ENHANCING MIDDLE SCHOOL STUDENTS COMPUTATIONAL THINKING THROUGH SCRATCH-BASED ANIMATION IN INTRODUCTORY PROGRAMMING LEARNING CASE STUDY: AL HARAKI ISLAMIC SCHOOL DEPOK

Zaqi KURNIAWAN[1*], Rizka TIAHARYADINI[2], Windhy WIDHYANTY[3]
[1,2,3] ***Universitas Budi Luhur, Jakarta, INDONESIA***
[*]***zaqi.kurniawan@budiluhur.ac.id***

ABSTRACT

This community service project investigates how middle school students at Al Haraki Islamic School in Depok might improve their computational thinking (CT) skills by incorporating Scratch-based animation into introductory programming. The community service program was carried out in phases, which included planning, conducting workshops, assessing participants, and concluding the program. Through interactive animation projects, students learned about pattern recognition, abstraction, algorithmic design, and problem decomposition. Data from the pre- and post-tests demonstrate a notable improvement in CT skills: algorithmic design rose from 58.4 to 82.1, issue decomposition from 61.2 to 79.5, pattern identification from 64.8 to 81.0, and abstraction from 59.7 to 77.3. During workshop sessions, the progressive learning process was reinforced. The average skill level increased from 2.0 at the beginning to 4.86 by the end, with abstraction demonstrating the most gain (+3.3). Surveys of feedback also showed significant motivation (mean = 4.5) and engagement (mean = 4.3) on a 5-point scale, demonstrating that Scratch-based animation promoted creativity and group learning in addition to enhancing cognitive abilities. Student confidence and the necessity of consistent teacher support to guarantee program sustainability were brought to light during reflection sessions. Overall, this strategy worked well to connect middle school students' abstract CT concepts with interesting, actual-life situations.

Keywords: Animation, Computational Thinking, Middle School, Programming Education, Scratch

1. INTRODUCTION

In the current digital age, computational thinking (ICT) has become a vital ability that is frequently cited as being just important for the twenty-first century as literacy and numeracy (Fanchamps & Vansteenkiste, 2023). According to recent research, including CT into middle school instruction early on improves kid's ability to solve problems and develops their creativity and logical thinking, two skills that are essential for both academic performance and future employment (Çelik & Sevgi, 2025). The barrier to entry for young learners has been greatly reduced by visual programming tools like Scratch, which enable kids to learn to code trough interactive stories, games, and animations while also fostering algorithmic thinking (Sidiq et al., 2025). Using Scratch-based animation in introductory programming is a contemporary and effective way to foster computational thinking in middle school students, as digital literacy and STEM preparedness become more and more important in global education (Sawada et al., 2025).

Through the integration of Scratch-based animation into beginning programming instruction, this community activity aims to improve middle school students computational thinking abilities. According to earlier research, visual programming environments such as Scratch reduce entrance barriers and encourage students algorithmic reasoning, creativity, and problem-solving skills (Fuentes-Peñailillo et al., 2024; Papadakis et al., 2017). Despite the fact that Scratch has been widely used in computer science education, not much known about how animation-driven projects specifically help in the development of computational thinking skills like pattern recognition, abstraction, and decomposition in middle school settings (Holstein & Cohen, 2025; Stewart & Baek, 2023). Student's ability to think computationally, their programming abilities, and their involvement with visual storytelling are important factors in this community services project. The study's thematic framework includes the development of computer literacy, creative learning via technology, and programming pedagogy. This community service project takes place in a middle school at Al Haraki Islamic School, a place where formal curricula are just beginning to incorporate computational thinking and where engaging and culturally relevant methods, like Scratch-based animation, are required to close the gap between student's real-world experiences and basic programming concepts.

Numerous studies have pointed out problems in the way computational thinking (CT) is presented in middle schools, despite the fact that it is becoming increasingly acknowledged as a crucial 21st-century skill. Previous studies show that although Scratch-based programming has been widely used to encourage creativity and problem-solving skills, students frequently find it difficult to get past simple coding assignments in order to develop deep CT competencies (Putra et al., 2025; Romano et al., 2019). Furthermore, rather than specifically scaffolding CT through organized learning methods, the majority of current research place more emphasis on project completion or coding syntax (Wei et al., 2021). As a result, there is a theoretical void in our knowledge of how animation-based methods might successfully combine enjoyable learning with focused CT skill development. From a practical perspective, teachers usually encounter difficulties incorporating CT into current curricula since there are insufficient training resources and pedagogical framework (Huizinga et al., 2019; Nguyen & Le, 2024). Early CT competency deficiencies may exacerbate digital literacy inequalities in society and hinder student readiness for the demands of future education and employment (Giménez et al., 2021). By addressing these issues and integrating Scratch-based animation into introductory programming, we can give teachers and students easily accessible, interesting, and educationally sound ways to enhance CT development, which will help create a more sustainable and equitable computational education systems (Perez et al., 2023; Su & Yang, 2023).

