International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 31, No. 6, June, 2026, pp. 1939-1947
Journal homepage: http://iieta.org/journals/isi

An Integrated Practical Assessment Model for Web Programming Education in Smart Campus Environments

Meliana Christianti Johan[1,2], Bernard Renaldy Suteja[2], Radiant Victor Imbar[2*], Armein Z. R. Langi[1]

[1] School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia
[2] Faculty of Smart Technology and Engineering, Maranatha Christian University, Bandung 40164, Indonesia

Corresponding Author Email: radiant.vi@it.maranatha.edu

https://doi.org/10.18280/isi.310616

Received: 3 February 2026
Revised: 2 April 2026
Accepted: 15 April 2026
Available online: 30 June 2026

Keywords:
higher education, learning assessment model, practical assessment, project-based learning, smart campus, web programming education

ABSTRACT

Assessment methods in programming education should evaluate not only theoretical understanding but also students' ability to apply knowledge in practical development tasks. Traditional examinations and assignment-based assessments often provide limited evidence of students' programming competence, collaboration, and problem-solving abilities. This study proposes an integrated practical assessment model for Basic Web Programming courses within a smart campus learning environment. The proposed model combines four assessment components: practical assignments, programming projects, practical examinations, and student work exhibition. A mixed-methods research design was adopted, involving needs analysis, model implementation, and evaluation with 61 first-year Information Technology students. Data were collected through assessment records, student satisfaction surveys, exhibition feedback, classroom observations, and peer evaluation information. The results showed that 95.6% of respondents expressed high satisfice.

1. INTRODUCTION

The implementation of a smart campus environment can improve the quality of education through the enhancement of student achievement, participation and motivation to learn. It also encourages integrated assessment methodologies such as displays of student work and practical projects that allow learners to apply theoretical knowledge in real-world scenarios while improving creativity and collaboration abilities. The use of modern digital technology allows higher education institutions to improve the quality of education and to provide more effective, efficient and attractive services to the academic community [1, 2].

Web programming has become a basic part of the teaching of Computer Science and Information Technology [3]. Improvements in information and communication technology have altered teaching approaches and had a great impact on programming education [4]. More recently, advances in artificial intelligence have opened up new avenues for personalized and adaptable learning. Sendeku et al. [5] proposed an AI-enabled adaptive learning framework that integrates cognitive, emotional and cultural learner characteristics to personalize learning experiences and enhance learner engagement. These achievements illustrate the potential of developing technology to promote student-centered and adaptive educational environments.

Project Based Learning (PBL) and student work exhibitions have been widely acknowledged as successful techniques to increase students' motivation and involvement in programming education. Previous research has shown that the combination of automated assessment with gamification tactics can boost participation and learning motivation as well as student confidence in programming related courses [6]. Despite these benefits, the automated programming assessment systems are still facing the challenges of scalability and integration with complementary assessment approaches [7]. Lubarda et al. [8] observed that first- and second-year students were more inclined towards participating in virtual pair programming activities compared to senior students. Similarly, Ou et al. [9] pointed out the necessity of creating an inclusive digital education ecosystem for preparing future generations for an artificial intelligence-driven society. Furthermore, engineering education is supposed to produce graduates who are not just technically proficient but also ethically responsible professionals [10].

Empirical studies have shown that instructional programming involving practical assignments, programming projects, and student work exhibitions can improve conceptual understanding and practical competence [11]. Considering these findings, the current work designs and assesses an integrated evaluation model for a Basic Web Programming course in a smart campus learning environment.

This study attempts to answer the following research questions:

RQ1: What assessment needs are identified in the Basic Web Programming course? (Context: students' learning experiences and course assessment activities)

RQ2: How to integrate practical assignments, programming projects, practical examinations and student work exhibitions into a structured assessment model of Basic Web Programming?

RQ3: How do students and visitors of the exhibition perceive the implementation of the integrated assessment model?

The planned output of this project is an integrated assessment model which can support instructors in assessing student learning in Basic Web Programming holistically. The model provides a practical approach to the creation of assessment activities that relate technical competence, conceptual understanding, creativity, cooperation, engagement and problem-solving in a learning environment in a smart campus.

2. LITERATURE REVIEW

Information technology encompasses the technological tools and systems developed to facilitate human activities [12]. Students interact with four main types of learning materials: General course documents, Lecture notes read during lecture sessions, Instructions and examples to follow in lab sessions, and Practice activities such as programming assignments [13].

This definition emphasizes that assessment is a systematic process of collecting, reviewing, and using information about educational programs undertaken for the purpose of improving student learning and development [14]. Understanding learning behavior allows for the exploration of students' interactions with version control systems and the identification of behavioral patterns that can inform the continuous improvement of computing education [15].

