Journal of Applied Business and Technology

Improving Student Learning in Web Programming through Practical Assessment and Project Exhibitions

Meliana Christianti Johan [a*], Bernard Renaldy Suteja [a], Rafael Yefta Wijaya [a], Joshua Christian Benedict [a], Alfian Febina Santika [a]

[a] Department of Informatics Engineering, Universitas Kristen Maranatha, Indonesia

Article History
Received
29 July 2025
Received in revised form
5 November 2025
Accepted
11 January 2026
Published Online
31 January 2026

*Corresponding author
meliana.christianti@it.maranatha.edu

Abstract
Assessment is one way to improve the quality of learning, so it must be designed to be relevant, appropriate, valid, and reliable in accordance with the objectives and outcomes of the course. In the Basic Web Programming course, conventional assessments such as homework and written exams often fail to measure students' ability to transfer theoretical knowledge into more practical applications. This study adopts the concept of project-based learning to identify key assessment needs and develop a student-centered evaluation model. Using a mixed-methods approach, this study involved 61 first-semester Information Technology (IT) students. The resulting model integrates four components: practical assignments, comprehensive programming projects, practical exams, and public exhibitions of student work. This study is supported by a rubric that holistically measures technical skills, conceptual understanding, teamwork, creativity and innovation, engagement, and problem solving. The results of the implementation in the odd semester of 2023/2024 show that 95.7% of respondents have a high level of satisfaction with the practical tasks, projects, and exhibitions. The survey results received 91.3% positive feedback from students and 97.3% high satisfaction from external visitors. This research can evaluate learning in basic web programming classes and enable educators to improve the quality of learning in line with the learning outcomes that students must achieve.

Keywords: Assessment; Learning Outcomes; Programming; Project-Based Learning; Student Engagement
DOI: https://doi.org/10.35145/n59xzn61
SDGs: Quality Education (4); Decent Work and Economic Growth (8); Industry, Innovation, and Infrastructure (9); Partnerships for the Goals (17)

1.0 INTRODUCTION

Learning web programming is now a big part of computer science and information technology classes (Mehmood et al., 2020) and web design is an important part of building websites because it is seen as high quality when it focuses on how well it works, how it looks, and how easy it is to use (Pradana et al., 2023). Assessment is a crucial component of learning (Elmaleh, 2021). However, the main challenge in teaching programming is to test students that fully measure their technical skills, understanding of ideas, and hands-on abilities. Programming assessments are not simply about checking “code correctness” and need to assess integrated creative, conceptual, and practical skills. Conventional assessments that focus on theory and homework are often inadequate for measuring these dimensions.

Bridson & Fleming (2021) identified serious problems with the traditional assessment approach using homework assignments. Instructors found that homework assignments were unable to accurately assess students' coding skills. New technology in information and communication has changed the way people learn, especially in programming classes (Barra et al., 2020). Learning through projects and showing off student work has been shown to help students feel more motivated and interested in programming. Studies show that using automatic grading and game-like features can make students more involved, excited to learn, and confident in programming (Hellín et al., 2023). Initial findings in the Basic Web Programming course show that conventional midterm/final exams are ineffective in testing understanding, thus necessitating their replacement with project-based practical exams and exhibitions of work as authentic assessments.

A main challenge in teaching programming is helping students build problem-solving skills they can use in real life, not just teaching them to write code that is correct. To address this, other current methods have been

suggested, such as Automated Programming Assessment Systems and Project-Based Learning. These will be discussed more in literature review.

2.0 LITERATURE REVIEW

Information technology, which has evolved to meet human needs and facilitate tasks (Novaliendry et al., 2021), is now increasingly used to help students achieve learning outcomes—the knowledge they must acquire, understand, and demonstrate. These outcomes, which often hold deeper significance in the professional world (Walters et al., 2021), are pursued using a variety of learning materials, including course documents, lecture notes, laboratory instructions, and practical assignments (Mai et al., 2022).