In spite of a growing variety of studies showing how Scratch may be used to help middle school kids develop computational thinking (CT), there are still a number of gaps and problems that need to be addressed. Although previous community service has shown that Scratch-based learning can enhance mathematical reasoning and academic performance, many studies concentrate more on content knowledge than on the comprehensive development of CT aspects including abstraction, deconstruction, and algorithmic design (Lei et al., 2020; Rodríguez-Martínez et al., 2020). Lack of research on how Scratch-based animation projects, as opposed to conventional coding exercise, specifically affect motivation and involvement in programming is further limitation (Sidiq et al., 2025). The instructional techniques required to support student's reflection on their computational practices have also been largely overlooked in research, leaving open the question of whether visual programming environments by themselves are adequate to foster higher-order CT skills. This community services aims to fill these gaps by examining how computational thinking in middle school student's introductory programming classes might be improved using Scratch-based animation activities, with focus on both affective engagement and cognitive skill development. The following inquiries thus serve as a guide for the community services: (1) how much do middle school students acquire important aspects of computational thinking through Scratch-based animation projects? (2) What effects do these initiatives have on student's engagement and motivation in beginning programming contexts?

There is still not much understanding about how programming tools like Scratch can successfully connect abstract computational principles with interesting, educational experiences, despite the increased emphasis on incorporating computational thinking into middle school curriculum. Prior research frequently emphasizes the possibility of visual programming environments, but often overlooks the ways in which innovative methods, such as animation-based projects. The gap highlights the need for a more thorough investigation of the ways in which Scratch-based animation might be strategically used in beginning classes to improve students theoretical understanding and practical problem-solving skills. Therefore, the purpose of this community service is to find out how much animation-driven programming activities help middle school students develop computational thinking abilities while also identifying the pedagogical advantages for teachers in creating exciting and significant learning pathways.

2. METHOD

The community service program was established as a cooperative educational project to improve Al Haraki Islamic middle schools student's computational thinking abilities by using animation exercises based on Scratch. Preparation stage, workshop delivery, evaluation stage, and program complete were the steps in this community services. The flow of activities can be seen in Figure 1 below, and the details of the activities are presented in the following sub-sections.

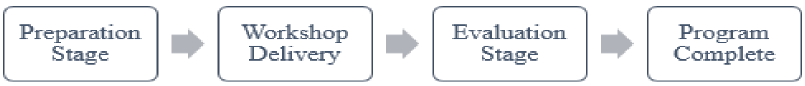

Figure 1. Stages of Community Service Activities

PREPARATION STAGE

The planning stage was created to guarantee that the implementation of workshops for Scratch-based animation would be carried out in an organized and efficient way. Participant identification, instructional material development, and technological infrastructure preparation were the main goals of this stage. First, middle school students expressed in beginning programming activities were chosen to attend Al Haraki Islamic School. The program's goals were synchronized with the curriculum through coordination with instructors and school officials. Second, the Scratch programming environment was connected with educational modules that introduced computational thinking ideas and included deconstruction pattern recognition, abstraction, and algorithmic design. The modules relatable storytelling and gamified learning components were designed with middle school students in mind. Third, technical preparation involved providing computer laboratories outfitted with appropriate hardware and software, maintaining stable internet connectivity, and installing the latest version of Scratch. The facilitators also attended briefing sessions to ensure that their teaching methods and assessment standards were in line. In order to reduce any possible interruptions during the workshop delivery stage, this precise planning was essential. Key activities in the Preparation Stage can see in Table 1, and Scratch application interface can see in Figure 2.