Barra et al. [4] showed that the transformation of assessment methods from face-to-face to fully online requires the use of automated student-centered assessment tools. Hellín et al. [6] showed that the implementation of gamification can increase students' motivation to attend classes, study harder, and boost their self-confidence.

Design-Based Learning (DBL) has been proven effective in improving students' web design skills. Research by Tsai et al. [16] shows that the use of DBL strategies in learning can improve the web design skills of students in the experimental group, especially those with poor web design skills at the initial stage. DBL involves constructing meaningful and engaging situations for learners to design works where they can apply prior knowledge and problem-solving skills. Research indicates that students with higher levels of knowledge integration experience greater improvements in their web design skills. In the context of web design learning, knowledge integration involves dynamic interdisciplinary integration for website layout, materials, colors, and functions. Studies show that students who have the ability to integrate knowledge can improve the effectiveness of DBL learning.

PBL is a learning strategy that focuses on the application of knowledge through the development of real projects, unlike problem-based learning, which emphasizes the acquisition of knowledge [17]. Discussion and problem-solving techniques are the most dominant strategies used because they help students develop logical and systematic problem-solving skills [18]. PBL is defined as a key active learning strategy that involves teamwork on projects.

Student work exhibition as part of the assessment process has shown a significant positive impact on student understanding. Studies indicate that presenting work and receiving direct feedback are highly beneficial for student learning. The main challenge in programming education is developing problem-solving skills that can be applied in real-world situations, not just the ability to write syntactically correct code. Research shows that practical application-based learning is more effective in developing programming skills that are relevant to industry needs.

Computational thinking (CT) is defined as a set of cognitive and metacognitive strategies that engage learners in active design and creation processes, mobilizing computational concepts and methods. CT is not only about understanding algorithms, but also includes problem solving, cooperative work, and attitudes in the broader concept of CT. The most effective development of CT occurs in the context of creative programming, not just procedural learning. There are five levels of creative engagement in educational programming activities [19]. Higher levels provide greater scope for creativity and in-depth CT development: Levels 1-2: Passive exposure and step-by-step procedural programming, Level 3: Individual content creation through programming, Level 4: Co-creation of content through team programming, Level 5: Participatory co-creation of knowledge through programming.

Ubaidullah and Hamid found that the dominant teaching method currently relies solely on discussion and problem solving. Although all instructors use this technique, this conventional approach has been criticized for producing passive students, especially in large classes, due to the lack of interaction between students and teachers [20]. The previous study [21] measured learning performance based on the final website project scores of students who combined the use of Cascading Style Sheets (CSS), HyperText Markup Language (HTML), and JavaScript. This project was a comprehensive application of all the skills learned during the 14-week course. To summarize the reviewed studies, Table 1 summarizes the main methodological approaches, advantages, and limitations of previous work related to programming education assessment, PBL, automated assessment, and student work exhibition.

Table 1. Summary of key studies related to programming education assessment

Study	Method/Focus	Advantages	Limitations/Research Gap
Barra et al. [4]	Automated assessment in programming courses	Provides scalable and immediate feedback for programming tasks	Focuses mainly on automated assessment; less attention to exhibition, teamwork, and public feedback
Hellín et al. [6]	Gamified learning environment	Improves motivation, attendance, confidence, and engagement	Emphasizes motivational design rather than multidimensional assessment of student projects
Mekterović et al. [7]	Scaling automated programming assessment systems	Supports large-scale assessment of programming code	Scalability and integration with broader assessment methods remain challenging
Lubarda et al. [8]	Virtual pair programming and online oral exams.	Supports social interaction and academic integrity in remote programming courses	Focuses on pair programming and oral exams; limited discussion of public exhibition-based assessment
Tsai et al. [16]	Design-Based Learning (DBL) for web design skills	Improves web design skills through authentic design tasks and knowledge	Does not explicitly integrate practical exams, visitor feedback, and exhibition-based assessment

		integration	
López-Pimentel et al. [17]	Sustainable project-based learning for web programming	Supports project-based application of programming knowledge	Focuses on PBL methodology; assessment dimensions and exhibition feedback are less explicit
Ubaidullah et al. [18, 20]	Problem-solving and programming learning support	Highlights the role of problem-solving and discussion in computational thinking	Does not provide an integrated assessment model combining process, product, and external feedback
Romero et al. [19]	Creative programming and computational thinking	Connects programming with creativity and higher-level engagement	Focuses on creative programming levels rather than classroom assessment implementation
Kuo et al. [21]	Web programming self-efficacy and learning performance	Uses final website project scores as evidence of learning performance	Mainly emphasizes final project performance; less attention to assessment sequence and visitor feedback

3. METHODOLOGY

This study utilized a mixed-methods approach within a research and development (R&D) framework to evaluate the effectiveness of integrated assessment tools in a Basic Web Programming course. The research design followed a systematic four-stage process: needs analysis, instrument design, implementation, and evaluation. The mixed-methods approach was chosen to integrate quantitative data (numerical and statistical) with qualitative insights (narrative and descriptive) to provide a more comprehensive and holistic understanding of the research problem.

