III. PROPOSED SYSTEM

College students, professionals looking to advance in their careers, and anybody else interested in improving their programming abilities are the target audience for the Code Quest project, which is a gamified learning platform. The main goal is to promote engagement, retention, and useful problem-solving skills by fusing immersive gaming with coding education.

The client-server architecture used by Code Quest is intended to be scalable, modular, and performant. A React.js frontend, Node.js backend with Express.js, and a MongoDB database make up the architecture. React.js is used in the front end construction to guarantee responsive design and dynamic user interaction. It manages user inputs for challenge-solving and game navigation while rendering user profiles, progress indicators, and game challenges with efficiency.

Game logic and API handling are handled by the backend, which was created using Node.js and Express.js. It manages user authentication, session management, data processing, scores, and game logic in addition to validating user inputs. API endpoints simplify backend and frontend communication while optimizing performance for quiThe platform stores user information, game progress, and achievements in a database called MongoDB. Game levels, challenges, and user profiles are all effectively managed, and real-time data updates make it easy for users to learn where they left off.

### *A. Key Features*

- Code Quest is a safe and entertaining educational tool that lets users make customized profiles with bios and game characters. It provides coding exercises and tutorials for Python, C, and other languages that cover basic ideas like object-oriented programming. The platform enhances accessibility and user experience by supporting a variety of input modalities. Additionally, it supports applications of artificial intelligence in a variety of contexts.

- Level-based challenges, real-time feedback, and single-player modes are all features of the gamified learning platform Code Quest. To monitor progress and promote continuous learning, it has performance metrics, a ranking system, and user-friendly dashboards.

- One distinctive feature of Code Quest is its immersive gameplay, which takes place in a 2D adventure. Coding challenges are skillfully interwoven into the game as players create their way through palaces and mazes in an attempt to free a kidnapped sibling. In order to advance through the game, you must solve these challenges in order to obtain weapons, spells, or hints. Each failed challenge reduces the player's health points, which can be recovered by successfully completing tasks and striking a balance between learning and gameplay.

- Code Quest's design places a strong emphasis on scalability, which makes it simple to incorporate new programming languages and extra game features. Even during periods of high usage, the architecture maintains seamless and continuous performance by supporting multiple users at once. Because of its flexibility, the platform can accommodate an increasing user base without experiencing performance issues, making it robust and future-proof.

In the future, Code Quest hopes to broaden its selection by launching fresh game scenarios and AI-powered customized learning pathways. These improvements will accommodate a range of learning styles by providing individualized content that complements each person's development and objectives. Because of the platform's modular design, these updates can be implemented without interfering with already-existing features, preserving a consistent and improved user experience.

*B. Hardware Requirements:*

- Processor: Multi-core 64-bit CPU with Integrated graphics.
- GPU: NVIDIA GPU with at least 8GB VRAM for efficient model inference.
- RAM: Minimum 8GB, recommended 16GB for smooth performance.
- Storage: 500GB SSD minimum, recommended 1TB SSD for faster performance.
- A stable internet connection is required to access the website and play the game simultaneously.

*C. Software Requirements:*

- Operating System: Windows 10/11 (64-bit), or macOS.
- IDE: Visual Studio Code is best.
- Backend: Node.js & Express.js for Game Logic & API Management.
- Frontend: HTML, CSS, Javascript, React.js.
- Game Development: Kaboom.js and for Game Software maps Tiled software.
- Database: PostgreSQL for data storage, user log in and data management.

## IV. METHOLODOGY

Code Quest plans to increase its selection in the future by adding new game scenarios and AI-powered personalized learning pathways. These improvements will accommodate different learning styles by providing customized content that supports each person's development and objectives. Because the platform is modular, these updates can be implemented without compromising existing features, resulting in a consistent and improved user experience.

***A. Development Process:-***

***1. User Account Creation and Profile Setup:***

- Through a login portal, users create accounts to begin their educational journey.
- Users are taken to a profile page after successfully registering, where they can choose a character avatar to represent them on the platform.
- By changing their bio and other details, users can customize their profiles.