Existing research discusses about Automated Programming Assessment Systems can check computer code automatically and give students fast feedback, which has been shown to help them learn better (Mekterović et al., 2023). However, using these systems still faces problems with handling many users at once and working with other ways of testing students. Research shows that learning programming through hands-on tasks, projects, and showing their work can greatly improve students' understanding and skills. This way, students can use what they have learned in real projects and get feedback from classmates and teachers (Takaoka et al., 2015). Lubarda et al. (2024).

Assessing programming education needs more than just checking if code is correct (Mekterović et al., 2023), creative programming, rather than just following procedures, is the best way to develop computational thinking. (Romero et al., 2017). Assessment means gathering, reviewing, and using information about educational programs to help students learn and grow (ElAtia et al., 2021) and studying learning behavior helps us see how students use version control systems, and these patterns can be used to improve computing education (Chen et al., 2024). Ubaidullah & Hamid (2019) found that most teaching still depends only on discussion and problem-solving, and Kuo & Kuo (2023) measured student performance by looking at final website project scores. Automated assessment (APAS) in programming: benefits, scalability constraints, and integration with other assessments; its relevance for rapid feedback and learning improvement. Students can learn programming well if they are frequently given assignments to solve programming problems individually. This approach encourages students to think logically independently, thereby improving their programming skills (Afandi, 2021).

Students with stronger knowledge integration achieved better results, which supports two conclusions: Design-Based Learning is effective for teaching web design, and this integration is a critical component of successful learning (Tsai et al., 2022). In project-based learning (PBL) and other PBL methods, the goal is not just to learn, like in problem-based learning (López-Pimentel et al., 2021). The hardest thing about teaching programming is getting students to do more than just write right code (Glagolev et al., 2021; Takaoka et al., 2015).

3.0 METHODOLOGY

This study uses mixed-method approach to evaluate how effective integrated assessment tools are in web programming courses. The research follows a research and development model with four stages: needs analysis, instrument design, implementation, and evaluation.

This research involved students from the Basic Web Programming course during the 2023/2024 odd semester. Our participants were 61 first-year undergraduate engineering students enrolled in Basic Web Programming during the fall 2023/2024 semester at Maranatha Christian University. The student work exhibition was introduced as a new part of the assessment process, using both online and offline formats. It allowed students to present their major assignments and apply what they had learned. The exhibition followed a model from project-based learning research.

Instruments and data:

1. Assessment component scores: practical assignments, projects, practical exams, exhibitions.
2. Student satisfaction survey (scale of 1–10) and exhibition visitor satisfaction survey (scale of 1–5).

4.0 RESULTS AND DISCUSSION

The results of this study reveal that most students believe programming assignments and projects help them understand the material better. The graph in Figure 1 shows the results of a poll asking students how satisfied they were with the quality of the classes they had received. Quantitative data based on the results of the survey shows that 47.8% of students (22 out of 46 respondents) gave the highest satisfaction rating (10/10) for the quality of the course, while 95.7% expressed high overall satisfaction (rating 8-10) with the practical assignments and programming projects. Based on the survey results, this study developed an assessment method consisting of four components, namely practical assignments, programming projects, practical exams, and student work exhibitions.