3.1 Participants and setting

This study was conducted in the Basic Web Programming course offered by the Informatics Engineering Program at a private university in Indonesia. At the beginning of the semester, 89 first-year undergraduate students were enrolled in the course. As part of the course assessment innovation, students were invited to participate in a student work exhibition challenge, in which they presented their web programming projects to a broader audience. The final analytic sample consisted of 61 first-year students who accepted the challenge and completed the required assessment activities.

The participants were selected using a purposive classroom-based sampling approach based on clearly defined inclusion criteria. The inclusion criteria were: (1) being enrolled in the Basic Web Programming course during the semester of implementation, and (2) participating in the assessment activities required for evaluating the proposed assessment model, particularly the student work exhibition challenge. Students who did not complete the full sequence of assessment activities or did not participate in the final exhibition were excluded from the final analysis.

The final sample consisted of 46 male students and 15 female students. All participants were first-year undergraduate students in the Informatics Engineering Program. Therefore, the sample represents students who completed the full assessment sequence and participated in the student work exhibition challenge within the observed Basic Web Programming course context. The findings should be interpreted as context-specific to first-year Informatics Engineering students in this course and institution, rather than as broadly representative of all Information Technology students or all web programming courses.

3.2 Data collection and instruments

Data collection was facilitated through several instruments to capture student assessment outcomes and perceptions of the student work exhibition. The instruments and data consist of:

(1) Assessment Component Records: Assessment records were collected from students who completed the required assessment activities.

(2) Student Satisfaction Surveys: Following the final examination, students were invited to complete a questionnaire designed to measure their perceptions of course quality and the student work exhibition experience. The survey was voluntary, and 46 valid responses were obtained from the final analytic sample of 61 students.

Table 2. Operational definitions and examples of the six assessment dimensions

Dimension [11]	Operational Definition	Example Evidence in the Basic Web Programming Course
Technical skills	Students’ ability to create, modify, and present functional web pages using Basic Web Programming concepts.	A web page that runs properly, uses appropriate HTML structure, applies CSS styling, and demonstrates basic interaction or navigation.
Conceptual understanding	Students’ ability to explain and apply fundamental web programming concepts, including page structure, styling, layout, and interaction logic.	Students can explain the purpose of their design choices, page structure, styling decisions, and basic program logic during project discussion or exhibition.
Teamwork	Students’ ability to collaborate, divide responsibilities, communicate, and contribute to the completion of a group-based project.	Students work together to prepare the project, divide development tasks, support presentation preparation, and respond to feedback as a team.
Creativity	Students’ ability to develop original ideas, interface designs, project themes, or features beyond minimum task requirements.	A project shows a unique theme, attractive interface, meaningful content, or additional features that reflect student initiative.
Engagement	Students’ active participation, persistence, and willingness to complete learning and exhibition activities.	Students attend learning activities, complete project preparation, participate in the exhibition, and interact with visitors during the event.
Problem solving	Students’ ability to identify technical or design problems, troubleshoot errors, and revise their work during project development.	Students revise layout problems, fix broken links or code errors, improve usability, or respond constructively to questions and suggestions.

Note: CSS = Cascading Style Sheets, HTML = HyperText Markup Language.

(3) Exhibition Visitor Surveys: Feedback from exhibition visitors was collected through a separate questionnaire employing a 5-point rating scale. Responses were obtained from both onsite and online visitors who attended the exhibition. In total, 735 valid responses were included in the analysis.

(4) Assessment Dimension Framework: The proposed assessment model was informed by six assessment dimensions that were used to guide the design and interpretation of the proposed assessment model: technical skill, conceptual understanding, teamwork, creativity, engagement, and problem-solving. These dimensions were not treated as separate inferential statistical variables in this study; rather, they served as an operational framework for describing the types of learning outcomes supported by practical assignments, programming projects, practical examinations, and the student work exhibition [11]. The operational definitions and examples of evidence for the six assessment dimensions are presented in Table 2.

In the exhibition context, visitor feedback was collected using a 1–5 rating scale for the student works observed by visitors. A score of 1 indicated very low appreciation, whereas a score of 5 indicated very high appreciation of the work exhibited. Visitors could also provide questions, comments, criticisms, or suggestions for the student work and for the overall exhibition event. Therefore, the visitor rating was used as descriptive feedback on public appreciation of student work, while the six assessment dimensions were used as a framework for interpreting the educational value of the assessment model.

