Asian Journal of Multidisciplinary Studies
Vol. 9, No. 1, (2026)
ISSN 2651-6691 (Print)
ISSN 2651-6705 (Online)

Development of a Dedicated LMS with Integrated Coding Platforms for IT Students of Pangasinan State University

Frederick F. Patacsil[1], Christine Lourrine S. Tablatin[2], Michael E. Acosta[3], Rhenel R. Bernisca[4], Nahum M. Quiros[5], Joshua C. Reyes[6], Paul Andrew V. Roa[7], Arni-Rie F. Tamayo[8]
[1,2,3,4,5,6,7,8]Pangasinan State University, Philippines

Article Info:
Received: ***10 Feb 2026****; Revised:* ***28 Feb 2026****; Accepted:* ***2 March 2026****; Available Online:* ***13 March 2026***

Abstract – *This study presents the design, development, and evaluation of a Learning Management System (LMS) platform tailored for the Information Technology students of Pangasinan State University. A descriptive-development research design was employed, combining needs assessment, system development using the Rapid Application Development (RAD) model, pilot implementation, and system evaluation. In order to prioritize the features of the system, the proponents surveyed 60 IT students to assess performance score, diagnostic testing, automatic level assignment, and the ability to retake assessments, as essential LMS components. An adaptive learning mechanism was implemented using a rule-based decision tree that personalizes content delivery based on performance indicators such as code accuracy, time efficiency, and consistency. A code-execution platform with AI assistance was integrated into the LMS to support learners' programming practices. Evaluation results showed high to very high effectiveness across usability, coding integration, material organization, learning enhancement, and overall satisfaction. The results also indicate that the LMS with an integrated coding platform and adaptive learning improves the students' programming experience, supports adaptive instruction, and promotes self-directed learning among the IT students.*

Keywords – *Learning Management System, adaptive learning, programming education, coding platform, rule-based decision tree*

INTRODUCTION

The ever-evolving field of information technology (IT) requires a dynamic approach to education that keeps pace with the industry's rapid advancements. The Pangasinan State University, a leading provider of IT education in the Philippines, recognizes the need to develop a tailored learning management system (LMS) that integrates live coding environment to better serve its IT student population. This research paper outlines the design, development, and implementation of a dedicated LMS with embedded coding tools, aimed at enhancing the learning experience and fostering the technical skills of IT students. Current e-learning systems typically provide content to students as a whole, requiring significant effort from the learners to identify, discriminate, and classify the relevant materials for their specific topics of study (Selvi & Panneerselvam, 2012). To address this challenge, the proposed LMS will leverage adaptive learning techniques to dynamically deliver content and programming exercises tailored to each student's knowledge level and learning pace. The research team will adopt a multifaceted approach in developing the LMS, drawing insights from a thorough review of related literature and gaining the design principles of self-regulated, learning-oriented online programming teaching and learning systems (Shershneva et al., 2018).

OBJECTIVES OF THE STUDY

Specifically, this study aims to: (1) Design and develop a dedicated learning management system for IT students at Pangasinan State University, integrating comprehensive coding platforms and tools; (2) Incorporate adaptive learning mechanisms within the LMS to personalize the delivery of content and programming exercises based on individual student performance and learning progress; (3) Evaluate the effectiveness of the integrated LMS-coding platform in

enhancing the overall learning experience of IT students at Pangasinan State University, including improvements in academic performance, technical skill development, and self-directed learning abilities.

MATERIALS AND METHODS

This study employed a descriptive–developmental research design. The descriptive component was utilized to determine the needs, preferences, and priority features of a dedicated Learning Management System (LMS) for Information Technology students of Pangasinan State University (PSU). The developmental component focused on the design, development, implementation, and pilot testing of the LMS integrated with coding platforms and adaptive learning mechanisms. The study followed a systematic process such as needs assessment, system design, and development using an iterative model, pilot implementation, and evaluation of effectiveness.