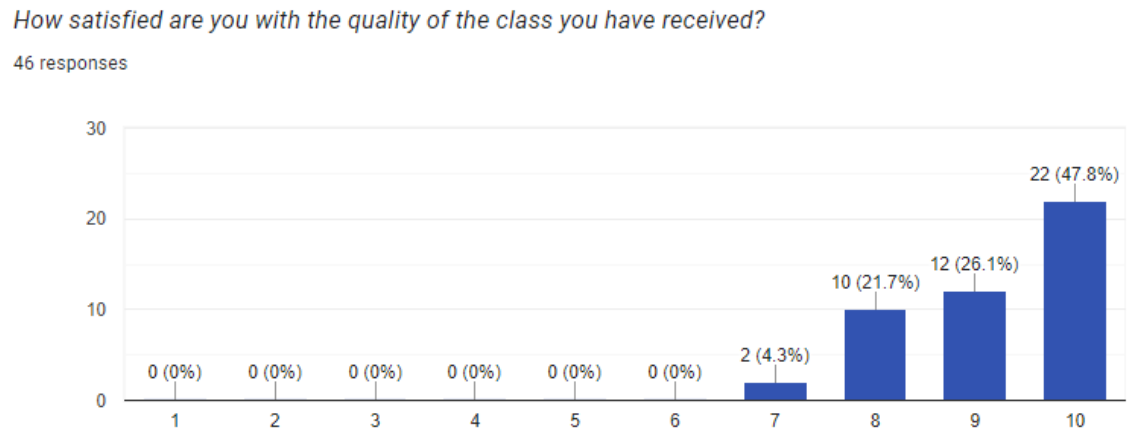

Figure 1. Graphical Representation of Student Satisfaction

The evaluation of the exhibition of work showed that 37.0% of students gave the highest rating (10/10), 28.3% gave a rating of 9/10, and 26.1% gave a rating of 8/10 for the quality of the exhibition of work held to display all the work of basic web programming students. Overall, this resulted in a 91.3% high satisfaction rate with exhibition experience. Figure 2 shows the results of a student satisfaction survey regarding the recent student work exhibition, which was the first event of its kind. In addition, Figure 3 shows the results of a guest satisfaction survey for the online and in-person versions of the exhibition. Guest satisfaction ratings across both online and in-person exhibition platforms yielded remarkable results: 78.2% of respondents (575/735) gave the highest satisfaction score (5/5), 19.0% gave a rating of 4/5, resulting in an overall high satisfaction rate of 97.3% with the student work exhibition experience.

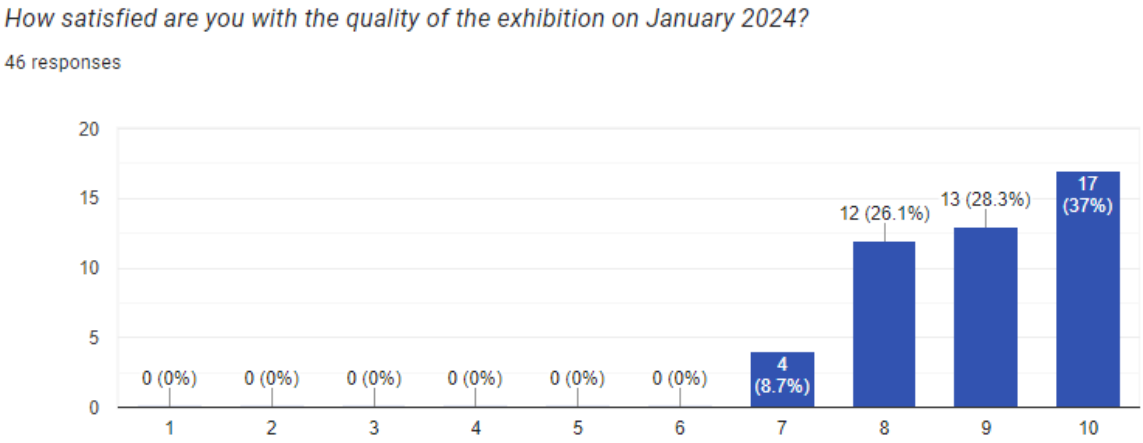

Figure 2. Graphical Representation of Student Satisfaction Survey Findings on the Work Exhibition