3.3 Survey design and qualitative feedback procedure

The student and visitor questionnaires were designed to capture perceptions of the assessment model and the student work exhibition. The design of the questions was aligned with the purpose of the study, namely, to evaluate and support students' learning experience in the Basic Web Programming course. The questionnaires used closed-ended rating-scale items and open-ended feedback prompts.

The student questionnaire employed a 1–10 rating scale to determine students' satisfaction with the quality of the course and the experience of the exhibition. The rating scale was developed to provide a straightforward and familiar answer structure to the students, while allowing variance in perceived pleasure. In addition to rating-scale items, students could also submit written comments regarding their learning experience, the value of practical assignments and programming projects, and their assessment of the student work exhibition. In the visitor questionnaire, visitor assessment of the student works observed during the exhibition was collected using a 1–5 rating system. A score of 1 indicated very low appreciation, whereas a score of 5 indicated very high appreciation. The visitor questionnaire also featured open-ended prompts for questions, comments, critiques or ideas about the student projects and the exhibition event.

Qualitative data were collected via open-ended survey answers, classroom observation notes, peer evaluation information and informal input during the project display. Since the qualitative part of the study was utilized to enhance interpretation and not to perform a comprehensive qualitative investigation, no formal interview methodology or separate interview coding system was employed. Qualitative feedback was evaluated topically to discover reoccurring remarks on involvement, project quality, teamwork, creativity, technical issues and presentation. These themes were then utilized to help with the interpretation of the descriptive survey findings.

Future implementations should establish a more formal interview guide, use systematic qualitative coding techniques, and engage multiple coders for coding agreements to increase repeatability. The current research employed qualitative information as supporting evidence for triangulation and not as distinct qualitative data collection.

3.4 Practical implementation and grading procedure

We assessed the evaluation approach within the normal framework of the Basic Web Programming course exhibition. The grading standards were based on the instructor's course evaluation criteria. The practical activities were evaluated according to the student's ability to accomplish the specified programming tasks, to apply the appropriate HTML, CSS and fundamental interaction principles and to overcome technical challenges that arose throughout the learning process. We assessed programming projects based on project completeness, functionality, design quality, innovation, and the student's ability to describe their creation. We conducted practical assessments to test individual comprehension and competency in the application of web programming principles under examination settings. The student work exhibition was an extra evaluation and feedback activity, enabling students to present their projects and get opinions from visitors, peers and instructors.

In this implementation, a formal weighted scoring rubric was not applied independently to each of the six characteristics of the evaluation. Instead, the six dimensions acted as an operational and interpretative framework to express the learning objectives supported by the assessment model. The visitor evaluations were not used as official marks, but as descriptive feedback reflecting public appreciation of student projects. The visitor survey answers were thus used as proof of exhibition reception instead of official performance evaluations.

Rater preparation: The instructor served as the lead academic rater for the course learning goals and evaluation criteria. Visitors were given a basic grading system and encouraged to voice admiration, comments, questions, concerns, or ideas about the student work. However, no official rater training was undertaken for the visits since they were not regular academic raters. Similarly, inter-rater reliability was not determined in this research since the visitor assessments were descriptive feedback rather than formal academic grading.

The findings should be interpreted with this restriction in mind. In future implementations, analytic rubrics with precise explicit weighting of each facet of assessment should be developed; instructors and evaluators should receive formal rater training; and inter-rater reliability should be calculated when multiple raters are used. These modifications would enhance the repeatability, reliability and comparability of the evaluation model across various classes and institutions.

3.5 Course integration and operational workflow

The proposed assessment model was included in the routine of the course Basic Web Programming. The execution of the course was conducted from the introduction of concepts, guided practice, project creation, practical assessment, exhibition preparation, public presentation, and reflection of feedback. In this sequence, evaluation was not only an end-of-course activity, but an activity inside the learning process throughout the semester.

At the beginning of the course, learning objectives, assessment components, project requirements, and exhibition expectations were introduced to students. Participants were informed that their learning performance would be evaluated through practical assignments, programming projects, practical examinations, peer contributions, and participation in the student work exhibition.

During regular learning sessions, fundamental web programming concepts, including site structure, layout, design, and basic interactivity, were introduced. Students then completed practical tasks that required the direct application of these concepts. Throughout the practical activities, the instructor monitored student progress, provided technical guidance, addressed programming difficulties, and delivered formative feedback. These activities generated process-based evidence related to technical skills, engagement, and problem-solving.