The population of the study consisted of all Information Technology (IT) students of Pangasinan State University. For pilot testing, the sample included First Year IT students of PSU Urdaneta Campus, particularly those enrolled in CC 102 Fundamentals of Programming and CC 103 Intermediate Programming. The pilot implementation utilized the Java Programming Language as the primary coding environment embedded within the LMS.

For Objective 1, the researchers adopted the Rapid Application Development (RAD) model to accelerate system development while ensuring continuous user feedback. The RAD model consisted of Requirements Planning, User Design, Construction, and Cutover (Deployment). This approach enabled iterative prototyping, immediate refinement, and stakeholder involvement throughout development. To determine essential LMS features, the researchers conducted a survey among IT students. Each proposed system feature was rated using a 5-point importance scale: 5 - Very Important, 4 - Important, 3 - Moderately Important, 2 - Slightly Important, and 1 - Not Important. The responses were tallied, computed using weighted mean, and interpreted using the following decision framework:

Table 1. 5-point Importance Scale

Weighted Mean Range	Interpretation	Development Action
5.00 – 3.50	Essential	Must Build
3.49 – 2.50	Important	High Priority
2.49 – 1.50	Low Importance	Low Priority
1.49 – 1.00	Optional	Backlog

Only features categorized as Essential (Must Build) and Important (High Priority) were included in the initial system release.

For Objective 2, the researchers implemented a Rule-Based Decision Tree Adaptive Learning Mechanism to support personalization. The adaptive learning engine followed this structured process: Input Data Capture, Performance Evaluation, Level Classification, Content Recommendation, and Feedback Loop.

For Objective 3, the evaluation instrument was developed by the researchers and validated by an academic expert (statistician) and an industry professional in software development. The instrument measured Academic Performance Perception, Technical Skill Development, Self-Directed Learning, System Usability, and Overall Learning Experience. The validated survey was administered to students enrolled in CC 102 Fundamentals of Programming and CC 103 Intermediate Programming at PSU Urdaneta Campus. The responses were tallied, computed using weighted mean, and interpreted using the following decision framework:

Table 2. Weighted Mean Interpretation Scale

Weighted Mean Range	Interpretation
4.51 – 5.00	Very High Effectiveness (Strongly Agree)
3.51 – 4.50	High Effectiveness (Agree)
2.51 – 3.50	Moderate/Neutral
1.51 – 2.50	Low Effectiveness (Disagree)
1.00 – 1.50	Very Low Effectiveness (Strongly Disagree)

Participation was voluntary. Respondents were informed about the purpose of the study. Data confidentiality and anonymity were ensured. Results were used solely for academic and research purposes.

RESULTS AND DISCUSSION

Design and development of a dedicated learning management system

A total of 60 IT students participated in the feature selection survey. The responses were tallied and analyzed using Weighted Mean, interpreted using the established development decision framework.

Table 3. Feature Prioritization Result

Feature	Mean	Interpretation	Development Action
Performance score calculation	4.52	Essential	Must Build
Initial diagnostic test	4.48	Essential	Must Build
Automatic level assignment	4.04	Essential	Must Build
Retake diagnostic assessment	4.04	Essential	Must Build

All identified features obtained weighted mean values ranging from 4.04 to 4.52, which fall within the 5.00–3.50 range, interpreted as Essential (Must Build). The results indicate that students strongly prefer a system that automatically computes performance scores provides an initial diagnostic assessment, assigns levels automatically, and allows retaking of diagnostic tests.

The highest-rated feature, Performance Score Calculation (4.52), highlights the importance of transparent and measurable evaluation in programming education. Students value real-time performance tracking, which aligns with outcome-based education principles. The high rating for Initial Diagnostic Test (4.48) suggests that students recognize the need to assess their baseline knowledge before proceeding with programming lessons. This supports differentiated instruction and personalized learning pathways. The strong preference for Automatic Level Assignment (4.04) and Retake Diagnostic Assessment (4.04) indicates students' desire for adaptive progression and mastery learning opportunities.

Based on these findings, the researchers proceeded with system development using the Rapid Application Development (RAD) model. The identified essential features were integrated into the LMS architecture, and a system flowchart was designed to illustrate the complete operational workflow, from user login, diagnostic assessment, adaptive routing, coding activity execution, performance evaluation, and feedback generation. The results confirm that the developed LMS directly reflects user-identified priorities, thereby strengthening system relevance and user acceptance.