Table 1. Activities in the Preparation Phase

Session	***Description***	***Responsible Party***	***Expected Outcome***
Participant selection	Identifying Al Haraki middle school students through teacher recommendations and interest surveys	School administrators, and teachers	List of motivated student participants
Curriculum & Module Design	Development of Scratch-based animation modules emphasizing computational thinking concepts	Research team, and facilitators	Structured teaching materials and lesson plan
Technical Infrastructure	Preparing computer labs, software installation, internet connectivity, and projector setup	IT staff and facilitators	Fully functional technical environment
Facilitator Training	Conducting briefing sessions to align instructional approach and assessment methods	Research team	Consistent delivery of workshop objectives

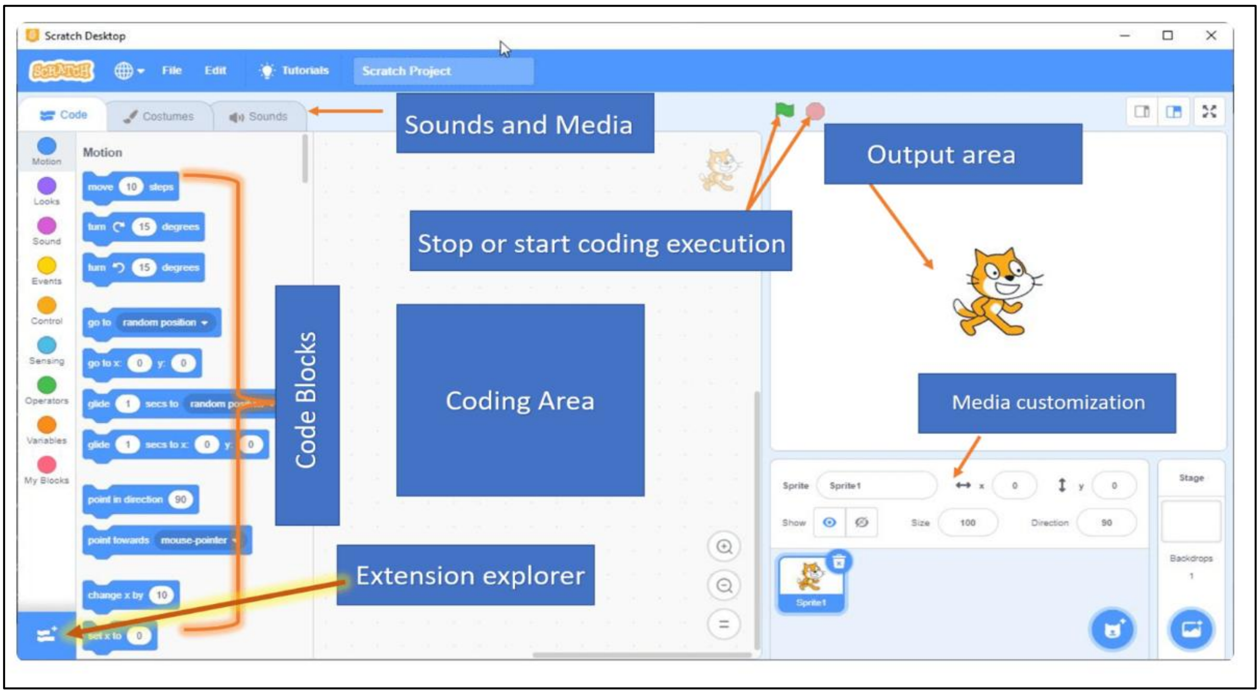

Figure 2. Scratch Desktop Interface

WORKSHOP DELIVERY

The goal of the workshop delivery phase was to use Scratch-based animation to give Al Haraki Islamic middle school students an interesting and dynamic learning experience focused on computational thinking. The workshop was split into several organized sessions, each of which was designed to strike a balance between theoretical ideas and practical application. Students learned about issue decomposition, pattern identification, abstraction, algorithmic design, and basic programming logic through imaginative animation exercises. To enable students to work together and reflect on their learning, each session began with a brief lecture and demonstration. This was followed by group-based exercises and guided practice. The workshop gradually increased the projects' complexity while still taking into account different degrees of prior knowledge in order to guarantee ongoing engagement. With teachers offering scaffolding, feedback, and real-time troubleshooting, the facilitation placed a strong emphasis on active engagement. Structure of workshop delivery can see in Table 2 and workshop documentation can see in Figure 3.