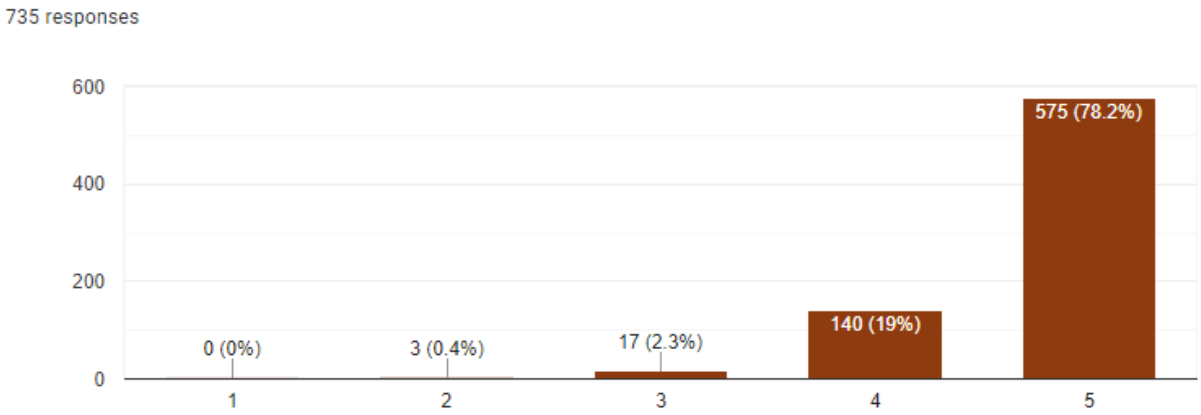

Figure 3. Exhibition Guest Satisfaction Survey Results (Online and Offline)

Guest evaluators were even more impressed. Both in online and face-to-face formats, 78.2% gave the highest satisfaction score (5 out of 5), and another 19.0% gave a score of 4 out of 5, for a total satisfaction rate of 97.3%. To understand which assessment components were most important, we performed a correlation analysis, as shown in Table 1.

Table 1. Pearson Correlation Analyses

Assessment Component	Performance Dimension	r	p-value	n
Practical Assignment Engagement*	Technical Skills Rubric Score	.51	.003**	61
Programming Project Quality	Conceptual Understanding Score	.58	.001**	61
Exhibition Participation	Development of Communication Skills	.44	.009**	61
Peer Feedback Engagement	Teamwork Dimension Score	.39	.018*	61

*Engagement measured by submission timeliness and completion rates
**p < 0.01; *p < 0.05

After holding the exhibition and reviewing the survey results, students reported that they gained valuable experience compared to taking midterm and final exams to test their understanding. Therefore, the best approach was to replace them with project-based practical tests and work exhibitions. Combining learning outcomes and creating practical evaluation tools enhanced to engage students more actively and support teachers in their roles. The technical skills assessment component of basic web programming education evaluates how well students can write, read, and execute programming code.

Besides learning technical skills, students are also evaluated on their ability to work effectively with others and their creativity in generating new project ideas. To figure out how much each person contributes, they use peer assessment, where students rate each other's work. The Assessment and Evaluation Instrument enhance Basic Web Programming Learning with six main assessment features, as shown in Table 2, based on teaching and learning outcomes.

Table 2. Assessment and Evaluation Tools for Enhancing Basic Web Programming Learning Quality

Dimensions	Instrument	Description	Method	Schedule
Technical Skills	Hands-on Tasks	Evaluating learners' capability to create and comprehend web programming code via practical activities.	Direct observation and hands-on testing	Each practical session
	Coding Projects	Tasks focused on building advanced web applications	Project-oriented evaluation	At the end of each module or semester

Dimensions	Instrument	Description	Method	Schedule
		that incorporate all learned concepts.		
	Midterm Exam	Evaluating learners' grasp of concepts and competencies acquired during the initial semester half.	Written and practical examinations	Twice yearly
	Final Exam	Complete assessment of comprehension and web programming competencies gained over the entire term.	Written and practical examinations	Once per term
Conceptual Understanding	Midterm Exam	Theory-based examination assessing comprehension of fundamental web programming principles like HTML, CSS, and JavaScript.	Written examinations	Twice yearly
	Final Exam	Comprehensive theoretical examination evaluating conceptual mastery of all course content.	Written examinations	Once per term
Teamwork	Coding Projects	Group-based projects for web application development.	Project monitoring and documentation	End of each unit or term
	Project Assessment	Evaluating individual student contributions within group assignment projects.	Survey forms or evaluation sheets	Each project cycle
Creativity and Innovation	Coding Projects	Assessing learners' capacity to develop	Project based assessment	End of each unit or term