Figure 1. Operational workflow of the System

Figure 1. Discusses the operational flow of the system. Upon login, students need to access the introductory learning content and practice exercises to ensure readiness for the Diagnostic Exam, which will determine the student's learning path. The system will provide meaningful feedback whenever the student accomplishes practice exercise problems. After taking the Diagnostic exam, the learner is directed to the corresponding level-specific learning contents and practice exercises. After completing the lessons and exercises, the learner may again take the diagnostic test until he passes and reaches level 2, or if his score will determine, level 3. Once the learner demonstrates sufficient competency across the levels, they may now proceed to take the summative assessment, after which the system evaluates performance and displays the learner's mastery level.

Incorporation of adaptive learning mechanisms within the LMS to personalize the delivery of content and programming exercises

The adaptive learning mechanism was implemented using a Rule-Based Decision Tree Algorithm, characterized by lightweight structure, transparency, and interpretability. This approach was selected to ensure clear logic routing during the early phase of adaptive deployment. The adaptive process consists of five major stages: Input Data Capture, Performance Evaluation, Level Classification, Content Recommendation, and Feedback Loop.

The system collects the following learner performance indicators such as number of attempts, average time per task, code accuracy (pass/fail per test case), and error types (syntax vs. logical errors). These variables serve as quantitative bases for evaluating student performance. The system computes a performance score using the formula:

performance_score = (accuracy * 0.5) + (time_efficiency * 0.2) + (consistency * 0.3)

Table 4. System Performance Evaluation Metrics

Metric	Description	Measurement Logic
Accuracy	Successful compilation and correct output across attempts	1.0 if fully correct; penalties for crashes or incorrect output
Consistency	Stability of correct execution	Percentage of passed test cases
Time Efficiency	Task completion efficiency	1 / (attempts × average time per task), normalized

Time efficiency values were normalized to a 0–1 scale using min-max normalization or sigmoid transformation to maintain compatibility with accuracy and consistency metrics. Without normalization, extreme differences in attempt counts and time duration would distort total scores. For example, a student completing a task in 1 attempt and 1 minute would obtain a raw score of 1.0, while another completing it in 3 attempts and 10 minutes would obtain 0.033, creating disproportionate scoring gaps. To ensure fairness time, efficiency was normalized and level 3 tasks were granted up to 1.2x tolerance weighting to account for increased complexity.

Based on computed performance scores: (a) ≥ 80 → Level 3 (Advanced); (b) 50–79 → Level 2 (Intermediate); and (c) < 50 → Level 1 (Beginner). This classification enables structured mastery-based progression. The LMS integrates Judge0 API for live Java code execution and developer-created instructional content. Content routing is defined as (a) Level 1, Simple exercises + tutorial videos + AI hints; (b) Level 2, standard exercises + AI hints; and (c) Level 3, challenge problems + optional projects. After each activity, updated performance data is recalculated, level classification may change, and new content is recommended. This establishes a continuous recalibration mechanism that ensures adaptive responsiveness.

The rule-based decision tree proved effective for early-phase adaptive routing because it is computationally lightweight, interpretable by instructors, and allows transparent tracking of student progression. The rule-based system allows faculty to understand and modify performance thresholds when necessary. This is particularly beneficial in academic settings where transparency and fairness are critical. The integration of automated coding evaluation, adaptive classification, and dynamic recommendation demonstrates the system's capability to personalize programming education in a structured and measurable manner.

Evaluation of the effectiveness of the integrated LMS-coding platform

A total of 60 students enrolled in CC 102 and CC 103 participated in the system evaluation survey. Responses were analyzed using Weighted Mean and interpreted according to the established effectiveness scale.