Table 2. Workshop Implementation

Session	*Focus Area*	*Activities*	*Duration*
1	Introduction to Scratch & Computational Thinking	Overview of Scratch interface, basic commands, and simple animation task	30 min
2	Problem Decomposition	Breaking down a story into smaller animated scenes	45 min
3	Pattern Recognition & Loops	Using loops and repetition for character movement	60 min
4	Abstraction & Modularization	Creating reusable code blocks for repeated animations	90 min
5	Algorithm Design & Creativity	Designing a complete mini-project: animated storytelling	120 min
6	Showcase & Peer Review	Presentation of student projects and peer feedback session	120 min

Figure 3. Workshop Documentation

EVALUATION STAGE

The assessment phase was created to gauge how well the Scratch-based animation workshop enhanced the computational thinking abilities of Al Haraki Islamic middle school students. This phase was separated into two primary parts: student feedback surveys and pre- and post-test evaluations. Assessments were given both before and after the test to gauge the students' development in computational thinking skills such issue decomposition, pattern detection, abstraction, and algorithmic design. Students' opinions about the program's general motivation, ease of learning, and engagement were recorded through feedback surveys. The findings from both tools enabled the researchers to cross-check information and guarantee a thorough assessment of learning objectives. This stage's primary indicators are presented in Table 3 below.

Table 3. Evaluation Indicators of Computational Thinking and Engagement

Indicators	*Measurement Tool*	*Scale Type*	*Description*
Problem Decomposition	Pre-test & Post-test	Score (0–100)	Ability to break down problems into smaller manageable parts
Pattern Recognition	Pre-test & Post-test	Score (0–100)	Capability to identify recurring patterns in problem contexts
Abstraction	Pre-test & Post-test	Score (0–100)	Skill in filtering essential information and ignoring irrelevant details
Algorithmic Design	Pre-test & Post-test	Score (0–100)	Ability to construct step-by-step solutions for given tasks
Student Engagement	Feedback Survey (Likert Scale)	1–5	Students’ level of interest and active participation
Motivation and Satisfaction	Feedback Survey (Likert Scale)	1–5	Students’ motivation and satisfaction with the learning experience

PROGRAM COMPLETE

The community service program's last step, known as "Program Complete," brought all of the activities together and made sure that the learning results would last. At this point, the workshop's creations were formally displayed by groups of participants in a final presentation session, where they exhibited their Scratch-based animations. This exercise encouraged them to confidently communicate computational thinking solutions to professors and peers while also acknowledging their accomplishments. Students' dedication was recognized with certificates of participation, and teachers were given overview instructions on how to incorporate Scratch into their upcoming classes. In order to identify program strengths, problems, and opportunities for growth, the study team also held a final reflection session and asked teachers and students for their opinions. At this point, the program was formally over, and everyone agreed to keep using computational thinking in classrooms. The order of completion tasks for the activities in this stage is shown in table 4 below.

Table 4. Program Closure and Sustainability Plan

Activity	***Description***
Final Project Presentation	Students showcase Scratch-based animation projects in small groups
Recognition & Certificates	Distribution of participation certificates to students and teachers
Teacher Support Materials	Delivery of instructional guides for continued classroom use
Reflection & Feedback Session	Open discussion on program impact, strengths, and challenges
Program Closure	Formal conclusion and reinforcement of sustainability commitment

3. RESULTS AND DISCUSSION

The results show that after participating in Scratch-based animation exercises, students' computational thinking abilities significantly improved. In-depth discussion of the results is provided in this section, along with important trends, student reactions, and the implications for beginning programming instruction at Al Haraki Islamic middle schools.

WORKSHOP DELIVERY AND IMPLEMENTATION

Al Haraki Islamic Middle School's workshop delivery showed how Scratch-based activities gradually improved students' computational thinking abilities. The Scratch environment and fundamental commands were presented to the students in the first session, which aided in the development of their foundational programming skills. Their capacity to break down difficulties grew dramatically as the lessons went on, especially in session two when it came to the storytelling decomposition exercises. Students' ability to identify patterns and use loops to animate repeating movements had significantly improved by the third session, demonstrating a greater comprehension of programming efficiency. In the fourth session, students demonstrated significant improvement in abstraction and modularization by effectively producing reusable code blocks that decreased redundancy.

Lastly, in the fifth session, students created comprehensive storytelling projects that incorporated all of the previously taught topics, showcasing their inventiveness and logical sequencing. As the program went on, teachers noticed increased levels of participation and teamwork as well. All things considered, the methodical scaffolding and incremental rise in task difficulty made it possible for students with varying levels of prior knowledge to stay interested and significantly advance their computational thinking abilities. Figure 4 shows the overall growth in students' computational thinking abilities during the workshop sessions.

