



Improving JavaScript Learning Engagement with Gamification Techniques

Wendyanto & Wirawan Istiono*

Faculty of Engineering and Informatics, Department of Informatics,
Universitas Multimedia Nusantara, Jalan Scientia Boulevard Gading, Curug Sangereng,
Serpong, Kabupaten Tangerang, Banten 15810, Indonesia

*E-mail: wirawan.istiono@umn.ac.id

Abstract. JavaScript is highly demanded in modern development, but traditional learning methods often fail to engage students. To address this, this study designed and developed a responsive, gamified web-based JavaScript learning application using the ReactJS library. The application integrates gamification elements such as dashboards, learning modules, challenges, points, levels, ranks, and achievements to enhance learning engagement. The application's usability and acceptance were evaluated using the Technology Acceptance Model (TAM) through a survey involving 35 participants. The evaluation yielded a Perceived Usefulness score of 90.00% and a Perceived Ease of Use score of 89.71%, resulting in a high overall average score of 89.85%. These results indicate that respondents strongly agree that the application is highly useful for escalating learning motivation and is easy to use. While the TAM results confirm high user acceptance and demonstrate that gamification effectively eases the JavaScript learning process, our future research will incorporate objective cognitive assessments to definitively measure its absolute pedagogical effectiveness.

Keywords: *gamification; Javascript learning; ReactJS library; technology acceptance model; web application.*

1 Introduction

The demand for skilled digital information technology workers in Indonesia reaches 600,000 per year, but only half of this number is being met. This is a crucial issue [1], because the supply of digital technology workers in Indonesia remains insufficient. According to Data Indonesia, the demand for IT professionals increased to 1.2 million in 2022 and is projected to reach 2 million by 2025 [2].

In today's technology-driven world, information technology (IT) proficiency has become an essential skill across various industries. Among the most sought-after IT skills is programming expertise. This demand stems from the pervasive role of technology in every aspect of modern life, from e-commerce and social media to mobile applications. Mastering a programming language is an IT skill that is

crucial for being able to utilize these technologies effectively [3]. JavaScript in particular stands out as a renowned and widely used programming language [4].

JavaScript is a platform-independent programming language that adheres to the ECMAScript standard for exchanging and disseminating information across various platforms [5, 6]. According to a Stack Overflow survey [7, 8], JavaScript is the most popular programming language with the largest user base. JavaScript is a crucial programming language because it is the language of web browsers [5, 9]. JavaScript's popularity stems from its ability to develop client-side applications due to the language's flexibility, support for a wide range of frameworks, and adaptability to various programming styles [10]. As a result, JavaScript is widely used in the creation of modern web applications [11, 12].

However, learning JavaScript can be a daunting task for many individuals. Traditional learning methods, such as reading textbooks or attending classes, often prove to be tedious and ineffective. A study by Chengchen, Dewaele, and Yanhong [13] concluded that boredom during foreign language learning can lead to a decline in motivation and performance [10]. Therefore, engaging and interactive learning methods are essential to enhance the learning process. Gamification serves as a prime example [14].

Gamification is a trend that focuses on the application of game mechanics to non-game contexts with the aim of engaging learners in learning activities. According to Kalogiannakis, Stamatios Papadakis, and Alkinoos-Ioannis Zourmpakis [15, 16], gamification increases engagement and promotes scientific thinking in science education. This proves the effectiveness of implementing gamification methods in the learning process [17]. In the context of learning, gamification can be applied by providing points, leaderboards, badges, and levels to increase motivation, user interaction, and social effects [18, 19].

Driven by these challenges, a gamified learning application is warranted to foster public interest in programming languages, thereby augmenting Indonesia's digital technology workforce. While several existing tools and platforms, such as CodeCombat or IntelliGame, have successfully applied gamification to programming education, they often exhibit certain limitations. Many of these platforms focus primarily on basic logic or general syntax through isolated game scenarios, lacking a seamless transition for learners into real-world, modern web development ecosystems. Furthermore, while research exists on gamifying Kotlin programming language learning [20, 21] and gamifying JavaScript learning specifically for Document Object Model (DOM) concepts, these existing solutions exhibit significant limitations. Most previous works focused primarily on introductory syntax or isolated problem-solving, lacking a continuous learning pathway that transitions students into modern, industry-standard web

development ecosystems. Furthermore, the existing literature often limits gamification to superficial points and badges without rigorously evaluating the psychological drivers of student engagement. This research gap presents a compelling opportunity for further investigation

To address this unresolved problem, the specific contribution of this paper is the holistic integration of the Octalysis gamification framework with modern web development education (ReactJS), which is then empirically validated using the Technology Acceptance Model (TAM). Unlike previous tools, this research fills the gap by providing a comprehensive, gamified curriculum that not only covers fundamental JavaScript but also directly applies it to component-based front-end development using the ReactJS library [22, 23]. By mapping specific Octalysis core drives to complex programming tasks and evaluating the platform's perceived usefulness and ease of use via TAM [24], this study goes beyond standard gamification to provide a substantiated, evidence-based approach to modern software engineering education. Later [25], the application will be evaluated using a Likert scale to get the quantitative result of user's interest and application usefulness [26].