Once students had sufficient fundamental competency, they worked on programming projects individually or in groups as planned by course activities. Throughout this phase, students produced web-based products, using the knowledge and skills acquired throughout the course. The teacher helped students to define project scope, monitor project progress, encourage teamwork and assist students with modifying technical or design concerns. Information from peer evaluation was utilized to help comprehend individual contribution and collaboration. The evaluation of individual comprehension and competency in the application of web programming ideas under examination settings was performed via practical exams. These tests complimented the project-based activities to ensure that students were assessed not just via group output or display performance, but also through individual demonstrations of competence.

Towards the conclusion of the term the students prepared their projects for the student work exhibition. During the last phase, the teacher helped students to finish their web projects, create presentation material, defend design choices and answer any queries from visitors. During the exhibition, students showed their work to visitors, peers and instructors. Visitor ratings and open-ended comments were collected through exhibition visitor surveys. The comments collected after the exhibition were utilized as descriptive evidence to reflect on the project quality, presenting abilities, involvement and public appreciation of the students. The operational sequence of the course integration is shown in Table 3.

This workflow shows that the assessment model was embedded throughout the course rather than added only as a final evaluation. The integration of practical tasks, projects, examinations, exhibition, and feedback enabled students to demonstrate learning through both process-based and product-based evidence.

Table 3. Operational workflow of course integration

Course Phase	Timing	Student Activities	Instructor Guidance	Assessment Evidence
Orientation	Beginning of course	Understand learning objectives, assessment components, project expectations, and exhibition challenge	Explain course goals, assessment components, project requirements, and exhibition process	Student readiness and understanding of assessment structure
Concept introduction and guided practice	Regular sessions	Learn web programming concepts and complete practical assignments	Explain concepts, demonstrate examples, monitor progress, and provide formative feedback	Practical assignment completion, technical skills, engagement, problem-solving
Project development	Middle to late course	Develop web projects individually or in groups	Guide project scope, monitor progress, support revision, and encourage teamwork	Project progress, creativity, conceptual application, teamwork
Practical examination	Midterm/final assessment period	Complete individual practical examination tasks	Assess individual competence under examination conditions	Individual technical and conceptual competence
Exhibition preparation	Before final exhibition	Finalize web projects and prepare presentation materials	Guide project refinement, presentation preparation, and response strategies	Project readiness, presentation preparation, communication
Student work exhibition	End of course	Present projects to visitors, peers, and instructors	Facilitate exhibition process and observe student work exhibition	Project demonstration, visitor feedback, engagement, public appreciation
Reflection and evaluation	After exhibition	Review feedback and reflect on learning experience	Interpret feedback and connect it to learning outcomes	Student reflection, qualitative feedback, perceived learning support

3.6 Data analysis

Data exported from Google Forms was verified for completeness and validity in Microsoft Excel prior to analysis. Only completed voluntary replies to the student survey were included in the study. A student satisfaction survey was completed by 46 of the 61 students in the final analytic sample; survey findings were presented using 46 as the response denominator. The analysis of the visitor survey comprised all visitor replies for a total of 735 valid responses.

Missing data were treated using a complete-case technique. No imputation was used since the analysis was descriptive and only included valid completed answers. Students who did not complete the requisite series of assessments or participate in the project showcase were eliminated from the final analytic sample. Outlier screening was done via range checks based on pre-defined response scales. The answers for student satisfaction were validated using the legitimate scale of 1–10 and visitor satisfaction was validated with the legitimate scale of 1–5. Replies beyond the prescribed range of scales or incomplete replies were not analyzed.

Adjustment for multiple comparisons was not performed since this research used descriptive analysis and did not undertake multiple inferential statistical tests. The findings were consequently viewed as descriptive evidence of the students' and visitors' perspectives on the assessment model and project display rather than inferential evidence for generalization to a large population level.

Qualitative data from classroom observation and peer evaluation helped evaluate student participation, individual contribution and creative processes that could not be conveyed by numerical ratings alone. To evaluate the suggested evaluation methodology, data triangulation was achieved by combining quantitative survey findings with qualitative classroom evidence.

The six assessment dimensions were used as an interpretive framework for discussing the educational contribution of the assessment model. They were not analyzed as separate statistical variables because no formal weighted scoring rubric was applied to each dimension in this implementation.

This research was exploratory and descriptive, with the intention of doing a requirements analysis, to build an evaluation model and to gather first input on implementation. Therefore, inferential statistical analysis was not performed. The quantitative data supplied were mostly satisfaction ratings and scores of visitor appreciation that were utilized to reflect impressions of the evaluation model rather than to verify causal effects. Furthermore, the research lacked a randomized control group, pre-test/post-test design, and repeated standardized performance measurements that would provide support to reliable causal inference, effect size calculation, or multivariate modeling. The data were therefore interpreted descriptively and should not be regarded as statistically verified proof of effectiveness.