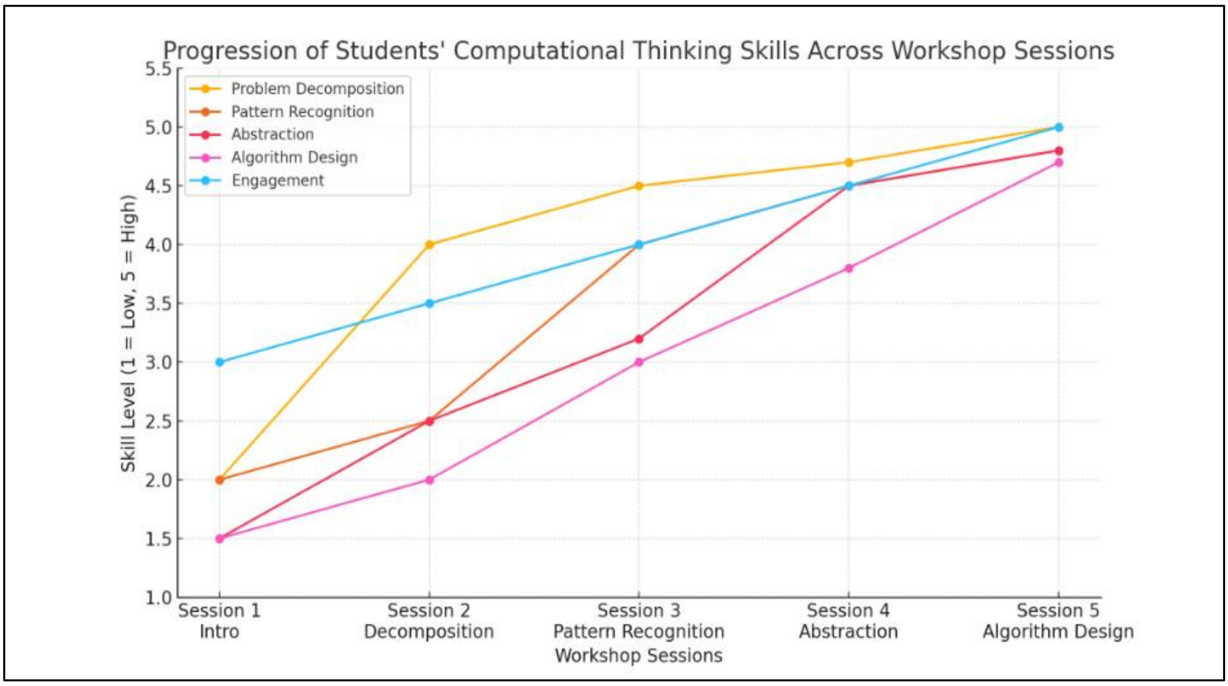

Figure 4. Growth of Computational Thinking Skills Across Workshop Sessions

Based on Figure 4, Strong relationships between the various computational thinking factors are shown by the examination of the workshop's delivery and implementation. The gradual increase in issue decomposition and pattern identification prepared the way for later expansion in abstraction and algorithm design, indicating a step-by-step learning process in which mastery of more complex ideas was aided by earlier abilities. Students were better able to identify recurrent patterns (+2.8) as they gained proficiency in problem decomposition (improvement of +3.0), which bolstered their capacity to abstract and modularize solutions (+3.3). The greatest increase was seen in abstraction, suggesting that higher-order thinking skills were directly improved by controlled exercise in previous sessions. In the meantime, there was a steady rise in student involvement (+2.0), which was positively connected with skill development, indicating that active participation promoted both creative and cognitive development. Students' skill levels increased on average from 2.0 at the start of the workshop to 4.86 by the end, demonstrating that the design of the workshop—which emphasized teamwork and gradually increased in complexity—was successful in enhancing computational thinking skills.

ANALYSIS OF PRE-TEST, POST-TEST, AND FEEDBACK SURVEY FINDINGS

The assessment findings showed that students' computational thinking abilities significantly improved following their participation in the Scratch-based animation workshop. Students performed better on all four computational thinking aspects when comparing their pre- and post-test results. In algorithmic design, the greatest improvement was noted, as the average score rose from 58.4 on the pre-test to 82.1 on the post-test. This implies that student's capacity to formulate methodical approaches to problem-solving improved. The average scores for problem deconstruction also increased significantly, from 61.2 to 79.5, suggesting that students improved their ability to deconstruct difficult problems into smaller, more manageable parts. The post-test scores improved in the areas of pattern recognition and abstraction as well (from 64.8 to 81.0 and from 59.7 to 77.3, respectively), indicating improved abilities to spot reoccurring patterns and isolate important information.