Dimensions	Instrument	Description	Method	Schedule
		original and inventive solutions via programming assignments.		
	Project Assessment	Students demonstrate their completed projects.	Oral presentation and group discussion	End of each unit or project
Engagement and Participation	Practical Assignment	Assessing student involvement during practical learning activities.	Direct observation	Each practical session
	Classroom Interaction	Student engagement in class dialogue and Q&A activities.	Direct observation	Every class meeting
Problem Solving	Practical Assignment	Practical activities focused on resolving web programming challenges.	Direct observation and hands-on testing	Each practical session
	Coding Projects	Assignments challenging students to address real-world issues using web programming principles.	Project-oriented evaluation	End of each unit or term

Besides helping students do well, this combined assessment model aims to connect what students learn with what companies need. In today's technology field, just having technical skills is not enough. Companies want people who can work well with others, communicate clearly, and think of new solutions.

Accreditation alignment: Map to program learning outcomes (e.g., teamwork, communication, critical thinking, lifelong learning, professionalism) through evidence of assignments, projects, practical exams, and exhibitions; explain how rubrics ensure OBE assessments that evaluate outcomes authentically and triangulation. Practical implications: The integration of these assessment components supports the role of lecturers as facilitators and increases student engagement, with the support of rapid feedback from automated systems and external validation through exhibitions.

Limitations: The study was conducted in one course at one institution in one semester with 61 students, so generalizations should be made with caution. The satisfaction survey was self-reported (students n=46), potentially containing self-reporting bias and social bias. Log-level analysis describes usage, but does not yet assess process quality (e.g., revision depth or commit quality) longitudinally. Not all inferential results (e.g., t-tests/ANOVA) are available in the initial implementation, so causal claims must be limited to associations. Cross-class/institutional replication, quasi-experimental design, enrichment of process analytics metrics (version control/repo analytics), and tracking of long-term impact on advanced courses are needed.

As a result of the implementation, Figure 4 shows the website created by student representatives containing complete details of the event, program timeline, and links to online exhibitions. It successfully displays a collection of photos and videos featuring student projects, as well as an interactive chat room for all visitors. This website will ensure that the event runs smoothly and attracts more visitors, both physically and digitally.