2 Research Method

2.1 Application design process

Figure 1 shows the design process of this JavaScript programming language learning application that was comprehensively carried out in several stages, commencing with the gamification design, application model creation, system flowchart development, Entity Relationship Diagram (ERD) construction, and the interface design. Each stage will be comprehensively explained below.

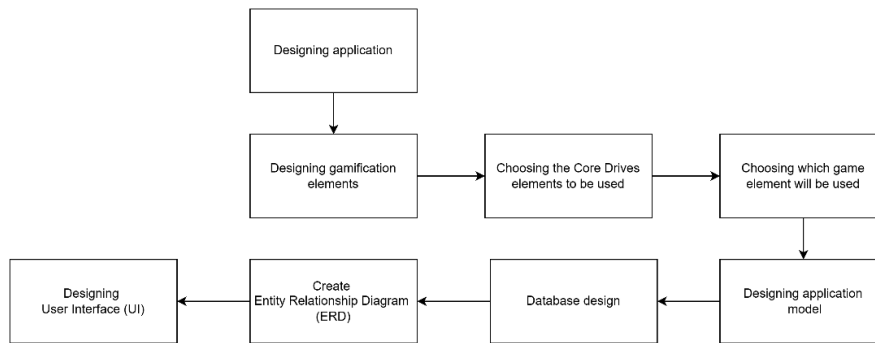


Figure 1 Flowchart of the application design process.

2.2 Gamification Design

In the design of this application, the gamification framework Octalysis was used, which was developed by a researcher named Yu-Kai Chou. From this framework, four specific drives were intentionally selected, i.e., Development & Accomplishment, Scarcity & Impatience, Unpredictability & Curiosity, and Loss & Avoidance. They directly align with the psychological needs of programming students, who often face steep learning curves and high dropout rates in self-paced learning [20, 26]. While the other four drives are also valuable, they require complex community building or narrative design, which fall outside the scope of an individual skill-acquisition tool. Therefore, this application strictly focuses on drives that foster individual progression, habit formation, and daily retention. Thus, the four elements used in this study are as follows:

1. Development & Accomplishment

Rather than simply defining this element as the desire to learn, our application uniquely maps this drive directly to the progression of JavaScript competencies. We implemented this by constructing a tiered learning path where users are rewarded with specific experience points (XP) and domain-specific badges (e.g., ‘JavaScript Wizard’) only upon successfully completing and passing ReactJS and JavaScript coding modules. This implementation ensures that players are continuously motivated to tackle increasingly difficult programming concepts to unlock the next available class.

2. Scarcity & Impatience

Instead of general scarcity, our implementation uniquely translates this drive into ‘time-gated’ programming exercises. This element focuses on the desire to feel something new and something that cannot be owned by the player at all times. We explicitly implemented this through a ‘Daily Challenge’ feature that has a strict time limit and availability window. By restricting access to these specific JavaScript challenges to a limited timeframe each day, the application builds a

sense of impatience and urgency, compelling the learner to log in daily rather than procrastinating.

3. Unpredictability & Curiosity

This element aims to make the player want to know what will happen next, so that the player's brain will continue to think about it. In our application, unpredictability is uniquely implemented through dynamic randomization of the daily coding challenges. When a user logs in, they are presented with a challenge where the difficulty level, the specific JavaScript/ReactJS topic (e.g., DOM manipulation, hooks, array methods), and the exact time limit are completely hidden until they accept the challenge. Because the challenges are randomly selected from a database, it stimulates curiosity and prevents the learning routine from feeling monotonous.

4. Loss & Avoidance

This element focuses on providing motivation for players to be able to avoid something bad, such as losing progress. We explicitly implemented this drive through a rigorous 'Streak' mechanism tied strictly to coding consistency. In this application, any player who fails to attempt or complete the daily JavaScript challenge within the 24-hour window will lose their accumulated streaks, resetting their consistency counter to zero. This implementation leverages the fear of losing a hard-earned visible record of dedication, effectively forcing users to maintain their daily coding habits.

After selecting the Core Drives elements, the next step was to select the game elements that were going to be implemented in this application. A total of eight game elements were implemented, including:

1. Level

This element is a basic game element. The level depends on the amount of experience (XP) that the user has collected. The higher the user's level, the higher the amount of experience required to move to the next level. Currently, the highest level that can be reached by a player is level 50. The amount of experience required for each level will increase by 200.

2. Points

Points are a game element that is also implemented in this application. The way a user can collect points is by completing a learning module, getting achievements, or completing a daily challenge. Later, points will be useful for use in the next game element, namely the leaderboard.

3. Leaderboard

This game element is an element that shows the ranking of all users based on the number of points that each user has. Later, on the leaderboard, all users will be displayed in a table format along with the accumulation of the number of points.