Table 5. System Effectiveness Evaluation Results

Item	Mean	Interpretation
Interface usability	4.09	High Effectiveness (Agree)
Coding platform integration	4.05	High Effectiveness (Agree)
Assignment submission	3.82	High Effectiveness (Agree)
Organization of materials	4.14	High Effectiveness (Agree)
Learning enhancement	4.05	High Effectiveness (Agree)
System responsiveness	3.95	High Effectiveness (Agree)
User confidence	3.95	High Effectiveness (Agree)
Feedback usefulness	3.77	High Effectiveness (Agree)
Recommendation	4.55	Very High Effectiveness (Strongly Agree)
Overall satisfaction	4.50	High Effectiveness (Agree)

All evaluated components obtained mean scores between 3.77 and 4.55, indicating High to Very High Effectiveness. The highest-rated item was Recommendation (4.55 – Very High Effectiveness). This suggests strong student acceptance and willingness to recommend the LMS to peers, indicating high perceived value and usability.

Other notable findings include (a) Organization of Materials (4.14), students appreciated structured and systematic content arrangement; (b) Interface Usability (4.09), the LMS interface was perceived as intuitive and user-friendly; (c) Coding Platform Integration (4.05), seamless embedding of the coding environment significantly enhanced learning convenience; and (c) Learning Enhancement (4.05), students agreed that the system improved their understanding of programming concepts. Although all ratings were positive, Feedback Usefulness (3.77) and Assignment Submission (3.82) received comparatively lower (yet still high) ratings.

This suggests opportunities for further improvement in automated feedback detail and submission workflow optimization. Overall satisfaction (4.50) indicates that students generally agree the system enhances their learning experience.

CONCLUSION AND RECOMMENDATION

This study aimed to develop and evaluate a Dedicated Learning Management System (LMS) with Integrated Coding Platforms for IT students of Pangasinan State University. Based on the findings of the study, the following conclusions were drawn:

The study successfully designed and developed a dedicated LMS tailored specifically for IT students. The feature selection results confirmed that performance score calculation, diagnostic testing, automatic level assignment, and retake assessment functionalities were considered essential by students. These identified high-priority features were systematically integrated into the system using the Rapid Application Development (RAD) model. The developed LMS effectively incorporates an embedded coding platform, automated evaluation mechanisms, structured content organization, and performance tracking tools. The system architecture reflects user-centered design principles, ensuring alignment with the actual needs of programming learners. Therefore, the first objective of designing and developing a functional and responsive LMS was fully achieved.

The integration of a rule-based decision tree adaptive engine successfully enabled personalized learning pathways within the LMS. By utilizing performance indicators such as accuracy, time efficiency, and consistency, the system computes a performance score that determines learner classification into Levels 1, 2, or 3. The adaptive routing mechanism ensures that students receive appropriate content recommendations based on their demonstrated competence. The normalization of time efficiency and difficulty-based adjustments enhanced fairness and reliability in performance evaluation. The feedback loop further supports continuous recalibration and progressive learning. The findings confirm that the LMS effectively incorporates an interpretable, lightweight adaptive learning mechanism that supports differentiated instruction and mastery-based progression. Thus, the second objective was satisfactorily attained.

The evaluation results revealed that the integrated LMS-coding platform achieved high to very high effectiveness ratings across all measured indicators. Students expressed high agreement regarding system usability, coding integration, organization of materials, learning enhancement, responsiveness, and confidence development. The very high rating in recommendation indicates strong user acceptance and perceived value. The results demonstrate that the system positively contributes to improved understanding of programming concepts, enhanced technical skill development, increased learner confidence, and strengthened self-directed learning. Therefore, the third objective was achieved, as the LMS was found to be effective in enhancing the overall learning experience of IT students.

The developed LMS may be officially adopted and deployed across IT courses in Pangasinan State University beyond CC 102 and CC 103. Additional programming languages (e.g., Python, C#, PHP, R, Dart, Kotlin) may be integrated into the coding platform to support higher-level courses. System scalability and server optimization should be enhanced to accommodate larger student populations. A mobile-responsive or dedicated mobile application version may be developed to increase accessibility.