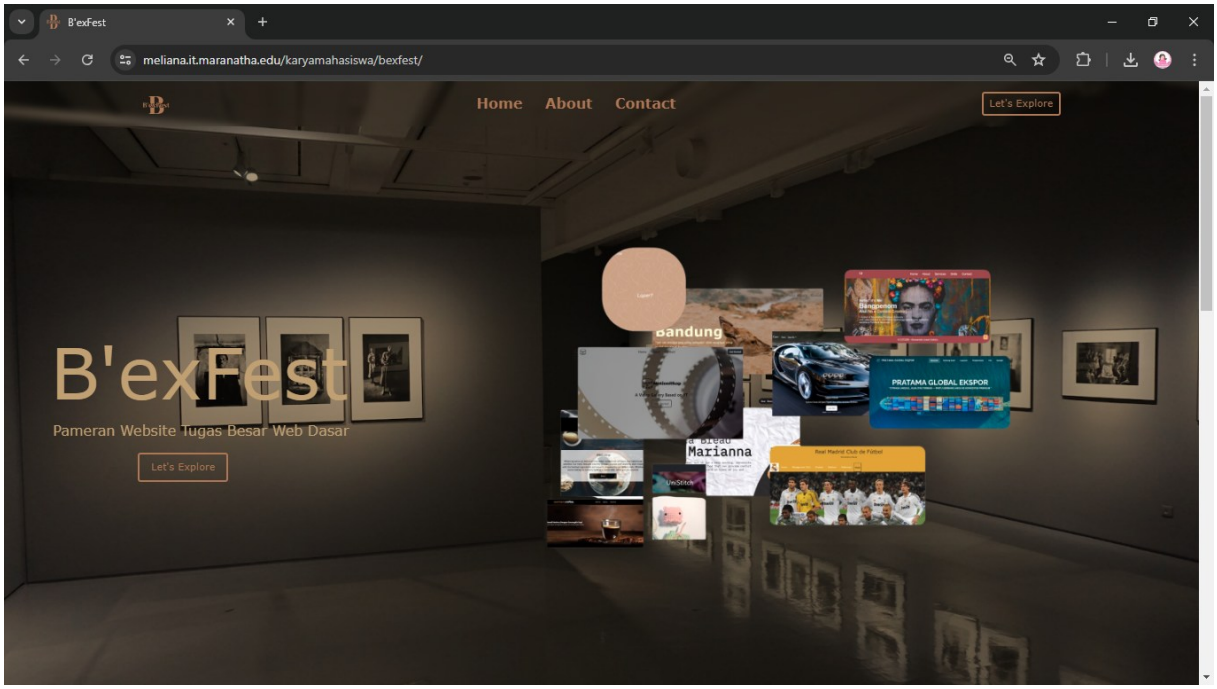

Figure 4. Student Work Results Creating Exhibition Event Websites

5.0 CONCLUSION

This research finds that creating the right ways to test students in Basic Web Programming is very important for better learning and student success. We found the main testing needs by carefully studying the subject, showing that good tests should check hands-on web programming skills, clear understanding of ideas, creativity in projects, and teamwork. The assessment system for this study has four main parts, namely practical assignments, programming projects, exams, and student displays. One of the most significant improvements was in the results of the students' web projects. The work displayed by the students showed that they had achieved their main task by being able to use the material they had learned. The work displayed shows that students succeeded in creating creative work. The results of this study provide a solid plan for creating a better, more comprehensive, and more useful way to test students, which helps schools improve teaching and student outcomes, especially in web programming. In addition, the practice-centered integrated assessment model, which includes assignments, projects, practical exams, and work exhibitions, shows high satisfaction from students and has proven to have the support of stakeholders. These findings reinforce the urgency of replacing the dominance of written exams with authentic assessments that are in line with industry needs and learning outcomes.

Acknowledgement

We would like to thank Maranatha Christian University and LPPM for their funding and support of this research. Special thanks to participating students and faculty members who contributed to this study.

References

Afandi, K. (2021). Identifikasi Proses yang Mempengaruhi Kemampuan Pemrograman Mahasiswa Baru tanpa Pengalaman Pemrograman. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *8*(4), 1785–1795. https://doi.org/10.35957/jatisi.v8i4.1120

Barra, E., López-Pernas, S., Alonso, A., Sánchez-Rada, J. F., Gordillo, A., & Quemada, J. (2020). Automated Assessment in Programming Courses: A Case Study during the COVID-19 Era. *Sustainability (Switzerland)*, *12*(18), 1–24. https://doi.org/10.3390/SU12187451

Bridson, K., & Fleming, S. D. (2021). Frequent, Timed Coding Tests for Training and Assessment of Full-Stack Web Development Skills: An Experience Report. *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, *2*, 24–30. https://doi.org/10.1145/3408877.3432549

Chen, J., Lau, S., Leinonen, J., Terragni, V., & Giacaman, N. (2024). Detecting Learning Behavior in Programming Assignments by Analyzing Versioned Repositories. *IEEE Access*, *12*(December), 188828–188844.

https://doi.org/10.1109/ACCESS.2024.3514843

ElAtia, S., Ipperciel, D., Zaiane, O., Bakhshinategh, B., & Thibaudeau, P. (2021). Graduate Attributes Assessment Program. *International Journal of Information and Learning Technology*, *38*(1), 117–134. https://doi.org/10.1108/IJILT-03-2020-0025

Elmaleh, J. (2021). *Improving Student Learning in an Introductory Programming Course Using Flipped Classroom and Competency Framework*. *February*. https://doi.org/10.1109/EDUCON.2017.7942823

Glagolev, V., Bazhenov, R., Kuldysheva, G., Ledovskikh, I., & Maximov, V. (2021). The use of real technical tasks for the implementation of websites for the formation of professional skills of web developers. *Journal of Physics: Conference Series*, *2001*(1). https://doi.org/10.1088/1742-6596/2001/1/012029