4. Achievements

Achievements are a game element in the form of an award when the user successfully achieves something. Each achievement that exists has its own requirements to be obtained by the user. In Table 1, it can be seen that each achievement has a prize in the form of additional points and experience.

Table 1 Achievements table.

Name	Requirements	Points	Experiences
Hello, World	Finished the 'Javascript Introduction' course	120 points	200 XP
JavaScript Wizard	Finished the 'Web Development with ReactJS' course	415 points	1000 XP
Code Connoisseur	Finished the 'Best Practices in JavaScript' course	275 points	300 XP
Challenge Accepted	Completed the daily challenge for the first time	75 points	100 XP
Daily Challenger	Has total of 5 streaks	300 points	500 XP
Challenge the Titan	Has total of 15 streaks	1000 points	2000 XP
Challenge Enthusiast	Completed 10 daily challenges	750 points	1500 XP

5. Progress tracking

Progress tracking is one of the game elements that is useful for seeing the user's progress in working on certain activities. In the JStudy application, this element is used to see the progress of user learning for a learning module that is being followed by displaying the percentage of module completion.

6. Rank

This game element is an element to assess the user's ability. The higher the user's rank, the more the user is considered to have good learning performance and broader knowledge of the learning material that has been studied. The user's rank is influenced by the level they have at the time. The higher the level, the higher the rank that the user achieves.

7. Challenge

Challenge is one of the game elements that is useful for training and testing the user's ability against the learning material they have studied. Challenges in this application can only be accessed once a day and affects the game element called Streaks. Each challenge provided is tailored to the material in the learning application.

8. Streak

Streak is a game element that counts the number of times the user has successfully completed challenges in a row. The number of streaks is reset to 0 when the user fails to complete a challenge or does not complete a challenge within a period of 24 hours.

2.3 Application Model

In this JavaScript learning application, there are scenarios where users perform activities within the application. Upon completion of these actions, the application will immediately send a request to the 'Supabase' database server. Once the request is received by the server, the retrieved data is sent back to the application in the form of a response. This response will then be displayed on the application, allowing users to view information such as their level, rank, points, achievements, and class.

2.4 Flowcharts

The purpose of the flowchart design is to illustrate the workflow of the JStudy application, a JavaScript learning platform that utilizes gamification.

Upon entering the Course Module page, data from the module is first retrieved from the database and then displayed. This page features navigation to the previous and next modules. If there is a previous module, data from that module is retrieved; similarly, this occurs when there is a next module. Once the user reaches the last module, the button changes to 'Finish.' Clicking this button updates the student courses data and returns the user to the Courses page, as shown in Figure 2.

The flowchart for the Daily Challenge page is shown in Figure 3. This page is where users must answer a randomly generated challenge question with a random difficulty level and topic. When the user enters this page, the challenge data is first retrieved, and then a confirmation modal is displayed showing the data from the challenge, such as the duration, difficulty level, and topic of the question. When the user chooses to accept the challenge, the question is displayed and the timer starts. When the user enters their answer, the answer is checked first. If it is incorrect, the user must continue entering answers until they are correct or the timer runs out. When the answer is correct, the timer stops and a congratulatory message is displayed. If the timer runs out and the user is still unable to answer correctly, a message is displayed stating that the user has failed to complete the challenge.

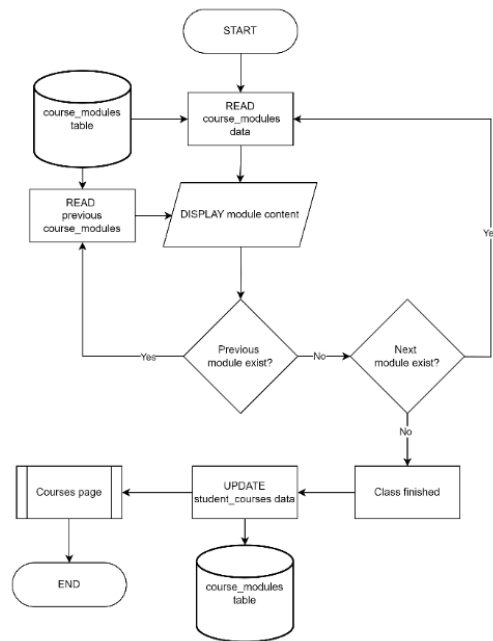


Figure 2 Course module flowchart.