These findings were further corroborated via feedback surveys. Students said they were very engaged (mean score = 4.3) and motivated/satisfied (mean score = 4.5) on a 5-point Likert scale. These findings demonstrate that in addition to improving technical computational thinking skills, the workshop promoted a supportive and inspiring learning atmosphere. The conclusion that Scratch-based animation is a useful tool for fostering computational thinking in middle school children is supported by the correlation between advances in quantitative tests and student feedback. Figure 5 shows the comparison of pre-test and post-test scores as well as the findings of a student feedback survey to better explain these gains.

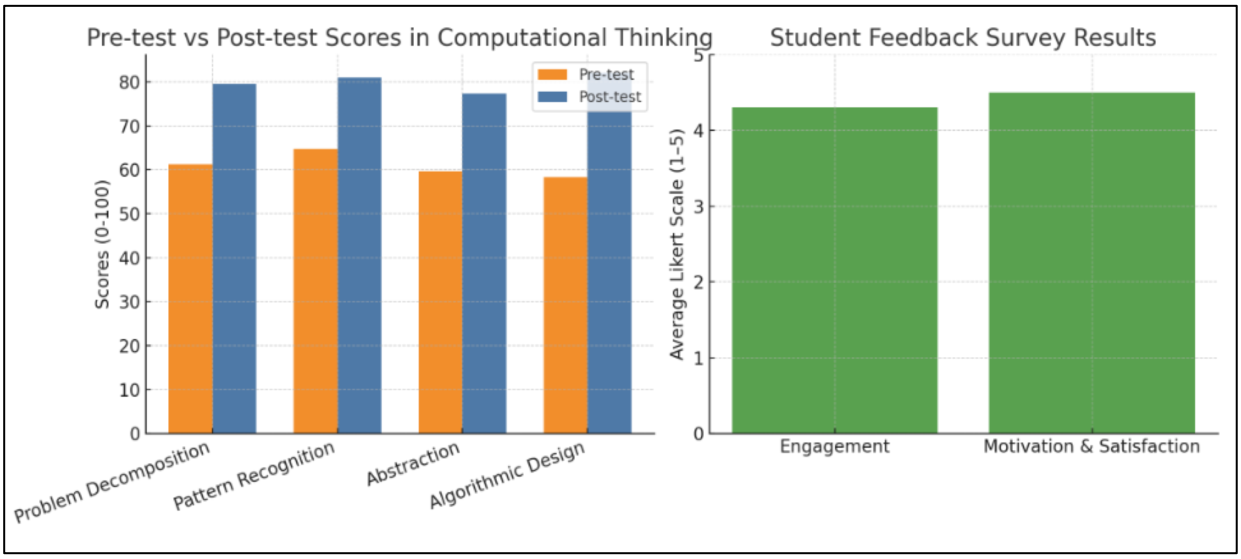

Figure 5. Students Engagement and Motivation Levels from Feedback Survey

PROGRAM REFLECTION AND SUSTAINABILITY INSIGHTS

The final stage of the community service program, successfully consolidated participants learning outcomes and highlighted their growth in computational thinking through the presentation of Scratch-based projects. Students showed off their animations to teachers and peers with assurance, showcasing their inventiveness and problem-solving abilities. The certification process encouraged a feeling of accomplishment and inspired people to keep learning about computational thinking. The educational resources offered were well received by teachers, who said they will help incorporate Scratch into future teaching methods. Reflection sessions showed that teachers stressed the value of ongoing support and resources, while students especially appreciated the chance to exhibit their work. Three key topics emerged from the feedback analysis: (1) increased self-assurance among students when presenting computational solutions; (2) acknowledgment of accomplishments as a source of inspiration; and (3) the necessity of ongoing instructor assistance to guarantee sustainability. All things considered, the closure phase not only acknowledged accomplishments but also reinforced teachers' and students' will to use computational thinking in future learning environments. As shown in Figure 6, the reflection session results can be categorized into three dominant themes: student confidence, recognition and motivation, and teacher support needs.