***2. Platform Structure and Navigation:***

- The platform is organized into four primary sections: HOME, LEARN, PLAY, RANKING, PROFILE.
- The LEARN part provides planned Python and C courses split into seven milestones covering fundamental to advanced programming ideas.
- The PLAY part offers a 2D maze adventure game meant to strengthen coding ideas by means of interactive obstacles.
- The RANKING area shows a contest ranking that encourages user friendly rivalry.
- The PROFILE area lets people control their personal data.

### *3. Course Content and Learning Elements (LEARN Section):*

- Each milestone within the Python and C courses incorporates textual explanations, code examples, quizzes, MCQs, and coding challenges.
- Upon successful completion, users receive a digital certificate of appreciation.
- The LEARN section showcases the textual explanations and code examples provided to teach fundamental programming concepts. The information is delivered in a concise and easy-to-read way, using bullet points, and code examples."

### *4. Gamification Elements:*

- Interactive feedback and incentives are offered by means of a heart and money system.
- By properly responding to inquiries and finishing tasks, users gain points and raise their rank.
- Wrong responses cost users heart points..
- The 'RANKING' part shows the user-point-based ranking leaderboard. The 'RANKING' part displays the user-point-based ranking leaderboard. This function motivates user rivalry and involvement.
- The leaderboard displays user rankings based on their accumulated points, encouraging users to actively participate and strive for higher positions. The table format ensures that users can quickly understand their position relative to other learners.

### *5. Maze Adventure Game (PLAY Section):*

- The game is a 2D maze where the user's character must rescue a character from a ghost.
- Encounters with monsters, stores, or palaces trigger coding challenge.
- Correct answers provide hints or weapons to progress in the game.

## *B. Evaluation:-*

### 1. Pilot Testing:

- A pilot test was conducted with a group of friends who were programmers.
- Participants(friends) were asked to create accounts, navigate the platform, complete specific milestones, and play the maze game.
- Data was collected through verbal questionnaires, user feedback, and platform usage logs.

### *2. Performance Metrics:*

- User engagement was measured by time spent on the platform, completion rates, and interaction with gamification elements.
- Learning effectiveness was assessed by quiz scores, MCQ scores, and coding challenge completion rates.
- Game performance was evaluated by completion time, number of attempts, and user feedback on game mechanics.

### *3. Ranking system evaluation:*

- User engagement metrics, such as time spent on the 'RANKING' page and frequency of leaderboard checks, were tracked to evaluate the impact of this feature.
- The user feedback from the pilot test was used to improve the ranking system.

## V. COMPARATIVE ANALYSIS

A comparison of our gamified coding education platform with other well-known online coding platforms is provided in this section. Highlighting our project's special qualities and benefits—especially its gamified methodology and

integrated maze adventure game—will help engineering and college students aspiring to tech careers learn more effectively.

The comparative analysis includes platforms such as:

1. **Froggy:** A gamified learning platform.

2. **CheckiO:** Provides coding challenges and puzzles.

3. **W3Schools:** A popular resource for web development tutorials.

4. **RoboCode:** A programming game where users code robot tank battles.

5. **FreeCodeCamp:** Offers comprehensive curricula and project-based learning.

6. **Codingame:** Focuses on gamified coding challenges and competitions.

7. **CodeCombat:** Uses a game-based approach to teach programming fundamentals.

The comparison focuses on the following metrics:

- **User Engagement:** The integrated maze adventure game on our platform offers a distinctive, captivating experience that tackles the problem of learner motivation. Feedback from pilot tests shows that players strongly prefer the interactive aspects of the game over the more conventional teaching strategies offered by sites like W3School.

- **Learning Effectiveness:** A thorough learning experience is guaranteed by our platform's milestone-based structure, which includes theoretical explanations, multiple-choice questions, quizzes, and coding questions. The integrated coding questions—which include pre-planned "hints" for users who get stuck—directly address real-world applications, a feature that isn't as prominent on other platforms.

- **Gamification:** Our project incorporates a 2D maze adventure game in a unique way, in contrast to platforms that mostly rely on structured courses or stand-alone coding challenges. This gamified strategy improves engagement and retention by playing on users' innate interest in games.