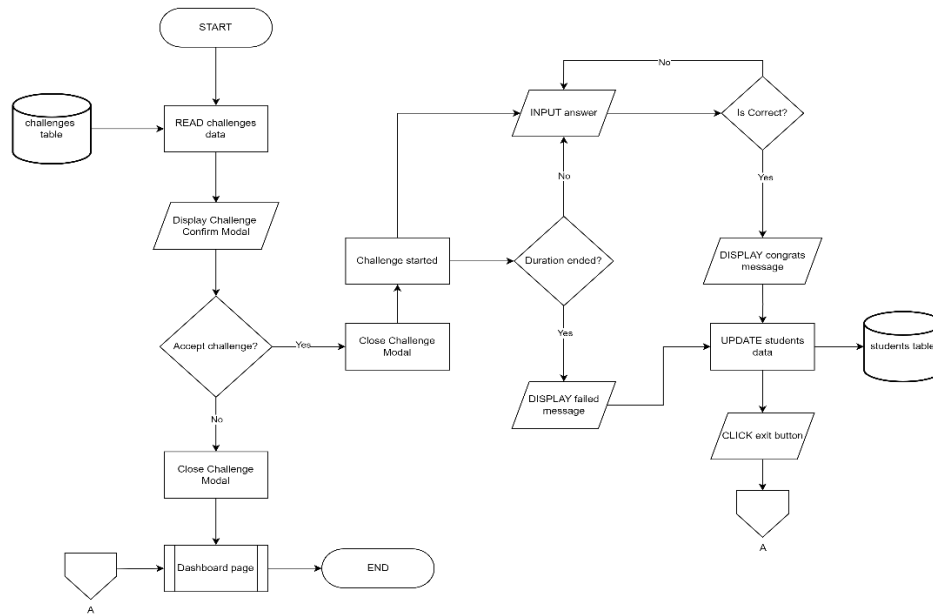


Figure 3 Daily challenge flowchart.

The flowchart for the gamification achievement elements is presented in Figure 4. The application checks the user's data and compares it with the conditions specified for each achievement. If the user's data meets the criteria for an achievement, a message will be displayed indicating that the user has successfully earned a new achievement and the user's data in the database will be updated.

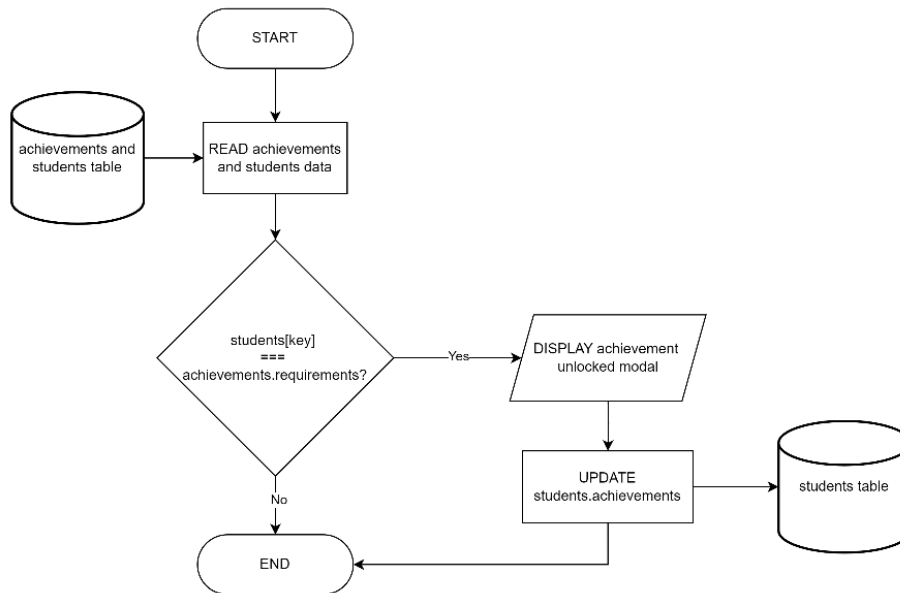


Figure 4 Achievement system flowchart.

The flowchart for the gamification rank elements is depicted in Figure 5. This flowchart illustrates the mechanism by which a user's rank can be elevated, for instance, from 'Beginner' to 'Advanced'. Each rank has specific criteria that must be met to be attained, which are based on the user's level. Once a user's level meets the requirements for entering a new rank, the user's rank is updated in the database.

The flowchart for the gamification streaks elements is presented in Figure 6. This flowchart explains how the number of streaks a user has can increase and be reset to 0. This depends on whether the user participates in the daily challenges given each day. If the user participates and successfully completes the given challenge, the number of streaks will increase by 1. If the user participates but fails to complete the given challenge, the number of streaks they have becomes 0. The number of streaks can also become 0 if the user does not complete the challenge within 24 hours.