Future enhancements may incorporate machine learning algorithms to complement the rule-based decision tree for more advanced personalization. Adaptive thresholds may be refined through longitudinal data analysis to further improve accuracy in level classification. More detailed analytics dashboards may be provided for instructors to monitor adaptive progression. Gamification elements (badges,

achievements, progress bars) may be integrated to further motivate learners.

Continuous system evaluation should be conducted each semester to ensure sustained effectiveness and usability improvements. The feedback mechanism may be enhanced to provide more detailed automated explanations for coding errors. A quasi-experimental study comparing LMS users and non-users may be conducted to measure objective improvements in academic performance. The system may be expanded for use in other university computing-related programs.

Future studies may examine the impact of adaptive LMS systems on the long-term development of programming competency. Comparative studies between rule-based and AI-driven adaptive engines may be conducted. Integration of AI-assisted debugging tools may be investigated. Broader multi-campus implementation studies may be performed to assess institutional scalability.

REFERENCES

Anonah, E. (2017). Paradigm and architecture of computing-augmented learning management system for computer science education. IGI Global, 7(2), 60–70. https://doi.org/10.4018/ijopcd.2017040105

Eom, S. B., & Laouar, M. R. (2020). The emergence of digital distance learning. https://doi.org/10.1145/3447568.3448524

Huang, T., Shu, Y., Chang, S., Huang, Y., Lee, S., Huang, Y., & Liu, C. (2014). Developing a self-regulated oriented online programming teaching and learning system. IEEE. https://doi.org/10.1109/tale.2014.7062599

Sanati, F., Soon, L., & Lin, Y. (2020). Intelligent teaching and learning platform for introductory programming subjects. Proceedings of the ACM. https://doi.org/10.1145/3408066.3408100

Seeling, P. (2016). Evolving an introductory programming course: Impacts of student self-empowerment, guided hands-on times, and self-directed training. IEEE. https://doi.org/10.1109/fie.2016.7757488

Selvi, S. T., & Panneerselvam, K. (2012). A self-regulated learning approach for programming language using cloud-based learning management system. IEEE. https://doi.org/10.1109/icrtit.2012.6206810

Shershneva, V. A., Vainshtein, Y., & Kochetkova, T. (2018). Adaptive system of web-based teaching. International Journal of Emerging Technologies in Learning, 9(4), 179–197. https://doi.org/10.25209/2079-3316-2018-9-4-179-197

Thuseethan, S., Achchuthan, S., & Kuhanesan, S. (2014). Usability evaluation of learning management systems in Sri Lankan universities. arXiv. https://doi.org/10.48550/arxiv.1412.0197

Xie, H., Chu, H., Hwang, G., & Wang, C. (2019). Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017. Computers & Education, 140, 103599. https://doi.org/10.1016/j.compedu.2019.103599

PLEASE INCLUDE CONTACT INFORMATION:
NAME: FREDERICK F. PATACSIL
CONTACT NO: +63932 277 4130
EMAIL ADDRESS:
FREDERICKPATACSIL@YAHOO.CO.UK

NAME: CHRISTINE LOURRINE S. TABLATIN
CONTACT NO: +639778122232
EMAIL ADDRESS: CTABLATIN@PSU.EDU.PH

NAME: MICHAEL E. ACOSTA
CONTACT NO: +639981624892
EMAIL ADDRESS: MACOSTA.URDANETA@PSU.EDU.PH

NAME: RHENEL R. BERNISCA
CONTACT NO: +639199706204

EMAIL ADDRESS: RHENELBERNISCA@PSU.EDU.PH

NAME: NAHUM M. QUIROS
CONTACT NO: +639605351110
EMAIL ADDRESS: NQUIROS@PSU.EDU.PH

NAME: JOSHUA C. REYES
CONTACT NO: +639095959442
EMAIL ADDRESS: MEETJCREYES@GMAIL.COM

NAME: PAUL ANDREW V. ROA
CONTACT NO: +639369028589
EMAIL ADDRESS: PROA@PSU.EDU.PH

NAME: ARNI-RIE F. TAMAYO
CONTACT NO: +639338754730
EMAIL ADDRESS: ARNIERIETAMAYO@PSU.EDU.PH