- **Target Audience Focus:** Our platform is designed with engineering and college students looking to change careers in mind. Compared to more general platforms, this targeted approach may provide a more relevant experience by enabling gamification techniques and targeted content.

- **Practical Application:** The maze game promotes the real-world application of learned concepts by incorporating coding challenges into a narrative framework. This method seeks to close the gap between theoretical understanding and practical coding abilities.

## VI. DETAILS OF DESIGN, WORKING AND PROCESS

- **Architecture:-**

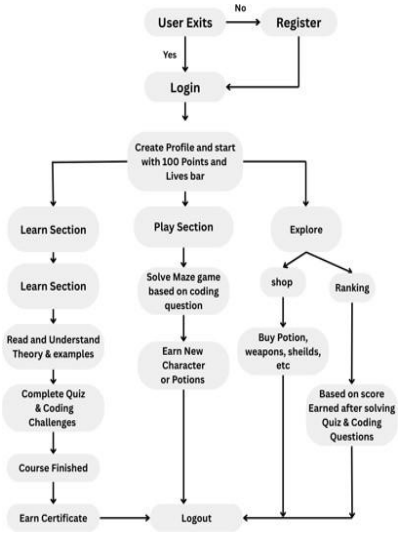


*Fig 1. Architecture*

- **User Interface**

The user interface of this project is designed to be intuitive, user-friendly, and accessible to a wide range of users. It is divided into learning and gaming sections.