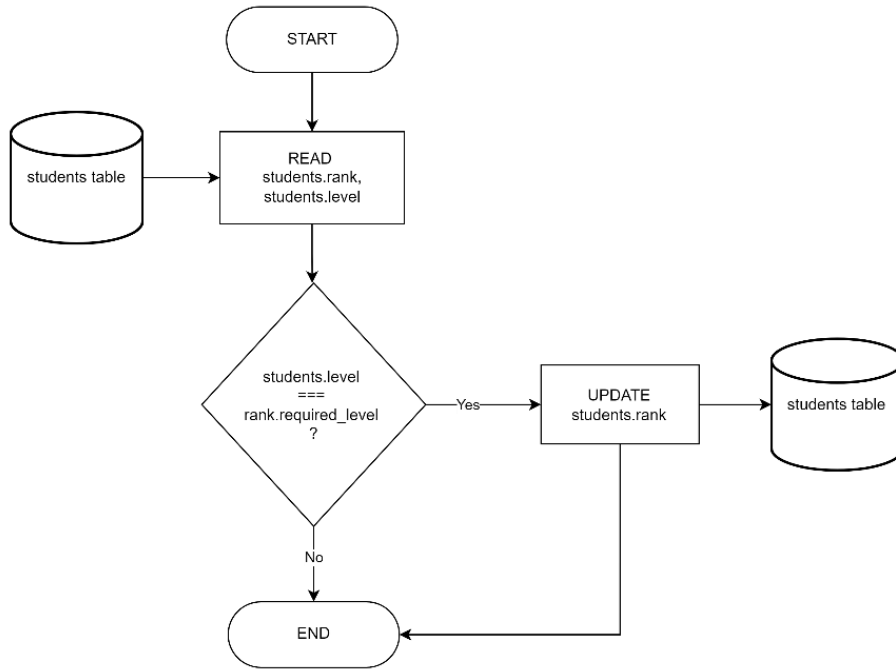


Figure 5 Rank system flowchart.

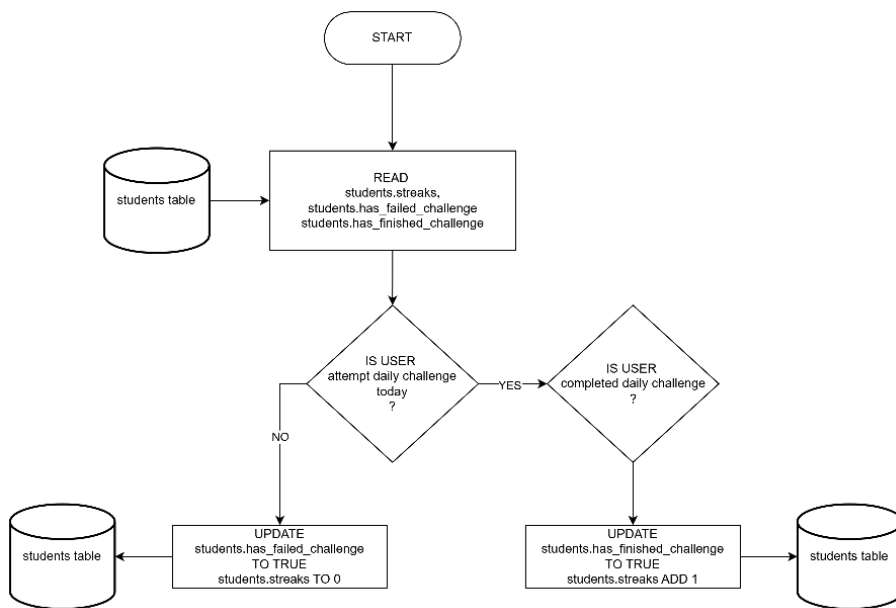


Figure 6 Streak system flowchart.

2.5 Entity Relationship Diagram (ERD)

Figure 7 presents the Entity Relationship Diagram (ERD) design of the JStudy application. It serves to provide an overview of the database schema and the relationships between each table in the JStudy application's database. The designed ERD consists of a total of four tables after undergoing two normalizations (2NF), namely students, student courses, courses, and course modules, which have their respective relationships.

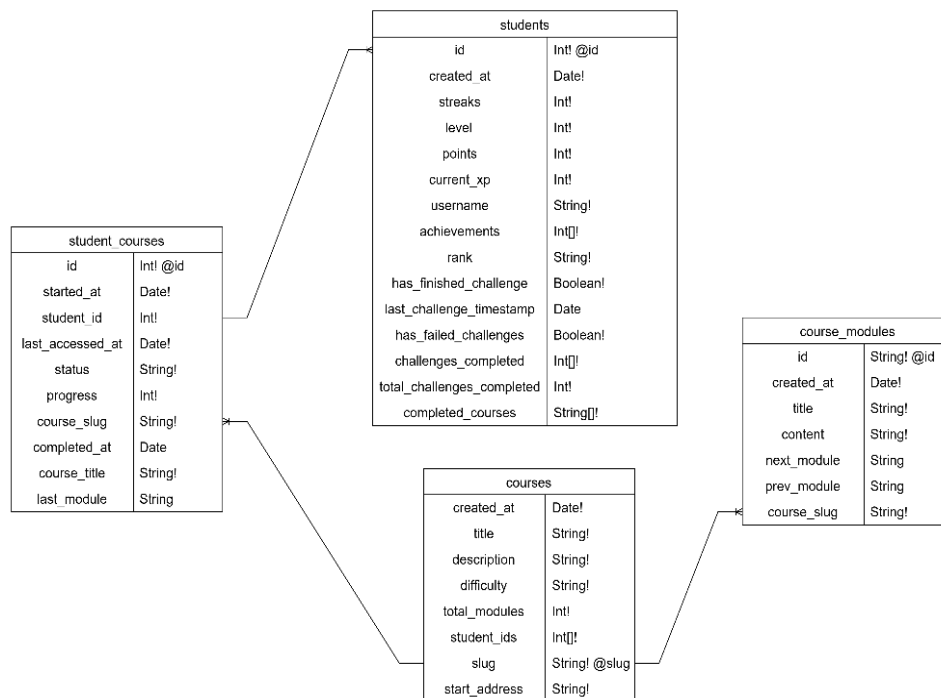


Figure 7 Entity Relationship Diagram (ERD).