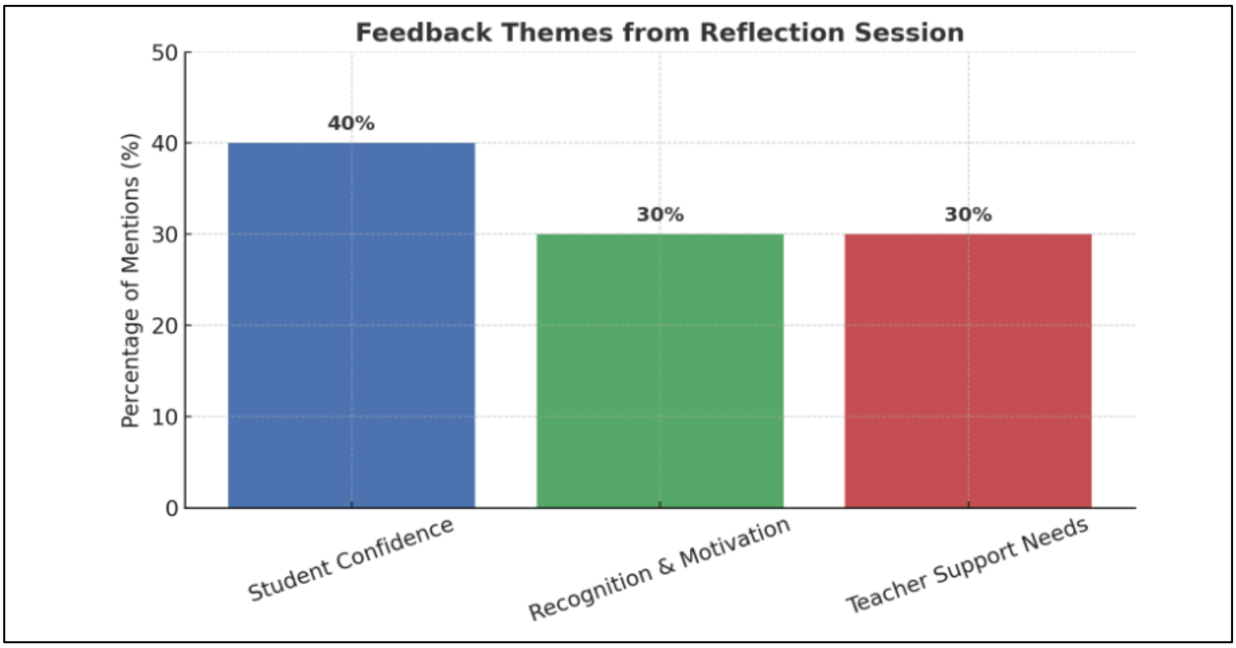

Figure 6. Distribution of Reflection Feedback Themes

The majority of participants (40%) highlighted increased student confidence and how presenting projects based on Scratch greatly improved their capacity to explain solutions including computational thinking. Meanwhile, 30% of respondents said that motivation and recognition were essential for maintaining excitement because they promoted ongoing participation by rewarding students for their accomplishments and offering certificates. Teachers emphasized the necessity of continuous teaching materials and guidance to properly incorporate Scratch into classroom practices in the equally significant subject of the requirement for teacher support (30%). According to this balance of themes, maintaining computational thinking in education would necessitate a dual emphasis on recognition mechanisms and ongoing teacher assistance, even though student confidence increase is the most notable accomplishment.

4. CONCLUSION

The effectiveness of Scratch-based animation in improving middle school pupils' computational thinking was shown by this community service project. Students' skills in problem decomposition, pattern recognition, abstraction, and algorithmic design were greatly enhanced by the workshop activities, which also increased their motivation, creativity, and engagement. In order to maintain the incorporation of computational thinking into subsequent sessions and guarantee long-term impact, educators were provided with useful materials. While the program shown promising results, ongoing assistance and organized teaching methods are necessary to enhance computational thinking abilities and maintain its use in classrooms.

ACKNOWLEDGMENT

Title of acknowledgment section must be written without section number. Acknowledgments should be written simply 50-100 words. The research funder is usually mentioned, including the contract number, as this will also be as documentation for the funder that the activity has been published.

REFERENCES

Çelik, E., & Sevgi, S. (2025). Middle school students' views on mathematical situations designed with Scratch. *Journal of Educational Technology and Online Learning*, *8*(2), 190–205. https://doi.org/10.31681/jetol.1636413

Fanchamps, L. J. A., & Vansteenkiste, M. (2023). The Effect of Visual Programming Environments on the Development of Computational Thinking and the Influence of Self-Regulating Ability in Upper Primary School Children. *In CTE-STEM Asia-Pacific Society for Computers in Education*, 8–10.