4. RESULTS AND DISCUSSION

As shown in Table 4, the results indicate positive responses from both students and exhibition visitors. Most students reported high satisfaction with the course quality and practical learning activities. Similarly, the student work exhibition received strong approval from students, suggesting that the exhibition functioned not only as a final presentation activity but also as additional space for demonstrating projects, explaining design decisions, and receiving feedback.

Visitor responses further supported the educational value of the exhibition. The high proportion of positive visitor ratings indicates that the exhibited student projects were well received by external audiences. In the context of web programming education, this public feedback is important because it exposes students to real audience expectations related to usability, interface design, content clarity, creativity, and project presentation. The overall distribution of high satisfaction responses is summarized in Figure 1.

Table 4. Summary of key quantitative results

Indicator	Respondents	Key Result	Interpretation
Course quality satisfaction	46 students	95.6% rated 8–10; 47.8% rated 10/10	Students perceived the course and practical learning activities positively
Exhibition experience satisfaction	46 students	91.3% rated 8–10	Students responded positively to the exhibition as part of the assessment process
Visitor satisfaction with exhibition	735 visitors	97.2% rated 4–5; 78.2% rated 5/5	Visitors gave strong positive appreciation to the exhibited student works

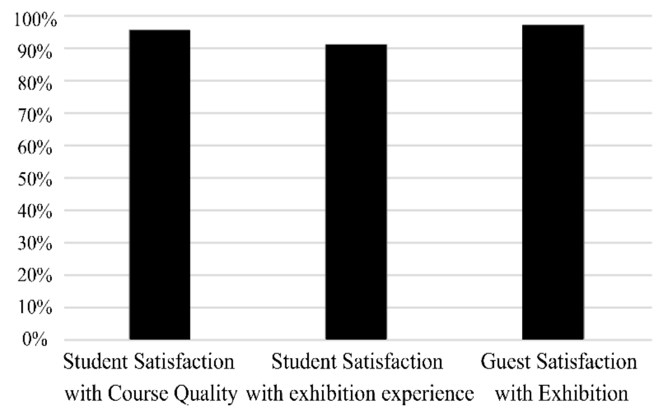

Figure 1. Summary of high satisfaction responses from students and exhibition visitors

As indicated in Figure 1, the three satisfaction indicators were always at a high level, and all the values were higher than 90%. The pattern shows that the proposed assessment model of evaluation was favorably appraised both by students as participants of the course and by visitors who assessed the display of the student work.

The findings show that a combination of all components might provide a more holistic evaluation framework than a sole reliance on midterm and final exams. Traditional tests are still effective for measuring conceptual comprehension but have limitations in evaluating students' abilities to create working online products, collaborate, solve technological challenges, and report on project results. Thus, the suggested methodology allows a formal evaluation with actual assessment activities closer to real programming practice.

The six assessment aspects provide a framework for interpreting the contribution of the proposed assessment model. Practical work supports mainly technical skills and problem solving, as students have to develop, test and rewrite code. Programming projects promote conceptual knowledge, creativity, and cooperation since students need to learn how to combine HTML, CSS, JavaScript, interface design, and group coordination into a comprehensive online product. The exhibition encourages interaction and communication since students are required to show their work, answer questions and react on criticism from a wider audience. Thus, the evaluation paradigm pushes the students from task completion to applied competency and reflective learning. Table 5. Mapping of Assessment Dimensions, Instruments, Implementation Methods, and Schedules for Basic Web Programming Course.

4.1 Comparative analysis with existing Project-Based Learning and student work exhibition methods

Although PBL and student work exhibition have been widely used in programming and design-related courses, the contribution of this study lies in the integration of these activities into a structured assessment model for Basic Web Programming. In conventional PBL implementation, the main focus is often placed on the completion of a final project. Similarly, student work exhibition is commonly used as presentation or showcase activities at the end of a course. In contrast, the proposed assessment model connects practical assignments, programming projects, practical examinations, and student work exhibition into one assessment sequence.

Table 5. Assessment components and learning dimensions in the Basic Web Programming course