2.6 User Interface (UI) design

The user interface design shown in Figure 8 was created to provide a visual representation of the JStudy application that was developed in this research. The interface design was created using a software called Figma, which is specifically designed for creating website and mobile interface designs.



Figure 8 Interface design for the JStudy Application.

3 Results And Discussion

3.1 Application Testing

Application testing was conducted using the black box testing method, where researchers try various test cases that have been created to ensure that the functions in the application run as expected [24]. In addition, the researchers also asked users to try and provide feedback on the learning application. After the testing was completed by the users, the users were asked to fill out a questionnaire that was created using Google Forms through a link provided by the researchers. To measure users' Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) of the JStudy application, statements were made based on the Technology Acceptance Model (TAM) model. Based on the tests conducted on 35 users, all users stated that they were willing to fill out the questionnaire provided.

The answers were divided into five categories, and the results of the questionnaire can be seen in Table 2 (VA = Very Agree, A = Agree, D = Doubt, DA = Disagree, VDA = Very Disagree). Table 3 presents the results based on the respondents' answers to each statement in the questionnaire. Then, to obtain more detailed results, further calculations were performed on each category of statements to obtain the average percentage.

Table 2 Questionnaire results.

Statement ID	VA	A	D	DA	VDA
<i>Perceived Usefulness</i>					
PU1	21	12	2	0	0
PU2	20	12	3	0	0
PU3	17	16	2	0	0
PU4	21	12	2	0	0
<i>Perceived Ease of Use</i>					
PEOU1	20	12	2	0	0
PEOU2	21	11	3	0	0
PEOU3	17	17	1	0	0
<i>Attitude Toward Using</i>					
ATU1	22	8	4	0	0
ATU2	23	11	1	0	0
ATU3	17	16	2	0	0
<i>Behavioral Intention to Use</i>					
BITU1	17	13	4	1	0
BITU2	19	13	2	1	0
BITU3	18	12	5	0	0
<i>Actual Use</i>					
AU1	22	12	1	0	0
AU2	23	12	0	0	0
AU3	22	12	1	0	0

Table 3 Percentage of perceived usefulness.

ID	Percentage	Result
<i>Percentage of Perceived Usefulness</i>		
PU1	90.86%	Very Agree
PU2	89.71%	Very Agree
PU3	88.57 %	Very Agree
PU4	90.86%	Very Agree
Final Result	90.00 %	Very Agree
<i>Percentage of Perceived Ease of Use</i>		
PEOU1	89.71%	Very Agree
PEOU2	90.29%	Very Agree
PEOU3	89.14%	Very Agree
Final Result	89.71%	Very Agree
<i>Percentage of Attitude Toward Using</i>		
ATU1	88.00%	Very Agree
ATU2	92.57 %	Very Agree
ATU3	88.57%	Very Agree
Final Result	89.71%	Very Agree
<i>Behavioral Intention to Use</i>		
BITU1	86.29%	Very Agree
BITU2	88.57%	Very Agree
BITU3	87.43%	Very Agree
Final Result	87.43%	Very Agree
<i>Percentage of Actual Use</i>		
AU1	92.00%	Very Agree
AU2	93.14%	Very Agree
AU3	92.00%	Very Agree
Final Result	92.38%	Very Agree

3.2 Evaluation

Based on the application testing and questionnaires collected from 35 respondents, the results yielded the following results: 90.00% for Perceived Usefulness, 89.71% for Perceived Ease of Use, 89.71% for Attitude Toward Using, 87.43% for Behavioral Intention to Use, and 92.38% for Actual Use. Consequently, the overall average score for the Technology Acceptance Model was 89.85%. We acknowledge that relying solely on TAM and subjective Likert scale perception without a formal experimental setup (such as a pre-test/post-test design or control groups) is a limitation in measuring absolute pedagogical effectiveness. In this initial study, the primary focus was on system development, user acceptance, and establishing the platform's motivational appeal through Octalysis gamification. As highlighted in recent educational frameworks, assessing the true cognitive impact requires objective metrics. Applying tools like the NASA-TLX questionnaire to measure cognitive load, tracking time-to-complete per module, or conducting comparative experimental studies against traditional learning methods would yield deeper insights. Therefore, measuring the definitive cognitive impact and practical learning efficiency of this application using a controlled experimental design is planned as the primary focus for our future work.