Fuentes-Peñailillo, F., Gutter, K., Vega, R., & Silva, G. C. (2024). Transformative Technologies in Digital Agriculture: Leveraging Internet of Things, Remote Sensing, and Artificial Intelligence for Smart Crop Management. *Journal of Sensor and Actuator Networks*, *13*(4), 39. https://doi.org/10.3390/jsan13040039

Giménez, A., Bordoy, S., Sánchez, A., López-Zamora, M., Sopena, J. M., & Luque, J. L. (2021). A supplemental <scp>computer-assisted</scp> intervention programme to prevent early reading difficulties in Spanish learners: A stratified random control trial. *Journal of Computer Assisted Learning*, *37*(2), 510–520. https://doi.org/10.1111/jcal.12504

Holstein, S., & Cohen, A. (2025). Scratch teachers' perceptions of teaching computational thinking with school subjects in a constructionist approach. *Thinking Skills and Creativity*, *56*, 101772. https://doi.org/10.1016/j.tsc.2025.101772

Huizinga, T., Nieveen, N., & Handelzalts, A. (2019). Identifying Needs for Support to Enhance Teachers' Curriculum Design Expertise. In *Collaborative Curriculum Design for Sustainable Innovation and Teacher Learning* (pp. 115–137). Springer International Publishing. https://doi.org/10.1007/978-3-030-20062-6_7

Lei, H., Chiu, M. M., Li, F., Wang, X., & Geng, Y. (2020). Computational thinking and academic achievement: A meta-analysis among students. *Children and Youth Services Review*, *118*, 105439. https://doi.org/10.1016/j.childyouth.2020.105439

Nguyen, L. Q., & Le, H. Van. (2024). Challenges in EFL Constructivist Classrooms from Teachers' Perspectives: A Case Study in Vietnam. *Sage Open*, *14*(2). https://doi.org/10.1177/21582440241245187

Papadakis, S., Kalogiannakis, M., Orfanakis, V., & Zaranis, N. (2017). The Appropriateness of Scratch and App Inventor as Educational Environments for Teaching Introductory Programming in Primary and Secondary Education. *International Journal of Web-Based Learning and Teaching Technologies*, *12*(4), 58–77. https://doi.org/10.4018/IJWLTT.2017100106

Perez, L., Clark, H. F., Hadad, R., Nava, I., & Giannotti, M. (2023). CT for equity: situating computational thinking, computer science, and social justice in a math and science teacher education program. In *International Encyclopedia of Education (Fourth Edition)* (pp. 312–320). Elsevier. https://doi.org/10.1016/B978-0-12-818630-5.13007-6

Putra, G. M. C., Prasetyaningtyas, F. D., Ansori, I., Kurnianto, B., Wahyuni, N. I., Xinzhe, Z., & Nurmanto, T. N. (2025). Effectiveness of Project-Based Scratch Programming to Improve Problem-Solving Skills of Elementary School Students. *International Journal of Social Learning (IJSL)*, *5*(2), 460–478. https://doi.org/10.47134/ijsl.v5i2.417

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, *28*(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Romano, S., Capece, N., Erra, U., Scanniello, G., & Lanza, M. (2019). On the use of virtual reality in software visualization: The case of the city metaphor. *Information and Software Technology*, *114*, 92–106. https://doi.org/10.1016/j.infsof.2019.06.007

Sawada, C., Terazono, M., Hagino, T., & Hattori, T. (2025). An Analysis of K-12 Programming Education Utilizing the Minecraft Metaverse Amidst the COVID-19 Pandemic. *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*, 1615–1616. https://doi.org/10.1145/3641555.3705122

Sidiq, M. R., Napitupulu, E., & Farihah, F. (2025). Enhancing Computational Learning through Visual Programming Media: An Empirical Study of Academic Achievement. *JINAV: Journal of Information and Visualization*, *6*(1), 1–9. https://doi.org/10.35877/454RI.jinav4019

Stewart, W., & Baek, K. (2023). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. *International Journal of Computer Science Education in Schools*, *6*(1), 35–58. https://doi.org/10.21585/ijcses.v6i1.156

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, *4*, 100122. https://doi.org/10.1016/j.caeo.2023.100122

Wei, X., Saab, N., & Admiraal, W. (2021). Assessment of cognitive, behavioral, and affective learning outcomes in massive open online courses: A systematic literature review. *Computers & Education*, *163*, 104097. https://doi.org/10.1016/j.compedu.2020.104097