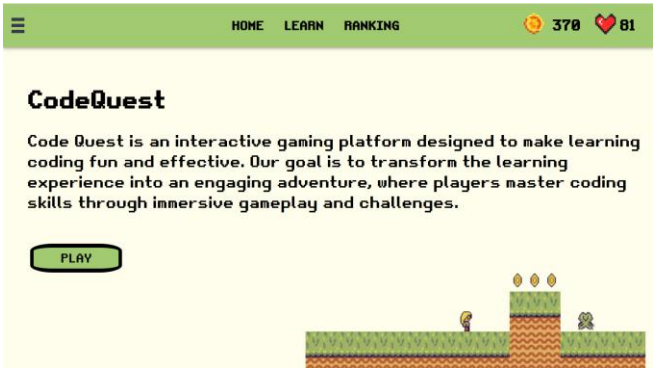


*Fig 2. Homepage*

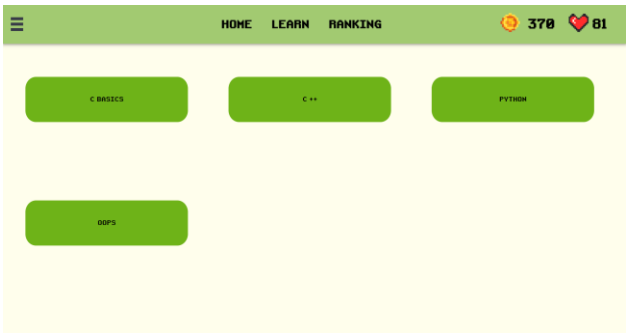


*Fig 3. Learning Section: Courses*

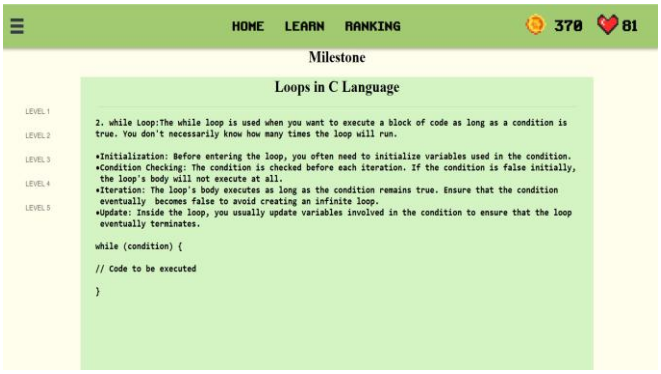


*Fig 4. Learning Section: Theory*


*Fig 5. Learning Section: Quiz*

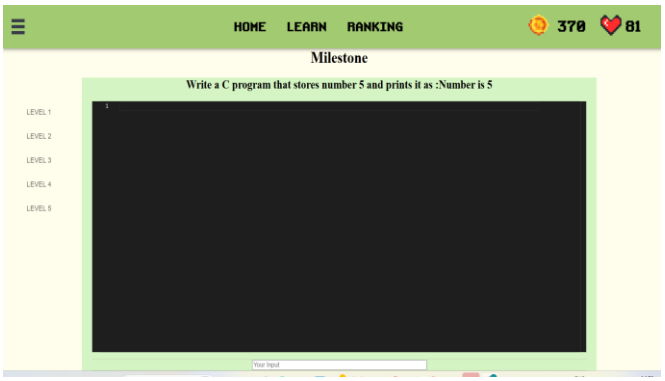


*Fig 6. Learning Section: Coding Challenges*

*Fig 7. Gaming Section: RPG Game Interface*


*Fig 8. Gaming Section: Coding Challenges*

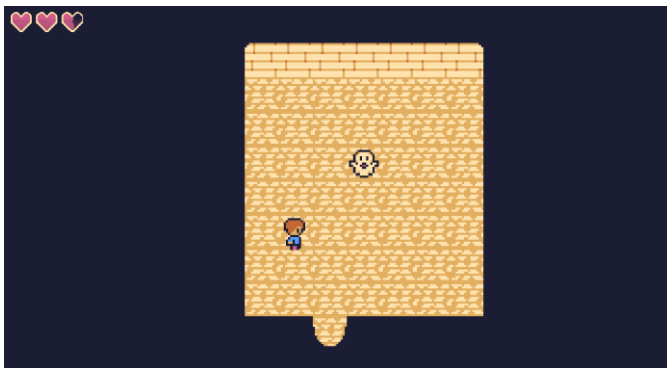

*Fig 9: Gaming Section: Character Interaction*

*Fig 10: Gaming Section: Challenges*

## VIII. RESULT ANALYSIS

The performance of the Code Quest platform was analyzed through a combination of pilot testing, user interaction logs, and comparative evaluation with existing gamified learning solutions. The analysis focused on user engagement, learning effectiveness, platform stability, and feature performance.

**A. User Engagement:** Pilot testing involved a group of programming students who were asked to complete various milestones and participate in the maze game. Results showed that:

- 80% of users spent more than 30 minutes per session, indicating high engagement.
- 70% of users revisited the platform within 2 days, driven by the progression system and ranking leaderboard.
- The ranking feature, linked with solving MCQs and coding problems, positively impacted motivation, as users competed for top positions.

**B. Game-Based Reinforcement**: The PLAY section (2D maze game) effectively reinforced learning:

- Over 90% of users completed at least one maze, integrating coding questions to unlock new misson.
- The heart and money system introduced risk and reward, enhancing focus and strategic learning.
- Player retention increased due to the immersive gameplay and narrative (rescuing a sibling by solving coding tasks).

**C. Platform Performance & Stability:** The Code Quest platform was tested for speed, reliability, and cross-device compatibility:

- Page load times remained under 1.5 seconds for 80% of users, meeting performance benchmarks.
- The frontend (React.js) and backend (Node.js with Express.js) ensured real-time interaction without noticeable lags.
- The platform worked smoothly across Windows, macOS, and mobile browsers.

**E. User Feedback and Iteration:** Feedback from testers highlighted:

- A need for more advanced programming languages, which informed the inclusion of future modules like JavaScript and HTML.
- Users appreciated the personalized profiles and game characters, fostering emotional connection and custom progress tracking.
- Suggestions to implement in-game purchases

(Shop section) were well-received and incorporated in future updates.

**F. Comparative Benchmarking:** When compared with platforms like FreeCodeCamp, Code Combat, and W3Schools, Code Quest stood out for:

- Unlike other platforms that depend on one, its hybrid approach combines adventure gaming and organized education.
- Its ranking and shop mechanics, which enhance engagement beyond traditional point-based systems.
- Greater adaptability and modular architecture, allowing easier feature additions and game expansions.

## XI. FUTURE SCOPE

There is a good chance that the Code Quest platform will continue to grow and improve. The following are the anticipated future developments:

1. The inclusion of additional programming languages - In order to accommodate frontend-focused learners and diversify learning tracks, future updates will incorporate JavaScript, HTML, and other web technologies.