4 Conclusion

This research successfully designed and developed the JStudy JavaScript learning application, offering a specific contribution by holistically integrating the Octalysis gamification framework with modern web development education (ReactJS) and empirically validating it using the Technology Acceptance Model (TAM). The elements selected from Octalysis were Development & Accomplishment, Scarcity & Impatience, Unpredictability & Curiosity, and Loss & Avoidance. To enhance the learning experience, the application incorporates various gamification elements, such as levels, ranks, challenges, points, progress tracking, streaks, leaderboards, and achievements. In the application testing phase, 35 respondents evaluated the application. The study measured two key aspects: Perceived Usefulness and Perceived Ease of Use. The Technology Acceptance Model (TAM) and a Likert scale were employed for this evaluation. The results indicated high levels of Perceived Usefulness (90.00%) and Perceived Ease of Use (89.71%). These findings suggest that gamified JavaScript learning applications can be both beneficial and user-friendly. However, we acknowledge that this initial study focused primarily on system development and motivational appeal. Relying solely on subjective perception without a formal experimental setup is a limitation of this study. Therefore, measuring the definitive cognitive impact, cognitive load (e.g., using NASA-TLX), and practical learning efficiency

of this application through controlled pre-test/post-test designs will be a primary focus for our future work.

Furthermore, to elevate the project beyond static, pre-authored content into an adaptive, intelligent learning environment, future iterations of this platform will explore the integration of AI copilots. Large Language Models (LLMs) have demonstrated significant potential in software education by acting as personalized tutors within the coding environment. Integrating technologies akin to OpenAI Codex [27] or cloud-based LLMs for coding assistance [28] could provide students with real-time, contextual help. Such an intelligent assistant could offer syntax hints, explain error messages in plain language, suggest alternative implementations, and answer conceptual questions, thereby transforming the application into an interactive, guided practice space. Second, to ensure the copilot's guidance is accurate and aligned with the course's curriculum, it will be augmented with Retrieval-Augmented Generation (RAG). Instead of relying solely on the LLM's parametric memory, a RAG system retrieves relevant information from curated course materials, official JavaScript documentation, and best practice guides [29]. Combining this with aligned human feedback for reinforcement learning on code generation [30] will ground the AI's suggestions in verified, authoritative sources. This approach significantly reduces the risk of AI hallucination and ensures that the guidance strictly reinforces the specific learning objectives of the platform. By establishing this gamified foundation, this project serves as a vital stepping stone toward the next generation of AI-augmented, knowledge-grounded programming education tools.

Acknowledgements

The authors would like to express their gratitude to Universitas Multimedia Nusantara for funding and supporting this research project through the Internal Research Grant Scheme. The authors also thank the Digital Interaction Laboratory, Faculty of Engineering and Informatics, Informatics Department for providing the necessary research facilities and infrastructure that made this work possible.

References

- [1] Gayatri, G., Jaya, I G.N.M. & Rumata, V.M., *The Indonesian Digital Workforce Gaps in 2021-2025*, *Sustain.*, **15**(1), 754, 2023. doi:10.3390/su15010754
- [2] Rinarta, K. & Kartika, L.G.S., *Distribution of Skill Requirements for Information Technology Graduates in Indonesia*, *Samā Jiva Jnānam (International J Soc Stud.)*, **1**(2), pp. 38-58, 2024. doi:10.25078/ijoss.v1i2.3311

- [3] de Ruiter, L.E. & Bers, M.U., *The Coding Stages Assessment: Development and Validation of an Instrument for Assessing Young Children's Proficiency in the Scratchjr Programming Language*, Comput Sci Educ., **32**(4), pp. 388-417, 2022. doi:10.1080/08993408.2021.1956216
- [4] Jiang Z, Zhong H, Meng N. *Investigating And Recommending Co-Changed Entities for JavaScript Programs*, J Syst Softw., **180**(1), pp.1-40, 2021. doi:10.1016/j.jss.2021.111027
- [5] Đurđev, D., *Popularity of Programming Language*. Aidasco Rev., **2**(2), pp. 24-29, 2024. doi:https://doi.org/10.59783/aire.2024.59
- [6] Misu MRH, Satter A. An Exploratory Study of Analyzing JavaScript Online Code Clones. IEEE Int Conf Progr Compr. 2022;2022-March:94–8. doi:10.1145/3524610.3528390
- [7] Moutidis, I., Williams, H.T.P., *Community Evolution on Stack Overflow*, PLoS One, **16**(6 June 2021), pp. 1-23, 2021. doi:10.1371/journal.pone.0253010 PubMed PMID: 34138921.
- [8] Khan Mohd, T., Thompson, J., Carmine, A. & Reuter, G., *Comparative Analysis on Various CSS and JavaScript Frameworks*, J Softw., **17**(6):282-291, 2022. doi:10.17706/jsw.17.6.282-291
- [9] Ferreira, F., Borges, H.S. & Valente, M.T., *On the (Un-)Adoption of Javascript Front-End Frameworks*, Softw - Pract Exp., **52**(4), pp.947-966, 2022. doi:10.1002/spe.3044
- [10] Fariz, M., Lazuardy, S. & Anggraini, D., *Modern Front End Web Architectures with React.Js and Next.Js.*, Int Res J Adv Eng Sci., **7**(1), pp.132-141, 2022.
- [11] Vyas, R., *Comparative Analysis on Front-End Frameworks for Web Applications*, Int J Res Appl Sci Eng Technol., **10**(7), pp. 298-307, 2022. doi:10.22214/ijraset.2022.45260
- [12] Goh, H.A., Ho, C.K. & Abas, F.S., *Front-End Deep Learning Web Apps Development and Deployment: A Review*, Appl Intell., **53**(12), pp.15923-15945, 2023. doi:10.1007/s10489-022-04278-6
- [13] Li, C., Dewaele, J.M. & Hu, Y. *Foreign Language Learning Boredom: Conceptualization and Measurement*, Appl Linguist Rev., **14**(2), pp.223-249, 2023. doi:10.1515/applirev-2020-0124
- [14] Jawad, H.M. & Tout, S., *Gamifying Computer Science Education for Z Generation*, Inf., **12**(11), pp.1-18, 2021. doi:10.3390/info12110453
- [15] Kalogiannakis, M., Papadakis, S. & Zourmpakis, A.I., *Gamification in Science Education, A Systematic Review of the Literature*, Educ Sci. 2021, **11**(1), pp.1-36. doi:10.3390/educsci11010022
- [16] Istiono, W., *Mandarin Language Learning with Gamification Method*, Int J Adv Trends Comput Sci Eng., **10**(5), pp. 3046-3052, 2021. doi:10.30534/ijatcse/2021/171052021