Hellín, C. J., Calles-Esteban, F., Valledor, A., Gómez, J., Otón-Tortosa, S., & Tayebi, A. (2023). Enhancing Student Motivation and Engagement through a Gamified Learning Environment. *Sustainability (Switzerland)*, *15*(19). https://doi.org/10.3390/su151914119

Kuo, Y. T., & Kuo, Y. C. (2023). African American Students' Academic and Web Programming Self-Efficacy, Learning Performance, and Perceptions towards Computer Programming in Web Design Courses. *Education Sciences*, *13*(12). https://doi.org/10.3390/educsci13121236

López-Pimentel, J. C., Medina-Santiago, A., Alcaraz-Rivera, M., & Del-Valle-soto, C. (2021). Sustainable project-based learning methodology adaptable to technological advances for web programming. *Sustainability (Switzerland)*, *13*(15), 1–25. https://doi.org/10.3390/su13158482

Lubarda, M. V., Phan, A. M., Schurgers, C., Delson, N., Ghazinejad, M., Baghdadchi, S., Minnes, M., Kim, M., Pilegard, C., Relaford-Doyle, J., Sandoval, C. L., & Qi, H. (2024). Virtual pair programming and online oral exams: effects on social interaction, performance, and academic integrity in a remote computer programming course. *Computer Science Education*, *3408*(May). https://doi.org/10.1080/08993408.2024.2344401

Mai, T. T., Bezbradica, M., & Crane, M. (2022). Learning behaviours data in programming education: Community analysis and outcome prediction with cleaned data. *Future Generation Computer Systems*, *127*, 42–55. https://doi.org/10.1016/j.future.2021.08.026

Mehmood, E., Abid, A., Farooq, M. S., & Nawaz, N. A. (2020). Curriculum, Teaching and Learning, and Assessments for Introductory Programming Course. *IEEE Access*, *8*, 125961–125981. https://doi.org/10.1109/ACCESS.2020.3008321

Mekterović, I., Brkić, L., & Horvat, M. (2023). Scaling Automated Programming Assessment Systems. *Electronics (Switzerland)*, *12*(4). https://doi.org/10.3390/electronics12040942

Novaliendry, D., Huda, A., Cuhanazriansyah, M. R., Sani, H. K., Hendra, H., & Karnando, J. (2021). E-Learning Based Web Programming Course in the COVID 19 Pandemic Time. *International Journal of Interactive Mobile Technologies*, *15*(20), 117–130. https://doi.org/10.3991/ijim.v15i20.23749

Pradana, F., Setyosari, P., Ulfa, S., & Hirashima, T. (2023). Development of Gamification-Based E-Learning on Web Design Topic. *International Journal of Interactive Mobile Technologies*, *17*(3), 21–38. https://doi.org/10.3991/ijim.v17i03.36957

Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, *14*(1). https://doi.org/10.1186/s41239-017-0080-z

Takaoka, E., Yonezu, T., & Namekawa, T. (2015). Programming education at high schools and universities: Design, development, and assessment. *IPSJ Online Transactions*, *8*(2015), 1–8. https://doi.org/10.2197/ipsjtrans.8.1

Tsai, C. Y., Shih, W. L., Hsieh, F. P., Chen, Y. A., & Lin, C. L. (2022). Applying the design-based learning model to foster undergraduates' web design skills: the role of knowledge integration. *International Journal of Educational Technology in Higher Education*, *19*(1). https://doi.org/10.1186/s41239-021-00308-4

Ubaidullah, N. H., & Hamid, J. (2019). A web-based learning programming portal: Do instructors need it to enhance novice students' computational thinking skill? *International Journal of Innovative Technology and Exploring Engineering*, *8*(9), 1945–1958. https://doi.org/10.35940/ijitee.i8513.078919

Walters, P., Nieweg, M., & Watson, J. (2021). *Learning Outcome Oriented Programmatic Assessment*. 1–17. https://doi.org/10.48550/arXiv.2101.10133