Dimension [11]	Instrument [11]	Learning Evidence	Assessment Purpose	Expected Competency
Technical skills	Hands-on Tasks	Practical coding exercises and implementation outcomes	Evaluate students' ability to apply web programming concepts in practical situations	Programming proficiency and technical competence
	Coding Projects	Functional web applications and project artifacts	Assess the integration of programming concepts into complete applications	Software development capability
	Midterm Exam	Examination scores and practical test results	Measure technical knowledge and skills acquired during the first half of the course	Foundational programming competence
	Final Exam	Comprehensive examination results	Evaluate overall mastery of web programming competencies	Comprehensive technical mastery
Conceptual understanding	Midterm Exam	Written examination responses	Assess understanding of fundamental concepts such as HTML, CSS, JavaScript, and web architecture	Theoretical knowledge
	Final Exam	Written examination responses	Evaluate conceptual mastery across the entire course content	Conceptual integration and retention
Teamwork	Coding Projects	Project documentation and collaboration records	Assess students' ability to work effectively in groups	Collaborative working skills
	Peer Assessment	Peer evaluation forms and contribution reports	Evaluate individual contributions within team projects	Individual accountability and collaboration
Creativity and innovation	Coding Projects	Originality of design, features, and implementation	Assess the development of innovative and creative solutions	Innovative and creative thinking
	Student Work Exhibition	Project presentations and exhibition feedback	Evaluate students' ability to communicate and showcase project outcomes	Communication and presentation skills
Engagement and participation	Practical Assignments	Participation records and completion rates	Measure active involvement in practical learning activities	Learning engagement
	Classroom Interaction	Observation records and participation logs	Assess engagement in discussion, Q&A sessions, and collaborative activities	Active participation and communication
Problem-solving ability	Practical Assignment	Problem-solving performance during coding activities	Evaluate students' ability to identify, analyze, and resolve programming challenges	Analytical and problem-solving skills
	Coding Projects	Solutions implemented within project development	Assess application of problem-solving strategies in authentic context	Complex problem-solving competence

Table 6. Comparison between conventional Project-Based Learning (PBL)/exhibition methods and the proposed assessment model

Aspect	Conventional PBL or Student Work Exhibition	Proposed Assessment Model
Main focus	Final project completion or end-of-course presentation	Integrated assessment sequence from practical tasks to exhibition
Assessment evidence	Mostly final product or presentation outcome	Process-based and product-based evidence
Assessment components	Project work and presentation	Practical assignments, programming projects, practical examinations, and student work exhibition
Learning dimensions	Often implicit or not systematically mapped	Explicitly mapped to technical skills, conceptual understanding, teamwork, creativity, engagement, and problem-solving.
Feedback source	Mainly instructors or peers	Instructor, peers, and external exhibition visitors
Role of exhibition	Showcase of student work	Assessment space for demonstration, communication, feedback, and reflection
Contribution	Supports active learning and project completion	Provides a structured framework for authentic and multidimensional assessment in Basic Web Programming

The specific improvement of the proposed assessment model is that each assessment component is linked to observable learning dimensions. Practical assignments are used to capture technical skills and problem-solving during regular learning activities. Programming projects are used to assess conceptual understanding, creativity, and teamwork. Practical examinations are retained to evaluate individual understanding and competence. The student work exhibition extends the assessment process by allowing students to present their work to a broader audience and receive feedback from visitors, peers, and instructors.

Compared with conventional PBL or exhibition-based assessment, this model does not treat the final project as the only evidence of student achievement. Instead, it combines process-based and product-based assessment. The process dimension is represented by practical assignments, classroom activities, project development, and peer contribution, while the product dimension is represented by the final project and exhibition performance. This structure provides a more comprehensive view of student learning in web programming because it captures not only the final output but also the learning process, collaboration, creativity, engagement, and problem-solving.

Therefore, the innovation claimed in this study should be understood as an instructional and assessment integration rather than a completely new educational method. The model adapts existing PBL and student work exhibition practices into a more structured assessment framework that is aligned with the learning needs of Basic Web Programming students and the smart campus learning environment. To clarify the specific contribution of the proposed assessment model, Table 6 compares conventional PBL or student work exhibition practices with the assessment model developed in this study.

4.2 Limitations and generalizability

The applicability of the proposed assessment model may vary across courses and educational contexts. The model is particularly suitable for practice-oriented computing courses, such as web programming, software development, database systems, and user interface design, where students are required to produce observable project outcomes. Its implementation in theoretical courses or non-computing disciplines may require adaptation of the exhibition format, assessment criteria, and feedback mechanisms. Consequently, the findings should be interpreted as analytically transferable rather than statistically generalized. Although the model provides a structured framework for integrating practical assignments, projects, examinations, and public exhibitions, its effectiveness may be influenced by contextual factors, including class size, student characteristics, instructor preparedness, technological infrastructure, and institutional support.

Another limitation concerns the assessment reliability procedure. This study did not include formal rater training or inter-rater reliability analysis because the instructor served as the main academic evaluator, while visitor ratings were used as descriptive feedback rather than formal grading scores. Future research should develop a standardized analytic rubric, involve multiple trained raters, and calculate inter-rater reliability to strengthen the reliability and reproducibility of the assessment model. Successful implementation of practical assessments in the smart campus ecosystem requires a user-centered design approach and ethical data governance to ensure long-term effectiveness [22].