- [17] Oswald, P.A. & Gunawan, D., *Avia Saga : A Gamified Mobile-Based Learning Management System*, Int J New Media Technol., **11**(1), pp. 35-42, 2024. doi: 10.31937/ijnmt.v11i1.3675
- [18] Jonathan, L. & Istiono, W., *Impact of Learning Computer Science in the Side Scrolling 2D Forms*, J Appl Comput Sci Math., **17**(1), pp. 9-14, 2023. doi:10.4316/jacsm.202301001
- [19] Johan, M.E., Sanjaya, S.A., Brata, A.Y., Brilliyant, A., *Gamification to Study Machine Learning using Octalysis Framework*, Ultim J Tek Inform., **15**(1), pp. 35-43, 2023. doi:10.31937/ti.v15i1.3118
- [20] Dimitrijević, N., Zdravković, N. & Milićević, V., *An Automated Grading Framework for the Mobile Development Programming Language Kotlin*. Int J Qual Res., **17**(2), pp. 313-324, 2023. doi:10.24874/IJQR17.02-01
- [21] Araújo, T.A., Campos, J., Ferreira, M.C. & Fernandes, C.S., *Improving Accessibility with Gamification Strategies: Development of a Prototype App.*, Int Conf Inf Commun Technol Ageing Well e-Health, ICT4AWE - Proc., (Ict4awe), pp. 237-242, 2024. doi:10.5220/0012702700003699
- [22] Haider, A., *Javascript frontend web app frameworks: React vs Svelte*, DiVA-portal, **10**, 2022. Available from: <https://www.diva-portal.org/smash/get/diva2:1681585/FULLTEXT01.pdf>
- [23] Skrzypiec, S. & Plechawska-Wójcik, M., *Comparative analysis of Angular and React frameworks Analiza porównawcza szkieletów programistycznych Angular i React*, J Comput Sci Inst., **28**(June), pp. 256-263, 2023.
- [24] Masrom, M., *Technology Acceptance Model and E-learning*, 12th Int Conf Educ., **May**, pp. 21-24, 2007.
- [25] Weber, P., Grönwald L. & Ludwig, T., *Reflection on the Octalysis Framework as a Design and Evaluation Tool*, CEUR Workshop Proc., **3147**, pp.75-84, 2022.
- [26] Jebb, A.T., Ng ,V. & Tay, L., *A Review of Key Likert Scale Development Advances: 1995–2019*, Front Psychol., **12**(May), pp. 1-14, 2021.
- [27] Chen, M., Tworek, J., Jun, H., Yuan, Q., Ponde, H., Pinto, D.O., et al. *Evaluating Large Language Models Trained on Code*. arXiv (Cornell Univ.)2021. doi: 10.48550/arXiv.2107.03374
- [28] Tan, C.W., Guo, S., Wong, M.F. & Hang, C.N., *Copilot for Xcode: Exploring AI-Assisted Programming by Prompting Cloud-based Large Language Models*. arXiv. 2023. doi:10.48550/arXiv.2307.14349
- [29] Tan, C.W., *From Code Generation to Software Testing: AI Copilot With Context-Based Retrieval-Augmented Generation*, IEEE Softw., **42**(4), pp.34-42, 2025. doi:10.1109/MS.2025.3549628
- [30] Wong, M.F. & Tan, C.W., *Aligning Crowd-sourced Human Feedback for Reinforcement Learning on Code Generation by Large Language Models*, IEEE Trans Big Data, pp. 1-12, 2024. doi:10.1109/TBDATA.2024.3523581