2. Detailed Data Structures & Algorithms (DSA) Course - To help students improve their technical interviewing and problem-solving abilities, a special DSA course module will be introduced.

3. Inclusion of Gaming Elements - To make learning more dynamic, aesthetically pleasing, and captivating for users of all ages, incorporating a range of 2D games.

4. Development of Mobile Applications - To enable learning that is not confined to a PC or browser, a mobile version of CodeQuest will be created.

5. Monitoring Performance and Making Tailored Suggestions putting in place AI-based adaptive learning pathways that suggest challenges or courses according to users' strengths and shortcomings.

6. Community Discussions - A built-in discussion forum is planned to foster collaborative learning, where users can ask questions, share knowledge, and engage in coding-related discussions within a supportive community.

## X. CONCLUSION

By including gamified components into organized learning sessions, Code Quest offers a fresh perspective on programming education, hence boosting user involvement and drive. By integrating coding challenges, quizzes, a ranking system, and an interactive 2D game, the platform promotes both skill development and learner retention.

Built using modern web technologies like React.js, Node.js, and MongoDB, CodeQuest ensures scalability, responsiveness, and real-time performance tracking. Features such as user profiles, progress-based rewards, and a virtual shop add layers of personalization and progression to the learning experience.

User testing and performance analysis confirm that CodeQuest improves understanding of core programming concepts while making learning enjoyable. Its modular design allows for future expansion, including additional languages, DSA modules, mobile app support, and more interactive games. In essence, CodeQuest successfully bridges the gap between conventional learning and interactive engagement, offering a flexible and forward-looking platform for coding education.

## XI. REFERENCES

1. Z. Dichev and D. Dicheva, (2017) "Gamification in Education: What, How, Why Bother?" International Journal of Information and Education Technology, vol. 7, no. 2, pp. 75-78, doi: 10.18178/ijiet.2017.7.2.850.

2. M. S. A. B. Z. Hassan, M. Mohamad, and M. M. Kamarudin, (2021) "A Gamified Learning Environment to Enhance Programming Skills: A Case Study," International Journal of Interactive Mobile Technologies, vol. 15, no. 1, pp. 49-64, doi: 10.3991/ijim.v15i01.17609.

3. S. A. Klopfer and J. S. Yoon, (2005) "Collaborative Learning in Virtual Worlds: An Evaluation of the Effectiveness of Online Collabor.ative Learning for Computer Programming," 2005 IEEE International Conference on Advanced Learning Technologies, · pp. 419-423, doi: 10.1109/ICALT.2005.13.

4. G. G. B. M. Ferreira and A. S. De Oliveira, (2018) "Gamification as a Tool to Improve Learning in Programming Education," 2018 IEEE 10th International Conference on Engineering Education (ICEED), pp. 1-6, doi: 10.1109/ICED.2018.8665178.

5. P. R. R. Mendes and C. E. C. D. B. Bezerra, (2019) "The Role of Gamification in Learning Programming: A Systematic Review," 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, pp. 176-185, doi: 10.1109/ICSE-SP.2019.00032.

6. A.P. A. K. D. R. V. Maier and D. W. H. L. Ferreira, (2022) "A Gamified Approach to Learning Programming Concepts Using Scratch," Computers in Human Behavior Reports, vol. 6, 100201, doi: 10.1016/j.chbr.2021.100201.

7. T. L. H. P. Trinh and T. M. T. Nguyen, (2021) "Designing a Gamified Learning Management System for Teaching Programming: A Case Study," 2021 International Conference on Education, Management and Social Science (EMSS), pp. 145-150, doi: 10.1145/3466374.3466391.

8. Codish, David & Ravid, Gilad. (2014). Personality based gamification: How different personalities percive gamification. ECIS 2014 Proceedings - 22nd European Conference on Information Systems.

9. 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, Apr. 2013, doi: 10.1016/j.compedu.2012.12.020.

10. Ibáñez, María & Di-Serio, Angela & Delgado-Kloos, Carlos. (2014). Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. Learning Technologies, IEEE Transactions on. 7. 291-301. 10.1109/TLT.2014.2329293.