5. CONCLUSIONS

The development of suitable evaluation tools for Basic Web Programming is vital to promote learning quality and student outcomes from this study. Through a comprehensive requirements analysis, significant assessment needs were identified, suggesting that a comprehensive assessment framework should incorporate should include practical web programming abilities, conceptual knowledge, creativity, collaboration, engagement, and problem-solving. The implementation gave descriptive evidence that the assessment approach was well regarded by the students and exhibition attendees alike. The student work exhibition extended assessment beyond conventional testing by enabling authentic demonstration, feedback exchange, and public engagement.

ACKNOWLEDGMENT

The research team gratefully acknowledges the invaluable support and assistance provided by the School of Electrical Engineering and Informatics at the Bandung Institute of Technology (ITB) and Maranatha Christian University in Bandung throughout the course of this study.

REFERENCES

[1] Imbar, R.V., Supangkat, S.H., Langi, A.Z., Arman, A.A., Johan, M.C. (2025). Smart campus framework: Definition, model, measurement from anthropocentric, systemic and technological perspectives. Journal of ICT Research and Applications, 18(3): 258-269. https://doi.org/10.5614/itbj.ict.res.appl.2025.18.3.5

[2] Pradana, F., Setyosari, P., Ulfa, S. (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

[3] 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

[4] 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, 12(18): 7451. https://doi.org/10.3390/SU12187451

[5] Sendeku, H.A, Daniel, R., Gopal, G.V. (2025). An AI-powered adaptive learning framework for personalized education, Information Dynamics and Applications, 4(4): 212-223. https://doi.org/10.56578/ida040403

[6] 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, 15(19): 117-130. https://doi.org/10.3390/su151914119

[7] Mekterović, I., Brkić, L., Horvat, M. (2023). Scaling automated programming assessment systems. Electronics, 12(4): 942. https://doi.org/10.3390/electronics12040942

[8] Lubarda, M.V., Phan, A.M., Schurgers, C., et al. (2025). Virtual pair programming and online oral exams: Effects on social interaction, performance, and academic integrity in a remote computer programming course. Computer Science Education, 35(3): 482-522. https://doi.org/10.1080/08993408.2024.2344401

[9] Ou, Q., Liang, W., He, Z., Liu, X., Yang, R., Wu, X. (2023). Investigation and analysis of the current situation of programming education in primary and secondary schools. Heliyon, 9(4): e15530. https://doi.org/10.1016/j.heliyon.2023.e15530

[10] Johan, M.C., Langi, A.Z., Nugraha, I.G.B.B., Imbar, R.V., Sinaga, P. (2026). Orchestrating value co-creation in digital education using the TISE-VALORIZE framework: From knowing to becoming. Frontiers in Education, 11: 1827542. https://doi.org/10.3389/feduc.2026.1827542

[11] Johan, M.C., Suteja, B.R., Wijaya, R.Y., Benedict, J.C., Santika, A.F. (2026) Improving student learning in Web programming through practical assessment and project exhibitions. Journal of Applied Business and Technology, 7(1): 70-77. https://doi.org/10.35145/n59xzn61

[12] 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
[13] 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
[14] 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
[15] Chen, J., Lau, S., Leinonen, J., Terragni, V., Giacaman, N. (2024). Detecting learning behavior in programming assignments by analyzing versioned repositories. IEEE Access, 12: 188828-188844. https://doi.org/10.1109/ACCESS.2024.3514843
[16] 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(4): 4. https://doi.org/10.1186/s41239-021-00308-4
[17] 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, 13(15): 8482. https://doi.org/10.3390/su13158482
[18] Ubaidullah, N.H., Mohamed, Z., Hamid, J., Sulaiman, S. (2021). Discovering the role of problem-solving and discussion techniques in the teaching programming environment to improve students' computational thinking skills. International Journal of Information and Education Technology, 11(12): 615-623. https://doi.org/10.18178/ijiet.2021.11.12.1572
[19] 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: 42. https://doi.org/10.1186/s41239-017-0080-z
[20] 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
[21] 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): 1236. https://doi.org/10.3390/educsci13121236
[22] Suazo-Galdamés, I.C., Chaple-Gil, A.M. (2025). Impact of intelligent systems and AI automation on operational efficiency and user satisfaction in higher education. Ingénierie des Systèmes d'Information, 30(4): 1057-1066. https://doi.org/10.18280/isi.300421

NOMENCLATURE

CSS	Cascading Style Sheets
CT	Computational Thinking
DBL	Design-Based Learning
HTML	HyperText Markup Language
ICT	Information and Communication Technology
IT	Information Technology
PBL	Project-Based Learning
Q&A	Questions and Answers
R&D	Research and Development
RQ	Research Question